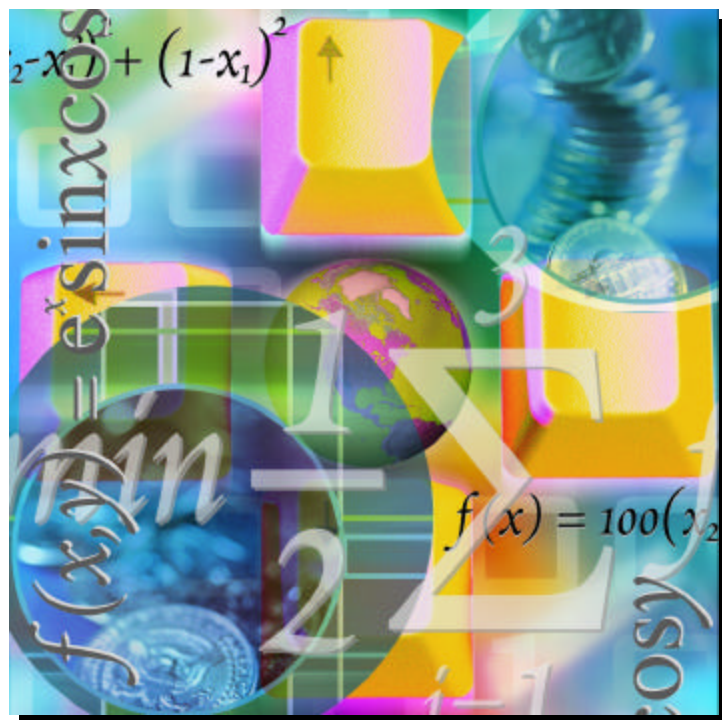




IMSL Fortran Library User's Guide STAT/LIBRARY Volume 1 of 2



Statistical Functions in Fortran

Trusted For Over **30** Years

IMSL[®]

IMSL Fortran Library User's Guide
STAT/LIBRARY Volume 1 of 2

Statistical Functions in Fortran

**Visual Numerics, Inc. – United States
Corporate Headquarters**
2000 Crow Canyon Place, Suite 270
San Ramon, CA 94583
PHONE: 925-807-0138
FAX: 925-807-0145
e-mail: info@vni.com
Westminster, CO
PHONE: 303-379-3040

Houston, TX
PHONE: 713-784-3131

Visual Numerics S. A. de C. V.
Florenca 57 Piso 10-01
Col. Juarez
Mexico D. F. C. P. 06600
MEXICO

PHONE: +52-55-514-9730 or 9628
FAX: +52-55-514-4873

Visual Numerics, Inc.
7/F, #510, Sect. 5
Chung Hsiao E. Road
Taipei, Taiwan 110
ROC

PHONE: +(886) 2-2727-2255
FAX: +(886) 2-2727-6798
e-mail: info@vni.com.tw

Visual Numerics International Ltd.
Sussex House
6 The Forbury
Reading, Berkshire RG1 3EJ
UNITED KINGDOM

PHONE: +44-1-189 25-3370
FAX: +44 -1-189-25-3371
e-mail: info@vniuk.co.uk
Support: support@vniuk.co.uk

Visual Numerics International GmbH
Zettachring 10
D-70567 Stuttgart
GERMANY

PHONE: +49-711-13287-0
FAX: +49-711-13287-99
e-mail: vni@visual-numeric.de

Visual Numerics Korea, Inc.
HANSHIN BLDG. Room 801
136-1, MAPO-DONG, MAPO-GU
SEOUL, 121-050
KOREA SOUTH

PHONE: +82-2-3273-2632 or 2633
FAX: +82-2-3273-2634
e-mail: info@vni.co.kr

Visual Numerics SARL
Immeuble le Wilson 1
70, avenue due General de Gaulle
F-92058 PARIS LA DEFENSE, Cedex
FRANCE

PHONE: +33-1-46-93-94-20
FAX: +33-1-46-93-94-39
e-mail: info@vni.paris.fr

Visual Numerics Japan, Inc.
GOBANCHO HIKARI BLDG. 4TH Floor
14 GOBAN-CHO CHIYODA-KU
TOKYO, JAPAN 102

PHONE: +81-3-5211-7760
FAX: +81-3-5211-7769
e-mail: vnijapan@vnij.co.jp

World Wide Web site: <http://www.vni.com>

COPYRIGHT NOTICE: Copyright 1994-2003 by Visual Numerics, Inc. All rights reserved. Unpublished—rights reserved under the copyright laws of the United States.
Printed in the USA.

The information contained in this document is subject to change without notice.

This document is provided AS IS, with NO WARRANTY. VISUAL NUMERICS, INC., SHALL NOT BE LIABLE FOR ANY ERRORS WHICH MAY BE CONTAINED HEREIN OR FOR INCIDENTAL, CONSEQUENTIAL, OR OTHER INDIRECT DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE OR USE OF THIS MATERIAL. [Carol: note case change]

IMSL, PV- WAVE, and Visual Numerics are registered in the U.S. Patent and Trademark Office by, and PV- WAVE Advantage is a trademark of, Visual Numerics, Inc.

TRADEMARK NOTICE: The following are trademarks or registered trademarks of their respective owners, as follows: Microsoft, Windows, Windows 95, Windows NT, Internet Explorer — Microsoft Corporation; Motif — The Open Systems Foundation, Inc.; PostScript — Adobe Systems, Inc.; UNIX — X/Open Company, Limited; X Window System, X11 — Massachusetts Institute of Technology; RISC System/6000 and IBM — International Business Machines Corporation; Sun, Java, JavaBeans — Sun Microsystems, Inc.; JavaScript, Netscape Communicator — Netscape, Inc.; HPGL and PCL — Hewlett Packard Corporation; DEC, VAX, VMS, OpenVMS — Compaq Information Technologies Group, L.P./Hewlett Packard Corporation; Tektronix 4510 Rasterizer — Tektronix, Inc.; IRIX, TIFF — Silicon Graphics, Inc.; SPARCstation — SPARC International, licensed exclusively to Sun Microsystems, Inc.; HyperHelp — Bristol Technology, Inc. Other products and company names mentioned herein are trademarks of their respective owners.

Use of this document is governed by a Visual Numerics Software License Agreement. This document contains confidential and proprietary information. No part of this document may be reproduced or transmitted in any form without the prior written consent of Visual Numerics.

RESTRICTED RIGHTS NOTICE: This documentation is provided with RESTRICTED RIGHTS. Use, duplication or disclosure by the US Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFAR 252.227-7013, and in subparagraphs (a) through (d) of the Commercial Computer software — Restricted Rights clause at FAR 52.227-19, and in similar clauses in the NASA FAR Supplement, when applicable. Contractor/Manufacturer is Visual Numerics, Inc., 2500 Wilcrest Drive, Suite 200, Houston, TX 77042-2759.

IMSL Fortran, C, and Java
Application Development Tools

Contents

Volume I

Introduction	xi
Chapter 1: Basic Statistics	1
Chapter 2: Regression	71
Chapter 3: Correlation	339
Chapter 4: Analysis of Variance	391
Chapter 5: Categorical and Discrete Data Analysis	477
Chapter 6: Nonparametric Statistics	591
Chapter 7: Tests of Goodness of Fit and Randomness	633

Volume 2

Chapter 8: Time Series Analysis and Forecasting	671
Chapter 9: Covariance Structures and Factor Analysis	897
Chapter 10: Discriminant Analysis	979
Chapter 11: Cluster Analysis	1005

Chapter 12: Sampling	1025
Chapter 13: Survival Analysis, Life Testing, and Reliability	1063
Chapter 14: Multidimensional Scaling	1127
Chapter 15: Density and Hazard Estimation	1171
Chapter 16: Line Printer Graphics	1207
Chapter 17: Probability Distribution Functions and Inverses	1241
Chapter 18: Random Number Generation	1305
Chapter 19: Utilities	1409
Chapter 20: Mathematical Support	1477
Reference Material	1499
Appendix A: GAMS Index	A-1
Appendix B: Alphabetical Summary of Routines	B-1
Appendix C: References	C-1
Product Support	i
Index	iii

vertical histogram 1144, 1146
von Mises distribution 1293

W

Weibull distribution 1295
weights 673
Wiener filter coefficients 774
Wiener forecast function 677
Wiener forecast operator 749
Wilcoxon rank sum test 609
Wilcoxon signed rank test 595
Wilcoxon two-sample test 618
Wilks' lambda 79, 186

Introduction

The IMSL Fortran Libraries

The IMSL Fortran Libraries consist of two separate, but coordinated Libraries that allow easy user access. These Libraries are organized as follows:

- MATH/LIBRARY general applied mathematics and special functions

The User's Guide for IMSL MATH/LIBRARY has two parts:

1. MATH/LIBRARY (Volumes 1 and 2)
2. MATH/LIBRARY Special Functions

- STAT/LIBRARY statistics

Most of the routines are available in both single and double precision versions. The same user interface is found on the many hardware versions that span the range from personal computer to supercomputer. Note that some IMSL routines are not distributed for FORTRAN compiler environments that do not support double precision complex data. The names of the IMSL routines that return or accept the type double complex begin with the letter "z" and, occasionally, "DC."

Getting Started

The IMSL STAT/LIBRARY is a collection of FORTRAN subroutines and functions useful in research and statistical analysis. Each routine is designed and documented to be used in research activities as well as by technical specialists.

To use any of these routines, you must write a program in FORTRAN (or possibly some other language) to call the STAT/LIBRARY routine. Each routine conforms to established conventions in programming and documentation. We give first priority in development to efficient algorithms, clear documentation, and accurate results. The uniform design of the routines makes it easy to use more than one routine in a given application. Also, you will find that the design consistency enables you to apply your experience with one STAT/LIBRARY routine to all other IMSL routines that you use.

Finding the Right Routine

The STAT/LIBRARY is organized into chapters; each chapter contains routines with similar computational or analytical capabilities. To locate the right routine for a given problem, you may use either the table of contents located in each chapter introduction, or one of the indexes at the

end of this manual. GAMS index uses GAMS classification (Boisvert, R.F., S.E. Howe, D.K. Kahaner, and J.L. Springmann 1990, *Guide to Available Mathematical Software*, National Institute of Standards and Technology NISTIR 90-4237). Use the GAMS index to locate which STAT/LIBRARY routines pertain to a particular topic or problem.

Often the quickest way to use the STAT/LIBRARY is to find an example similar to your problem and then to mimic the example. Each routine document has at least one example demonstrating its application. The example for a routine may be created simply for illustration, it may be from a textbook (with reference to the source) or it may be from the statistical literature, in which case IMSL routine GDATA retrieves the data set.

Organization of the Documentation

This manual contains a concise description of each routine, with at least one demonstrated example of each routine, including sample input and results. You will find all information pertaining to the STAT/LIBRARY in this manual. Moreover, all information pertaining to a particular routine is in one place within a chapter.

Each chapter begins with an introduction followed by a table of contents that lists the routines included in the chapter. Documentation of the routines consists of the following information:

- IMSL Routine's Generic Name
- Purpose: a statement of the purpose of the routine. If the routine is a function rather than a subroutine the purpose statement will reflect this fact.
- Function Return Value: a description of the return value (for functions only).
- Required Arguments: a description of the required arguments in the order of their occurrence. Input arguments usually occur first, followed by input/output arguments, with output arguments described last. Furthermore, the following terms apply to arguments:
 - **Input** Argument must be initialized; it is not changed by the routine.
 - **Input/Output** Argument must be initialized; the routine returns output through this argument; cannot be a constant or an expression.
 - **Input or Output** Select appropriate option to define the argument as either input or output. See individual routines for further instructions.
 - **Output** No initialization is necessary; cannot be a constant or an expression. The routine returns output through this argument.
- Optional Arguments: a description of the optional arguments in the order of their occurrence.
- Fortran 90 Interface: a section that describes the generic and specific interfaces to the routine.
- Fortran 77 Style Interfaces: an optional section, which describes Fortran 77 style interfaces, is supplied for backwards compatibility with previous versions of the Library.
- Example: at least one application of this routine showing input and required dimension and type statements.
- Output: results from the example(s).

- Comments: details pertaining to code usage.
- Description: a description of the algorithm and references to detailed information. In many cases, other IMSL routines with similar or complementary functions are noted.
- Programming notes: an optional section that contains programming details not covered elsewhere.
- References: periodicals and books with details of algorithm development.
- Additional Examples: an optional section with additional applications of this routine showing input and required dimension and type statements.

Naming Conventions

The names of the routines are mnemonic and unique. Most routines are available in both a single precision and a double precision version, with names of the two versions sharing a common root. The root name is also the generic interface name. The name of the double precision specific version begins with a “D_.” The single precision specific version begins with an “S_.” For example, the following pairs are precision specific names of routines in the two different precisions: S_UVSTA/D_UVSTA (the root is “UVSTA,” for “Basic Univariate Statistics”) and S_TWFRQ/D_TWFRQ (the root is “TWFRQ,” for “Two-Way Frequency Table”). Of course the generic name can be used as an entry point for all precisions supported.

Except when expressly stated otherwise, the names of the variables in the argument lists follow the FORTRAN default type for integer and floating point. In other words, a variable whose name begins with one of the letters “I” through “N” is of type INTEGER, and otherwise is of type REAL or DOUBLE PRECISION, depending on the precision of the routine.

An assumed size array with more than one dimension that is used as a FORTRAN argument can have an assumed-size declarator for the last dimension only. In the MATH/LIBRARY routines, the information about the first dimension is passed by a variable with the prefix “LD” and with the array name as the root. For example, the argument LDA contains the leading dimension of array A. In most cases, information about the dimensions of arrays is obtained from the array through the use of Fortran 90’s *size* function. Therefore, arguments carrying this type of information are usually defined as optional arguments.

Where appropriate, the same variable name is used consistently throughout a chapter in the STAT/LIBRARY. For example, in the routines for random number generation, NR denotes the number of random numbers to be generated, and R or IR denotes the array that stores the numbers.

When writing programs accessing the STAT/LIBRARY, the user should choose FORTRAN names that do not conflict with names of IMSL subroutines, functions, or named common blocks. The careful user can avoid any conflicts with IMSL names if, in choosing names, the following rules are observed:

- Do not choose a name that appears in the Alphabetical Summary of Routines, **at the end of the *User’s Manual***, nor one of these names preceded by a D_, S_, D_, C_, or Z_.
- Do not choose a name consisting of more than three characters with a numeral in the second or third position.

For further details, see the section on “Reserved Names” in the Reference Material section of this manual.

Using Library Subprograms

The documentation for the routines uses the generic name and omits the prefix, and hence the entire suite of routines for that subject is documented under the generic name.

Examples that appear in the documentation also use the generic name. To further illustrate this principle, note the `OWFRQ` documentation (see Chapter 1, Basic Statistics), for tallying observation into a one-way frequency table. A description is provided for just one data type. There are two documented routines in this subject area: `S_OWFRQ` and `D_OWFRQ`.

These routines constitute single-precision and double-precision versions of the code.

The appropriate routine is identified by the Fortran 90 compiler. Use of a module is required with the routines. The naming convention for modules joins the suffix “_int” to the generic routine name. Thus, the line “use `OWFRQ_INT`” is inserted near the top of any routine that calls the subprogram “`OWFRQ`”. More inclusive modules are also available. For example, the module named “`imsl_libraries`” contains the interface modules for all routines in the library.

Programming Conventions

In general, the IMSL STAT/LIBRARY codes are written so that computations are not affected by underflow, provided the system (hardware or software) places a zero value in the register. In this case, system error messages indicating underflow should be ignored.

IMSL codes also are written to avoid overflow. A program that produces system error messages indicating overflow should be examined for programming errors such as incorrect input data, mismatch of argument types, or improper dimensioning.

In many cases, the documentation for a routine points out common pitfalls that can lead to failure of the algorithm.

Library routines detect error conditions, classify them as to severity, and treat them accordingly. This error-handling capability provides automatic protection for the user without requiring the user to make any specific provisions for the treatment of error conditions. See the section on “User Errors” in the Reference Material for further details.

Module Usage

Users are required to incorporate a “use” statement near the top of their program for the IMSL routine being called when writing new code that uses this library. However, legacy code which calls routines in the previous version of the library without the use of a “use” statement will continue to work as before. Also, code which employed the “use `numerical_libraries`” statement from the previous version of the library will continue to work properly with this version of the library.

Users wishing to update existing programs so as to call other routines from this library should incorporate a use statement for the specific new routine being called. (Here, the term “new routine” implies any routine in the library, only “new” to the user’s program.) Use of the more encompassing “`imsl_libraries`” module in this case could result in argument mismatches for the “old” routine(s) being called. (This would be caught by the compiler.)

Users wishing to update existing programs so as to call the new generic versions of the routines must change their calls to the existing routines so as to match the new calling sequences and use either the routine specific interface modules or the all encompassing “`imsl_libraries`” module.

Programming Tips

It is strongly suggested that users force all program variables to be explicitly typed. This is done by including the line “`IMPLICIT NONE`” as close to the first line as possible. Study some of the examples accompanying an IMSL Fortran Library routine early on. These examples are available online as part of the product.

Each subject routine called or otherwise referenced requires the “`use`” statement for an interface block designed for that subject routine. The contents of this interface block are the interfaces to the separate routines available for that subject. Packaged descriptive names for option numbers that modify documented optional data or internal parameters might also be provided in the interface block. Although this seems like an additional complication, many typographical errors are avoided at an early stage in development through the use of these interface blocks. The “`use`” statement is required for each routine called in the user’s program.

However, if one is only using the Fortran 77 interfaces supplied for backwards compatibility then the “`use`” statements are not required.

Optional Subprogram Arguments

IMSL Fortran Library routines have *required* arguments and may have *optional* arguments. All arguments are documented for each routine. For example, consider the routine `ORDST` that determines order statistics. The required arguments are `X`, `NOS`, `OS`, and `NMISS`. The input data for the problem are the `X` array and `NOS`, the number of order statistics; the output is returned in the `OS` array. The number of missing values is returned in `NMISS`. This routine has as optional arguments `NOBS`, `IOPT`, and `IOS`. If one wishes to calculate a different set of order statistics than the default (the first `NOS` order statistics) then the optional argument given by the “`IOPT=`” keyword should be used in the argument list. See Example 2 in Chapter 1, “Basic Statistics” of `ORDST` for an example of this functionality.

For compatibility with previous versions of the IMSL Libraries, the `NUMERICAL_LIBRARIES` interface module includes backwards compatible positional argument interfaces to all routines which existed in the Fortran 77 version of the Library. Note that it is not necessary to use “`use`” statements when calling these routines by themselves. Existing programs which called these routines will continue to work in the same manner as before.

Error Handling

The routines in the IMSL STAT/LIBRARY attempt to detect and report errors and invalid input. Errors are classified and are assigned a code number. By default, errors of moderate or worse severity result in messages being automatically printed by the routine. Moreover, errors of worse severity cause program execution to stop. The severity level as well as the general nature of the error is designated by an “error type” with numbers from 0 to 5. An error type 0 is no error; types 1 through 5 are progressively more severe. In most cases, you need not be concerned with our method of handling errors. For those interested, a complete description of the error-handling

system is given in the Reference Material, which also describes how you can change the default actions and access the error code numbers.

Printing Results

Several routines in the IMSL STAT/LIBRARY have an option for printing results. These routines have an optional argument, `IPRINT`, to control the printing. In any routine that allows printing, if `IPRINT = 0`, (the default) then no printing is done (except possibly error messages). Some routines allow various amounts of printing; one value of `IPRINT` might result in printing only summary statistics, while another value might cause more detailed statistics or intermediate results to be printed. Other routines in the STAT/LIBRARY do not print any of the results. In all routines, of course, the output is returned in FORTRAN variables, so if the routine does not do printing, or if you use the default `IPRINT` value, you can print the results yourself. The STAT/LIBRARY contains some special routines just for printing arrays. For example, `WRRRN` and `WRRRL` are two convenient routines for printing matrices. See Chapter 19, "Utilities," for detailed descriptions of these routines.

A commonly used routine in the examples is the IMSL routine `UMACH`, which retrieves the FORTRAN device unit number for printing the results. Because this routine obtains device unit numbers, it can be used to redirect the input or output. The section on "Machine- Dependent Constants" in the Reference Material contains a description of the routine `UMACH`.

Missing Values

Many of the routines in the IMSL STAT/LIBRARY allow the data to contain missing values. These routines recognize as a missing value the special value referred to as 'not a number,' or NaN. The actual value is different on different computers, but it can be obtained by reference to the IMSL routines `AMACH` or `DMACH`, described in the "Machine-Dependent Constants" section of the Reference Material. In routines that allow missing values, two common arguments are `NMISS` and `NRMISS`. The definitions of these arguments vary somewhat depending on the specific routine. However, in a data structure where the rows represent observations and the columns represent variables, `NRMISS` is the number of rows containing missing values and `NMISS` is the total number of missing values.

The way that missing values are treated depends on the individual routine, and is described in the documentation for the routine.

Routines that Accumulate Results over Several Calls

Often in statistical analyses, not all of the data are available in computer memory at once. Many of the routines in the STAT/LIBRARY accept a part of the data, accumulate some statistics, and continue accepting data and accumulating statistics until all of the data have been processed. The routines that allow the data to be processed a little at a time have an argument called "ID0."

Using IMSL Fortran Library on Shared-Memory Multiprocessors

The IMSL Fortran Library allows users to leverage the high-performance technology of shared memory parallelism (SMP) when their environment supports it. Support for SMP systems within the IMSL Library is delivered through various means, depending upon the availability of technologies such as OpenMP, high performance BLAS, and hardware-specific IMSL algorithms. Use of the IMSL Fortran Library on SMP systems can be achieved by using the appropriate link environment variable when building your application. Details on the available link environment variables for your installation of the IMSL Fortran Library can be found in the online README file of the product distribution.

This introduction has acquainted you with a few general characteristics of IMSL STAT/LIBRARY. If you are using the STAT/LIBRARY at a computer center, the computer center consultant will provide the details necessary to use the IMSL routines on your computer system.

Also, additional general information for all users is available in the Reference Material at the end of this manual.

Chapter 1: Basic Statistics

Routines

1.1. Frequency Tabulations		
One-way frequency table.....	OWFRQ	3
Two-way frequency table.....	TWFRQ	7
Frequencies in multivariate data.....	FREQ	14
1.2. Univariate Summary Statistics		
Moments and inferences for normal distribution.....	UVSTA	18
1.3. Ranks and Order Statistics		
Numerical ranking.....	RANKS	26
Letter value summary.....	LETTR	32
Order statistics.....	ORDST	34
Empirical quantiles.....	EQTIL	38
1.4. Parametric Estimates and Tests (See also Univariate Summary Statistics)		
Two-sample t tests and F tests.....	TWOMV	41
Estimate the parameter in a binomial distribution.....	BINES	48
Estimate the parameter in a Poisson distribution.....	POIES	50
Estimation in censored normal data.....	NRCES	53
1.5. Grouped Data		
Statistics for grouped data.....	GRPES	56
1.6. Continuous Data in a Table		
Compute cell means and sums of squares.....	CSTAT	60
Median polish of a two-way table.....	MEDPL	66

Usage Notes

Frequency Tabulations

The routines for frequency tabulations accept raw data in the form of vectors or matrices and produce counts. Two of these routines assume generally that the data are continuous and tally the observations into groups based on grouping information that the user supplies. Another routine for

frequency tabulations assumes basically that the data are discrete and counts the number of observations with each value. Other analyses of discrete data or count data can be performed using IMSL routines in Chapter 5, “Categorical and Discrete Data Analysis.”

Univariate Summary Statistics

The routine `UVSTA` (page 18) computes the sample mean, variance, minimum, maximum, and other basic statistics for each variable in a data set. It also computes confidence intervals for the mean and variance if the sample is assumed to be from a normal distribution.

Ranks and Order Statistics

The routines for ranks and order statistics accept data from a single sample stored in a vector. Ranks, order statistics, and sample quantiles form the basis for many nonparametric and robust statistical techniques (see Conover 1980 and Hoaglin et al. 1983). Letter values, computed by the routine `LETR` (page 32), are a special set of order statistics particularly useful in exploratory data analysis (see Hoaglin 1983).

Parametric Estimates and Tests

The routines described in this section compute statistics for simple inferences about the parameters in normal, binomial, and Poisson distributions. General discussions of estimation techniques for these distributions can be found in Johnson and Kotz (1969, 1970a, 1970b). The routine `UVSTA` (page 18), for univariate summary statistics, also computes statistics for simple inferences about the parameters in a single normal distribution.

Grouped Data

The routine `GRPES` (page 56) computes several basic statistics, such as arithmetic means, geometric means, harmonic means, and moments about the arithmetic mean for grouped data. The second, third, and fourth moments are computed both with and without Sheppard’s corrections.

Continuous Data in a Table

The routine `CSTAT` (page 60) accepts data sets with both classification variables and response variables. The classification variables define cells in a table. Within each cell, means and sums of squares are computed for the response variables. Further analysis of the response variables, in particular, assessment of the effects of the classification variables, may be performed using the routines described in Chapter 4 on analysis of variance. An alternative for two-way tables is median polish, which is more resistant to outliers, but which is more exploratory. That is, no test is performed to confirm statistically that row and/or column effects are present. The routine `MEDPL` (page 66) in this section performs median polish. (See Tukey, 1977; Velleman and Hoaglin, 1981; and Emerson and Hoaglin, 1983.) For count data (frequencies), the routines described in Chapter 5, “Categorical and Discrete Data Analysis,” are appropriate for determining the amount of association among the rows and columns.

OWFRQ

Tallies observations into a one-way frequency table.

Required Arguments

X — Vector of length *NOBS* containing the data. (Input)

K — Number of intervals. (Input)

TABLE — Vector of length *K* containing the counts. (Output)

Optional Arguments

NOBS — Number of observations. (Input)

Default: *NOBS* = size(*X*,1).

IOPT — Tallying option. (Input)

Default: *IOPT* = 0.

IOPT Action

- 0 Intervals of equal length, determined from the data, are used. Let *XMIN* and *XMAX* be the minimum and maximum values in *X*, respectively. Then, *TABLE*(1) is the tally of observations less than or equal to $XMIN + (XMAX - XMIN)/K$, *TABLE*(2) is the tally of observations greater than $XMIN + (XMAX - XMIN)/K$ and less than or equal to $XMIN + 2 * (XMAX - XMIN)/K$, and so on. *TABLE*(*K*) is the tally of observations greater than $XMAX - (XMAX - XMIN)/K$.
- 1 Intervals of equal length are used just as in the case of *IOPT* = 0, except the upper and lower bounds are taken as the user supplied variables *XLO* and *XHI*, instead of the actual minimum and maximum in the data. Therefore, the first and the last intervals are semi-infinite in length. *K* must be greater than 2.
- 2 *K* - 1 cutpoints are input in *DIV*. The tally in *TABLE*(1) is the number of observations less than or equal to *DIV*(1). For *I* greater than 1 and less than *K*, the tally in *TABLE*(*I*) is the number of observations greater than *DIV*(*I* - 1) and less than or equal to *DIV*(*I*). The tally in *TABLE*(*K*) is the number of observations greater than *DIV*(*K* - 1). *K* must be greater than 1.
- 3 Class marks are input in *DIV* and a constant class half-width is input in *CLHW*. The total of the elements in *TABLE* may be less than *NOBS*. The tally in *TABLE*(*I*) is the number of observations between *DIV*(*I*) - *CLHW* and *DIV*(*I*) + *CLHW*.

XLO — If `IOPT = 1`, `XLO` is the lower bound at which to begin forming the class intervals. (Input)

`XLO` is used only if `IOPT = 1`.

XHI — If `IOPT = 1`, `XHI` is the upper bound to use in forming the class intervals. (Input)

`XHI` is used only if `IOPT = 1`.

CLHW — If `IOPT = 3`, `CLHW` is the half-width of the class intervals. (Input)

`CLHW` is not used if `IOPT` is not equal to 3.

DIV — Vector of varying length and contents depending on `IOPT`. (Input if `IOPT = 2` or `3`; output if `IOPT = 0` or `1`.)

The contents of `DIV` are in ascending order.

IOPT Contents

0 `DIV` is of length `K` containing interval midpoints. (`DIV` is output.)

1 `DIV` is of length `K` containing interval midpoints. Since the first and last intervals are semi-infinite in length, `DIV(1)` contains `XLO` minus half the interval length, and `DIV(K)` contains `XHI` plus half the interval length. (`DIV` is output.)

2 `DIV` is a vector of length `K - 1` containing cutpoints. (`DIV` is input.)

3 `DIV` is of length `K` containing classmarks. (`DIV` is input.)

FORTRAN 90 Interface

Generic: `CALL OWFRQ (X, K, TABLE [, ...])`

Specific: The specific interface names are `S_OWFRQ` and `D_OWFRQ`.

FORTRAN 77 Interface

Single: `CALL OWFRQ (NOBS, X, K, IOPT, XLO, XHI, CLHW, DIV, TABLE)`

Double: The double precision name is `DOWFRQ`.

Example 1

The data for these examples are from Hinkley (1977) and Velleman and Hoaglin (1981). They are the measurements (in inches) of precipitation in Minneapolis/St. Paul during the month of March for 30 consecutive years. In the first example, we set `IOPT = 0`. This option may be appropriate if we do not know the range of the data. Notice that the midpoints of the class intervals, output in `DIV`, are not “pretty” numbers.

```

USE OWFRQ_INT
USE UMACH_INT
INTEGER K, NOBS
PARAMETER (K=10, NOBS=30)
!
INTEGER NOUT
REAL DIV(K), TABLE(K), X(NOBS)
!
DATA X/0.77, 1.74, 0.81, 1.20, 1.95, 1.20, 0.47, 1.43, 3.37, &
      2.20, 3.00, 3.09, 1.51, 2.10, 0.52, 1.62, 1.31, 0.32, 0.59, &
      0.81, 2.81, 1.87, 1.18, 1.35, 4.75, 2.48, 0.96, 1.89, 0.90, &
      2.05/
!
CALL UMACH (2, NOUT)
!
CALL OWFRQ (X, K, TABLE, DIV=DIV)
WRITE (NOUT,99999) DIV, TABLE
99999 FORMAT (' Midpoints: ', 10F5.2, '/', ' Counts: ', 10F5.0)
END

```

Output

```

Midpoints:  0.54 0.98 1.43 1.87 2.31 2.76 3.20 3.64 4.09 4.53
Counts:    4.   8.   5.   5.   3.   1.   3.   0.   0.   1.

```

Description

The routine `OWFRQ` groups numerical data into categories, which can be defined in any of four different ways as chosen by `IOPT`. If `IOPT = 0`, `K` intervals of equal length are formed between the minimum and maximum values in the data, and then the data are tallied in these intervals. The midpoints of the intervals are output in `DIV`.

If `IOPT = 1`, `K - 2` intervals of equal length are formed between `XLO` and `XHI`, and then the data are tallied in these intervals. In this option, there is one group that consists of data less than `XLO` and one group of data greater than `XHI`. This option is similar to `IOPT = 0`, except with this option, the midpoints of the classes are under control of the user. The midpoints of the intervals are output in `DIV`. The first and last values of `DIV`, respectively, contain `XLO` minus half the class width and `XHI` plus half the class width.

For `IOPT = 2` or `3`, the intervals need not be equally spaced. If `IOPT = 2`, the intervals need not be equal in length. In this case, the intervals are defined by their boundaries, the “cutpoints”, which are input in `DIV`. The number of cutpoints is one less than the number of intervals. The first cutpoint defines the upper bound of the first interval, and the last cutpoint defines the lower bound of the last interval.

If `IOPT = 3`, the intervals are all of length twice `CLHW`, and they are centered on the class marks input in `DIV`. This option can be used to exclude portions of the data.

The examples use all of these options with the same data set.

Additional Examples

Example 2

In this example, we set `IOPT = 1` and choose `XLO` and `XHI` so that the intervals will be 0.0 to 0.5, 0.5 to 1.0, and so on. This means that the midpoints of the class intervals, output in `DIV`, will be 0.25, 0.75, and so on.

```
USE OWFRQ_INT
USE UMACH_INT
INTEGER K, NOBS
PARAMETER (K=10, NOBS=30)
!
INTEGER IOPT, NOUT
REAL DIV(K), TABLE(K), X(NOBS), XHI, XLO
!
DATA X/0.77, 1.74, 0.81, 1.20, 1.95, 1.20, 0.47, 1.43, 3.37, &
      2.20, 3.00, 3.09, 1.51, 2.10, 0.52, 1.62, 1.31, 0.32, 0.59, &
      0.81, 2.81, 1.87, 1.18, 1.35, 4.75, 2.48, 0.96, 1.89, 0.90, &
      2.05/
!
CALL UMACH (2, NOUT)
IOPT = 1
XLO = 0.5
XHI = 4.5
!
CALL OWFRQ (X, K, TABLE, IOPT=IOPT, XLO=XLO, XHI=XHI, DIV=DIV)
WRITE (NOUT,99999) DIV, TABLE
99999 FORMAT (' Midpoints: ', 10F5.2, '/', ' Counts: ', 10F5.0)
END
```

Output

```
Midpoints:  0.25 0.75 1.25 1.75 2.25 2.75 3.25 3.75 4.25 4.75
Counts:     2.   7.   6.   6.   4.   2.   2.   0.   0.   1.
```

Example 3

In this example, we input class boundaries in `DIV`. We choose the same intervals as in the example above: 0.0 to 0.5, 0.5 to 1.0, and so on. `DIV` begins with the first cutpoint *between* classes.

```
USE OWFRQ_INT
USE UMACH_INT
INTEGER K, NOBS
PARAMETER (K=10, NOBS=30)
!
INTEGER IOPT, NOUT
REAL DIV(K-1), TABLE(K), X(NOBS)
!
DATA X/0.77, 1.74, 0.81, 1.20, 1.95, 1.20, 0.47, 1.43, 3.37, &
      2.20, 3.00, 3.09, 1.51, 2.10, 0.52, 1.62, 1.31, 0.32, 0.59, &
      0.81, 2.81, 1.87, 1.18, 1.35, 4.75, 2.48, 0.96, 1.89, 0.90, &
      2.05/
DATA DIV/0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5/
```

```

!
  CALL UMACH (2, NOUT)
  IOPT = 2
!
  CALL OWFRQ (X, K, TABLE, IOPT=IOPT, DIV=DIV)
  WRITE (NOUT,99999) DIV, TABLE
99999 FORMAT ('  Cutpoints:      ', 9F5.1, /, '      Counts: ', 10F5.0)
  END

```

Output

```

Cutpoints:      0.5  1.0  1.5  2.0  2.5  3.0  3.5  4.0  4.5
Counts:         2.   7.   6.   6.   4.   2.   2.   0.   0.   1.

```

Example 4

In this example, we set `IOPT = 3`, and set the values in `DIV` and `CLHW` so that the intervals will be the same as in the previous two examples.

```

  USE OWFRQ_INT
  USE UMACH_INT
  INTEGER      K, NOBS
  PARAMETER   (K=10, NOBS=30)
!
  INTEGER      IOPT, NOUT
  REAL         CLHW, DIV(K), TABLE(K), X(NOBS)
!
  DATA X/0.77, 1.74, 0.81, 1.20, 1.95, 1.20, 0.47, 1.43, 3.37, &
        2.20, 3.00, 3.09, 1.51, 2.10, 0.52, 1.62, 1.31, 0.32, 0.59, &
        0.81, 2.81, 1.87, 1.18, 1.35, 4.75, 2.48, 0.96, 1.89, 0.90, &
        2.05/
  DATA DIV/0.25, 0.75, 1.25, 1.75, 2.25, 2.75, 3.25, 3.75, 4.25, &
        4.75/
!
  CALL UMACH (2, NOUT)
  IOPT = 3
  CLHW = 0.25
!
  CALL OWFRQ (X, K, TABLE, IOPT=IOPT, CLHW=CLHW, DIV=DIV)
  WRITE (NOUT,99999) DIV, TABLE
99999 FORMAT ('  Class marks: ', 10F5.2, /, '      Counts: ', 10F5.0)
  END

```

Output

```

Class marks:    0.25  0.75  1.25  1.75  2.25  2.75  3.25  3.75  4.25  4.75
Counts:         2.   7.   6.   6.   4.   2.   2.   0.   0.   1.

```

TWFRQ

Tallies observations into a two-way frequency table.

Required Arguments

X— Vector of length `NOBS` containing the data for one variable. (Input)

Y — Vector of length *NOBS* containing the data for the other variable. (Input)

KX — Number of intervals for the variable *X*. (Input)

KY — Number of intervals for the variable *Y*. (Input)

TABLE — *KX* by *KY* matrix containing the counts. (Output)

Optional Arguments

NOBS — Number of observations. (Input)

Default: *NOBS* = size (*X*,1).

IOPT — Tallying option. (Input)

Default: *IOPT* = 0.

***IOPT* Action**

0 Intervals of equal lengths for each variable, determined from the data, are used. Let *XMIN* and *XMAX* be the minimum and maximum values in *X*, respectively, with similar meanings for *YMIN* and *YMAX*. Then, *TABLE*(1, 1) is the tally of observations with the *X* value less than or equal to $XMIN + (XMAX - XMIN)/KX$, and the *Y* value less than or equal to $YMIN + (YMAX - YMIN)/KY$. The other table entries are determined similarly.

1 Intervals of equal lengths are used just as in the case of *IOPT* = 0, except the upper and lower bounds are taken as the user-supplied variables *XL0*, *XHI*, *YLO*, and *YHI* instead of the actual minima and maxima in the data. Therefore, the first and the last intervals for both variables are semi-infinite in length. *KX* and *KY* must be greater than 2.

2 *KX* - 1 cutpoints are input in *DIVX*, and *KY* - 1 cutpoints are input in *DIVY*. The tally in *TABLE*(1, 1) is the number of observations for which the *X* value is less than or equal to *DIVX*(1), and the *Y* value is less than or equal to *DIVY*(1). For *I* greater than 1 and less than *KX* and *J* greater than 1 and less than *KY*, the tally in *TABLE*(*I*, *J*) is the number of observations with *X* greater than *DIVX*(*I* - 1) and less than or equal to *DIVX*(*I*) and with *Y* greater than *DIVY*(*J* - 1) and less than or equal to *DIVY*(*J*). The tally in *TABLE*(*KX*, *KY*) is the number of observations for which the *X* value is greater than *DIVX*(*KX* - 1) and the *Y* value is greater than *DIVY*(*KY* - 1). *KX* and *KY* must be greater than 1.

3 Class marks are input in *DIVX* and *DIVY* and a constant class half-width are input in *CLHWX* and *CLHWY*. The total of the elements in *TABLE* may be less than *NOBS*. The tally in *TABLE*(*I*, *J*) is the number of observations with *X* value between $DIVX(I) - CLHWX$ and $DIVX(I) + CLHWX$, and with *Y* value between $DIVY(J) - CLHWY$ and $DIVY(J) + CLHWY$.

XLO — If $IOPT = 1$, XLO is the lower bound at which to begin forming the class intervals for X . (Input)
 XLO is only used if $IOPT = 1$.

YLO — If $IOPT = 1$, YLO is the lower bound at which to begin forming the class intervals for Y . (Input)
 YLO is only used if $IOPT = 1$.

XHI — If $IOPT = 1$, XHI is the upper bound to use in forming the class intervals for X . (Input)
 XHI is only used if $IOPT = 1$.

YHI — If $IOPT = 1$, is the upper bound to use in forming the class intervals for Y . (Input)
 YHI is only used if $IOPT = 1$.

CLHWX — If $IOPT = 3$, $CLHWX$ is the half-width of the class intervals for X . (Input)
 $CLHWX$ is only used if $IOPT = 3$.

CLHWY — If $IOPT = 3$, $CLHWY$ is the half-width of the class intervals for Y . (Input)
 $CLHWY$ is only used if $IOPT = 3$.

DIVX — Vector of varying length and contents depending on $IOPT$. (Input if $IOPT = 2$ or 3 ; output if $IOPT = 0$ or 1)
The contents of $DIVX$ are in ascending order.

***IOPT* Contents**

- 0 DIV is of length KX containing interval midpoints for the X variable. ($DIVX$ is output.)
- 1 DIV is of length KX containing interval midpoints for the X variable. Since the first and last intervals are semi-infinite in length, $DIVX(1)$ contains $XLO - \text{half the interval length}$, and $DIVX(KX)$ contains $XHI + \text{half the interval length}$. ($DIVX$ is output.)
- 2 $DIVX$ is a vector of length $KX - 1$ containing cutpoints. ($DIVX$ is input.)
- 3 $DIVX$ is of length KX containing classmarks. ($DIVX$ is input.)

DIVY — Vector of varying length and contents depending on $IOPT$. (Input if $IOPT = 2$ or 3 ; output if $IOPT = 0$ or 1)
The contents of $DIVY$ are in ascending order. See $DIVX$.

LDTABL — Leading dimension of $TABLE$ exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDTABL = \text{size}(TABLE, 1)$.

FORTRAN 90 Interface

Generic: CALL TWFRQ (X, Y, KX, KY, TABLE [,...])

Specific: The specific interface names are S_TWFRQ and D_TWFRQ.

FORTRAN 77 Interface

Single: CALL TWFRQ (NOBS, X, Y, KX, KY, IOPT, XLO, YLO, XHI, YHI,
 CLHWX, CLHWY, DIVX, DIVY, TABLE, LDTABL)

Double: The double precision name is DTWFRQ.

Example 1

The data for X in these examples are the same as those used in the routine for one-way frequency tabulation, OWFRQ (page 3). The data for Y were created by adding small integers to the data in X. In the first example, we set IOPT = 0. This option may be appropriate if we do not know the range of the data. Notice that the midpoints of the class intervals, output in DIVX and DIVY, are not “pretty” numbers. Routine WRRRN (see Chapter 19, Utilities) is used to print the frequencies. This printing routine puts column and row numbers above and to the left of the matrix being printed. For example, the “4” in the second row and second column of the output is the first number that represents a frequency. That frequency is the number of occurrences of pairs of observations in which both values are in the lowest groups.

```
USE TWFRQ_INT
USE UMACH_INT
USE WRRRN_INT

INTEGER      KX, KY, LDTABL, NOBS
PARAMETER   (KX=5, KY=6, LDTABL=5, NOBS=30)
!
INTEGER      NOUT
REAL         DIVX(KX), DIVY(KY), TABLE(LDTABL,KY), X(NOBS), Y(NOBS)
!
DATA X/0.77, 1.74, 0.81, 1.20, 1.95, 1.20, 0.47, 1.43, 3.37, &
      2.20, 3.00, 3.09, 1.51, 2.10, 0.52, 1.62, 1.31, 0.32, 0.59, &
      0.81, 2.81, 1.87, 1.18, 1.35, 4.75, 2.48, 0.96, 1.89, 0.90, &
      2.05/
DATA Y/1.77, 3.74, 3.81, 2.20, 3.95, 4.20, 1.47, 3.43, 6.37, &
      3.20, 5.00, 6.09, 2.51, 4.10, 3.52, 2.62, 3.31, 3.32, 1.59, &
      2.81, 5.81, 2.87, 3.18, 4.35, 5.75, 4.48, 3.96, 2.89, 2.90, &
      5.05/
!
CALL UMACH (2, NOUT)
!
CALL TWFRQ (X, Y, KX, KY, TABLE, DIVX=DIVX, DIVY=DIVY)
WRITE (NOUT,99999) DIVX, DIVY
99999 FORMAT (' Midpoints for X (Rows):      ', 5F5.2, '/', ' Midpoints ' &
             ', 'for Y (Columns): ', 6F5.2)
CALL WRRRN ('Frequencies', TABLE)
END
```


Output

Midpoints for X (Rows): 0.76 1.65 2.53 3.42 4.31
Midpoints for Y (Columns): 1.88 2.69 3.51 4.33 5.14 5.96

	Frequencies					
	1	2	3	4	5	6
1	4.000	2.000	4.000	2.000	0.000	0.000
2	0.000	4.000	3.000	2.000	1.000	0.000
3	0.000	0.000	1.000	2.000	0.000	1.000
4	0.000	0.000	0.000	0.000	1.000	2.000
5	0.000	0.000	0.000	0.000	0.000	1.000

Description

The routine `TWFRQ` groups bivariate numerical data into categories, which can be defined in any of four different ways as chosen by `IOPT`. This routine is very similar to routine `OWFRQ` (page 3) for univariate data. If `IOPT=0`, `KX` intervals of equal length are formed for the first variable (in `X`) between the minimum and maximum values in `X` and similarly `KY` intervals are formed for the second variable (in `Y`). The data are then tallied in these intervals. The midpoints of the intervals for the first variable are output in `DIVX` and those of the second in `DIVY`.

If `IOPT=1`, `K-2` intervals of equal length are formed between `XLO` and `XHI` for the data in `X` and likewise for `Y`. The data are then tallied in these intervals. In this option, there is one group that consists of data less than `XLO` and one group of data greater than `XHI`. This option is similar to `IOPT=0`, except in this case, the midpoints of the classes are under control of the user. The midpoints of the intervals are output in `DIVX` and `DIVY`.

For `IOPT=2` or `3`, the intervals need not be equally spaced. If `IOPT=2`, the intervals need not be equal in length. In this case, the intervals are defined by their boundaries, the “cutpoints”, which are input in `DIVX` and `DIVY`. The number of cutpoints is one less than the number of intervals. The first cutpoint defines the upper bound of the first interval, and the last cutpoint defines the lower bound of the last interval.

If `IOPT=3`, the intervals are all of length twice `CLHWX` for `X` and twice `CLHWY` for `Y`, and they are centered on the class marks input in `DIVX` and `DIVY`. This option can be used to exclude portions of the data. The examples use all of these options with the same data set.

Additional Examples

Example 2

In this example, we set `IOPT=1` and choose `XLO`, `XHI`, `YLO`, and `YHI` so that the intervals will be 0 to 1, 1 to 2, and so on for `X`, and 1 to 2, 2 to 3, and so on for `Y`. This means that the midpoints of the class intervals, output in `DIVX` and `DIVY`, will be 0.5, 1.5, 2.5, and so on. The “5” in the third row and fourth column of the printed output below, (i.e., the second row and the third column of the frequencies `TABLE`) represents five pairs of observations with the `X` value between 1.0 and 2.0 and the `Y` value between 3.0 and 4.0.

```
USE TWFRQ_INT
USE UMACH_INT
USE WRRRN_INT
```

```

INTEGER    KX, KY, LDTABL, NOBS
PARAMETER  (KX=5, KY=6, LDTABL=5, NOBS=30)
!

INTEGER    IOPT, NOUT
REAL      DIVX(KX), DIVY(KY), TABLE(LDTABL,KY), &
          X(NOBS), XHI, XLO, Y(NOBS), YHI, YLO
!

DATA X/0.77, 1.74, 0.81, 1.20, 1.95, 1.20, 0.47, 1.43, 3.37, &
     2.20, 3.00, 3.09, 1.51, 2.10, 0.52, 1.62, 1.31, 0.32, 0.59, &
     0.81, 2.81, 1.87, 1.18, 1.35, 4.75, 2.48, 0.96, 1.89, 0.90, &
     2.05/
DATA Y/1.77, 3.74, 3.81, 2.20, 3.95, 4.20, 1.47, 3.43, 6.37, &
     3.20, 5.00, 6.09, 2.51, 4.10, 3.52, 2.62, 3.31, 3.32, 1.59, &
     2.81, 5.81, 2.87, 3.18, 4.35, 5.75, 4.48, 3.96, 2.89, 2.90, &
     5.05/
!

CALL UMACH (2, NOUT)
IOPT = 1
XLO = 1.0
XHI = 4.0
YLO = 2.0
YHI = 6.0
!

CALL TWFRQ (X, Y, KX, KY, TABLE, IOPT=IOPT, XLO=XLO, YLO=YLO, &
           XHI=XHI, YHI=YHI, DIVX=DIVX, DIVY=DIVY)
WRITE (NOUT,99999) DIVX, DIVY
99999 FORMAT (' Midpoints for X (Rows):      ', 5F5.2, '/', ' Midpoints ' &
            , 'for Y (Columns): ', 6F5.2)
CALL WRRRN ('Frequencies', TABLE)
END

```

Output

```

Midpoints for X (Rows):      0.50 1.50 2.50 3.50 4.50
Midpoints for Y (Columns):  1.50 2.50 3.50 4.50 5.50 6.50

```

	Frequencies					
	1	2	3	4	5	6
1	3.000	2.000	4.000	0.000	0.000	0.000
2	0.000	5.000	5.000	2.000	0.000	0.000
3	0.000	0.000	1.000	3.000	2.000	0.000
4	0.000	0.000	0.000	0.000	0.000	2.000
5	0.000	0.000	0.000	0.000	1.000	0.000

Example 3

In this example, we input class boundaries in `DIVX` and `DIVY`. We choose the same intervals as in the example above: 0 to 1, 1 to 2, and so on. `DIVX` and `DIVY` begins with the first cutpoint *between* classes.

```

USE TWFRQ_INT
USE UMACH_INT
USE WRRRN_INT

INTEGER    KX, KY, LDTABL, NOBS
PARAMETER  (KX=5, KY=6, LDTABL=5, NOBS=30)

```

```

!
INTEGER      IOPT, NOUT
REAL         DIVX(4), DIVY(5), TABLE(LDTABL,KY), X(NOBS), Y(NOBS)
!
DATA X/0.77, 1.74, 0.81, 1.20, 1.95, 1.20, 0.47, 1.43, 3.37, &
      2.20, 3.00, 3.09, 1.51, 2.10, 0.52, 1.62, 1.31, 0.32, 0.59, &
      0.81, 2.81, 1.87, 1.18, 1.35, 4.75, 2.48, 0.96, 1.89, 0.90, &
      2.05/
DATA Y/1.77, 3.74, 3.81, 2.20, 3.95, 4.20, 1.47, 3.43, 6.37, &
      3.20, 5.00, 6.09, 2.51, 4.10, 3.52, 2.62, 3.31, 3.32, 1.59, &
      2.81, 5.81, 2.87, 3.18, 4.35, 5.75, 4.48, 3.96, 2.89, 2.90, &
      5.05/
DATA DIVX/1.0, 2.0, 3.0, 4.0/
DATA DIVY/2.0, 3.0, 4.0, 5.0, 6.0/
!
CALL UMACH (2, NOUT)
IOPT = 2
!
CALL TWFRQ (X, Y, KX, KY, TABLE, IOPT=IOPT, DIVX=DIVX, DIVY=DIVY)
WRITE (NOUT,99999) DIVX, DIVY
99999 FORMAT ('  Cutpoints for X (Rows):      ', 4F5.2, '/', '  Cutpoints ' &
            , 'for Y (Columns): ', 5F5.2)
CALL WRRRN ('Frequencies', TABLE)
END

```

Output

```

Cutpoints for X (Rows):      1.00 2.00 3.00 4.00
Cutpoints for Y (Columns):  2.00 3.00 4.00 5.00 6.00

```

	Frequencies					
	1	2	3	4	5	6
1	3.000	2.000	4.000	0.000	0.000	0.000
2	0.000	5.000	5.000	2.000	0.000	0.000
3	0.000	0.000	1.000	3.000	2.000	0.000
4	0.000	0.000	0.000	0.000	0.000	2.000
5	0.000	0.000	0.000	0.000	1.000	0.000

Example 4

In this example, we set `IOPT = 3`, and set the values in `DIVX`, `DIVY`, `CLHWX`, and `CLHWY` so that the intervals will be the same as in the previous two examples.

```

USE TWFRQ_INT
USE UMACH_INT
USE WRRRN_INT

INTEGER      KX, KY, LDTABL, NOBS
PARAMETER   (KX=5, KY=6, LDTABL=5, NOBS=30)
!
INTEGER      IOPT, NOUT
REAL         CLHWX, CLHWY, DIVX(KX), DIVY(KY), TABLE(LDTABL,KY), &
            X(NOBS), Y(NOBS)
!
DATA X/0.77, 1.74, 0.81, 1.20, 1.95, 1.20, 0.47, 1.43, 3.37, &
      2.20, 3.00, 3.09, 1.51, 2.10, 0.52, 1.62, 1.31, 0.32, 0.59, &
      0.81, 2.81, 1.87, 1.18, 1.35, 4.75, 2.48, 0.96, 1.89, 0.90, &

```

```

        2.05/
DATA Y/1.77, 3.74, 3.81, 2.20, 3.95, 4.20, 1.47, 3.43, 6.37, &
      3.20, 5.00, 6.09, 2.51, 4.10, 3.52, 2.62, 3.31, 3.32, 1.59, &
      2.81, 5.81, 2.87, 3.18, 4.35, 5.75, 4.48, 3.96, 2.89, 2.90, &
      5.05/
DATA DIVX/0.5, 1.5, 2.5, 3.5, 4.5/
DATA DIVY/1.5, 2.5, 3.5, 4.5, 5.5, 6.5/
!
CALL UMACH (2, NOUT)
IOPT = 3
CLHWX = 0.5
CLHWY = 0.5
!
CALL TWFRQ (X, Y, KX, KY, TABLE, IOPT=IOPT, CLHWX=CLHWX, &
           CLHWY=CLHWY, DIVX=DIVX, DIVY=DIVY)
WRITE (NOUT,99999) DIVX, DIVY
99999 FORMAT (' Class marks for X (Rows): ', 5F5.2, '/', ' Class ', &
           'marks for Y (Columns): ', 6F5.2)
CALL WRRRN ('Frequencies', TABLE)
END

```

Output

```

Class marks for X (Rows):    0.50 1.50 2.50 3.50 4.50
Class marks for Y (Columns): 1.50 2.50 3.50 4.50 5.50 6.50

```

	Frequencies					
	1	2	3	4	5	6
1	3.000	2.000	4.000	0.000	0.000	0.000
2	0.000	5.000	5.000	2.000	0.000	0.000
3	0.000	0.000	1.000	3.000	2.000	0.000
4	0.000	0.000	0.000	0.000	0.000	2.000
5	0.000	0.000	0.000	0.000	1.000	0.000

FREQ

Tallies multivariate observations into a multiway frequency table.

Required Arguments

X — NOBS by NCOL matrix containing the data. (Input)

INDCL — Index vector of length NCLVAR containing the column numbers in X that are the classification variables. (Input)

MAXTAB — An upper bound for the total number of cells in the frequency table. (Input)
This is the product of the number of distinct values taken by all of the classification variables since the table includes the empty cells.

MAXCL — An upper bound for the sum of the number of distinct values taken by all of the classification variables. (Input)

NCLVAL — Vector of length `NCLVAR` containing, in its i -th element, the number of levels or categories of the i -th classification variable. (Output, if `IDO = 1`; Input/Output, if `IDO = 2`.)
Each variable must have more than one level.

CLVAL — Vector of length `NCLVAL(1) + NCLVAL(2) + ... + NCLVAL(NCLVAR)` containing the values of the classification variables. (Output, if `IDO = 1`; input/output, if `IDO = 2`.)
Since in general the length of `CLVAL` will not be known in advance, `MAXCL` is an upper bound for this length. The first `NCLVAL(1)` elements of `CLVAL` contain the values for the first classification variable. The next `NCLVAL(2)` contain the values for the second variable. The last `NCLVAL(NCLVAR)` positions contain the values for the last classification variable.

TABLE — Vector of length `NCLVAL(1) * NCLVAL(2) * ... * NCLVAL(NCLVAR)` containing the frequencies in the cells of the table to be fit. (Output, if `IDO = 1`; input/output, if `IDO = 2`)

Since, in general, the length of `TABLE` will not be known in advance, `MAXTAB` is an upper bound for this length. Empty cells are included in `TABLE`, and each element of `TABLE` is nonnegative. The cells of `TABLE` are sequenced so that the first variable cycles from 1 to `NCLVAL(1)` one time, the second variable cycles from 1 to `NCLVAL(2)` `NCLVAL(1)` times, and so on, up to the `NCLVAR`-th variable, which cycles from 1 to `NCLVAL(NCLVAR)` most rapidly (`NCLVAL(1) * NCLVAL(2) * ... * NCLVAL(NCLVAR - 1)` times). That is to say, the second element of `TABLE` is the count for the first value for each classification variable except the last one and the second value of the last classification variable (assuming that variable takes more than one distinct value).

Optional Arguments

IDO — Processing option. (Input)
Default: `IDO = 1`.

IDO Action

- 1 This is the first (or the only) invocation of `FREQ` for this data set. Initialization and updating for the data in `X` are performed.
- 2 This is an additional invocation of `FREQ`, and updating for the data in `X` is performed.

NOBS — Number of observations. (Input)
Default: `NOBS = size(x,1)`.

NCOL — Number of columns in `X`. (Input)
Default: `NCOL = size(x,2)`.

LDX — Leading dimension of x exactly as specified in the dimension statement in the calling program. (Input)

Default: $LDX = \text{size}(x,1)$.

IFRQ — Frequency option. (Input)

$IFRQ = 0$ means that all frequencies are 1.0. For positive $IFRQ$, column number $IFRQ$ of x contains the frequencies.

Default: $IFRQ = 0$.

NCLVAR — Number of classification variables. (Input)

$NCLVAR$ must be greater than one.

Default: $NCLVAR = \text{size}(\text{INDCL},1)$.

FORTRAN 90 Interface

Generic: CALL `FREQ` (X , $INDCL$, $MAXTAB$, $MAXCL$, $NCLVAL$, $CLVAL$,
 $TABLE$ [, ...])

Specific: The specific interface names are `S_FREQ` and `D_FREQ`.

FORTRAN 77 Interface

Single: CALL `FREQ` (IDO , $NOBS$, $NCOL$, X , LDX , $IFRQ$, $NCLVAR$, $INDCL$,
 $MAXTAB$, $MAXCL$, $NCLVAL$, $CLVAL$, $TABLE$)

Double: The double precision name is `DFREQ`.

Example

The data for this example are taken from the examples used in routine `TWFRQ` (page 7), but modified so that the values of all points within a given interval of Example 2 for `TWFRQ` are exactly equal to the class mark for that interval. The results from this example, therefore, are the same as for Example 2 for `TWFRQ`, except that $TABLE$ is a vector. (The elements of the vector are sequenced as the columns of the matrix.)

```
USE FREQ_INT
USE UMACH_INT
INTEGER   LDX, MAXCL, MAXTAB, NCLVAR, NCOL
PARAMETER (LDX=30, MAXCL=15, MAXTAB=40, NCLVAR=2, NCOL=2)
!
INTEGER   I, INDCL(NCLVAR), NCLVAL(NCLVAR), NOUT, &
          NVAL1, NVAL2
REAL      CLVAL(MAXCL), TABLE(MAXTAB), X(LDX,NCOL)
!
DATA X/0.50, 1.50, 0.50, 1.50, 1.50, 1.50, 0.50, 1.50, 3.50, &
     2.50, 2.50, 3.50, 1.50, 2.50, 0.50, 1.50, 1.50, 0.50, &
     0.50, 0.50, 2.50, 1.50, 1.50, 1.50, 4.50, 2.50, 0.50, &
     1.50, 0.50, 2.50, &
     1.50, 3.50, 3.50, 2.50, 3.50, 4.50, 1.50, 3.50, 6.50, &
     3.50, 4.50, 6.50, 2.50, 4.50, 3.50, 2.50, 3.50, 3.50, &
```

```

1.50, 2.50, 5.50, 2.50, 3.50, 4.50, 5.50, 4.50, 3.50, &
2.50, 2.50, 5.50/
!
CALL UMACH (2, NOUT)
INDCL(1) = 1
INDCL(2) = 2
CALL FREQ (X, INDCL, MAXTAB, MAXCL, NCLVAL, CLVAL, TABLE)
NVAL1 = NCLVAL(1)
NVAL2 = NCLVAL(2)
WRITE (NOUT,99999) (CLVAL(J),J=NVAL1+1,NVAL1+NVAL2), &
(CLVAL(I), (TABLE((I-1)*NVAL2+J),J=1,NVAL2),I=1,NVAL1)
99999 FORMAT ('      Frequencies for All Combinations of Values', /, &
8X,6F7.2,/,5(F7.2,6F7.0,/))
END

```

Output

```

Frequencies for All Combinations of Values
      1.50   2.50   3.50   4.50   5.50   6.50
0.50      3.    2.    4.    0.    0.    0.
1.50      0.    5.    5.    2.    0.    0.
2.50      0.    0.    1.    3.    2.    0.
3.50      0.    0.    0.    0.    0.    2.
4.50      0.    0.    0.    0.    1.    0.

```

Comments

1. Workspace may be explicitly provided, if desired, by use of `F2EQ/DF2EQ`. The reference is

```

CALL F2EQ (IDO, NOBS, NCOL, X, LDX, IFRQ, NCLVAR, INDCL, MAXTAB,
MAXCL, NCLVAL, CLVAL, TABLE, IWK, WK)

```

The additional arguments are as follows:

IWK — Workspace of length `NCLVAR`.

WK — Workspace of length `NCLVAR`.

2. Informational errors
Type Code

```

4      1  MAXCL is too small. Increase the length of CLVAL.
4      2  MAXTAB is too small. Increase the length of TABLE.

```

Description

The routine `FREQ` determines the distinct values in multivariate data and computes frequencies for the data. The routine accepts the data in the matrix `X`, but performs computations only for the variables (columns) in `X` specified in `INDCL`. In general, the variables for which frequencies should be computed are discrete; that is, they should take on a relatively small number of different values. Variables that are continuous can be grouped first.

The routine `OWFRQ` (page 3) or `TWFRQ` (page 7) can be used to group variables and determine the frequencies of groups. The routine `FREQ` fills the vector `CLVAL` with the unique values of the variables and tallies the number of unique values of each variable in the vector `NCLVAL`. Each combination of one value from each variable forms a cell in a multiway table. The frequencies of these cells are entered in `TABLE` so that the first variable cycles through its values exactly once and the last variable cycles through its values most rapidly. Some cells may not correspond to any observation in the data; that is, “missing cells” are included and have 0’s in `TABLE`.

The length of the vectors `CLVAL` and `TABLE` depend on the data. The parameters `MAXCL` and `MAXTAB` are used as checks that the arrays sizes are not exceeded.

UVSTA

Computes basic univariate statistics.

Required Arguments

X — $|\text{NROW}|$ by $\text{NVAR} + m$ matrix containing the data, where m is 0, 1, or 2 depending on whether any column(s) of X correspond to weights and/or frequencies. (Input)

STAT — 15 by NVAR matrix containing in each row statistics on all of the variables.

(Output, if `IDO` = 0 or 1; input/output, if `IDO` = 2 or 3.)

The columns of *STAT* correspond to the columns of *X*, except for the columns of *X* containing weights or frequencies. (The columns beyond the weights or frequencies column are shifted to the left.)

I **STAT(I, *)**

1 contains means

2 contains variances

3 contains standard deviations

4 contains coefficients of skewness

5 contains coefficients of excess (kurtosis)

6 contains minima

7 contains maxima

8 contains ranges

9 contains coefficients of variation, when they are defined. If the coefficient of variation is not defined for a given variable, `STAT(9, *)` contains a zero in the corresponding position.

- 10 contains numbers (counts) of nonmissing observations
- 11 is used only when `CONPRM` is positive, and, in this case, contains the lower confidence limit for the mean (assuming normality)
- 12 is used only when `CONPRM` is positive, and, in this case, contains the upper confidence limit for the mean (assuming normality)
- 13 is used only when `CONPRV` is positive, and, in this case, contains the lower confidence limit for the variance (assuming normality).
- 14 is used only when `CONPRV` is positive, and, in this case, contains the upper confidence limit for the variance (assuming normality).
- 15 is used only when weighting is used (`IWT` is nonnegative), and, in this case, contains the sums of the weights.

Optional Arguments

IDO — Processing option. (Input)

Default: `IDO = 0`.

IDO Action

- 0 This is the only invocation of `UVSTA` for this data set, and all the data are input at once.
- 1 This is the first invocation, and additional calls to `UVSTA` will be made. Initialization and updating for the data in `X` are performed. The means are output correctly, but the other quantities output in `STAT` are intermediate quantities.
- 2 This is an intermediate invocation of `UVSTA`, and updating for the data in `X` is performed.
- 3 This is the final invocation of this routine. If `NROW` is not zero, updating is performed. The wrap-up computations for `STAT` are performed.

NROW — The absolute value of `NROW` is the number of rows of data currently input in `X`.

(Input)

Default: `NROW = size(X,1)`.

`NROW` may be positive, zero, or negative. Negative `NROW` means that the `-NROW` rows of data are to be deleted from some aspects of the analysis, and this should be done only if `IDO` is 2 or 3 and the wrap-up computations for `STAT` have not been performed. When a negative value is input for `NROW`, it is assumed that each of the `-NROW` rows of `X` has been input (with positive `NROW`) in a previous invocation of `UVSTA`. Use of negative values of `NROW` should be made with care and with the understanding that some quantities in `STAT` cannot be updated properly in this case. In particular, the minima,

maxima, and ranges are not updated because of deletion. It is also possible that a constant variable in the remaining data will not be recognized as such.

NVAR — Number of variables (not including the weight or frequency variable, if used). (Input)
Default: `NVAR = size (x,2)`.

LDX — Leading dimension of `X` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDX = size (x,1)`.

IFRQ — Frequency option. (Input)
`IFRQ = 0` means that all frequencies are 1.0. For positive `IFRQ`, column number `IFRQ` of `X` contains the frequencies.
Default: `IFRQ = 0`.

IWT — Weighting option. (Input)
`IWT = 0` means that all weights are 1.0. For positive `IWT`, column `IWT` of `X` contains the weights.
Default: `IWT = 0`.

MOPT — Missing value option. (Input)
NaN (not a number from routine `AMACH(6)`) is interpreted as the missing value code and any value in `X` equal to NaN is excluded from the computations.
Default: `MOPT = 0`.

MOPT Action

- 0 The exclusion is listwise. (The entire row of `X` is excluded if any of the values of the row is equal to the missing value code.)
- 1 The exclusion is elementwise. (Statistics for variables with nonmissing values are updated.)

CONPRM — Confidence level for two-sided interval estimate of the means (assuming normality), in percent. (Input)
If `CONPRM ≤ 0`, no confidence interval for the mean is computed; otherwise, a `CONPRM` percent confidence interval is computed, in which case `CONPRM` must be between 0.0 and 100.0. `CONPRM` is often 90.0, 95.0, or 99.0. For a one-sided confidence interval with confidence level `ONECL`, set `CONPRM = 100.0 - 2.0 * (100.0 - ONECL)`.
Default: `CONPRM = .95.0`.

CONPRV — Confidence level for two-sided interval estimate of the variances (assuming normality), in percent. (Input)
The confidence intervals are symmetric in probability (rather than in length). See also the description of `CONPRM`.
Default: `CONPRV = .95.0`.

IPRINT — Printing option. (Input)

Default: IPRINT = 0.

IPRINT	Action
---------------	---------------

0	No printing is performed.
---	---------------------------

1	Statistics in STAT are printed if IDO = 0 or 3.
---	---

2	Intermediate means, sums of squares about the mean, minima, maxima, and counts are printed when IDO = 1 or 2, and all statistics in STAT are printed when IDO = 0 or 3.
---	---

LDSTAT — Leading dimension of STAT exactly as specified in the dimension statement in the calling program. (Input)

Default: LDSTAT = size (STAT,1).

NRMIS — Number of rows of data encountered in calls to UVSTA that contain any missing values. (Output, if IDO = 0 or 1; input/output, if IDO = 2 or 3.)

Rows with a frequency of zero are not counted.

FORTRAN 90 Interface

Generic: CALL UVSTA (X, STAT [, ...])

Specific: The specific interface names are S_UVSTA and D_UVSTA.

FORTRAN 77 Interface

Single: CALL UVSTA (IDO, NROW, NVAR, X, LDX, IFRQ, IWT, MOPT, CONPRM, CONPRV, IPRINT, STAT, LDSTAT, NRMIS)

Double: The double precision name is DUVSTA.

Example 1

This example uses data from Draper and Smith (1981). There are 5 variables and 13 observations.

```
USE UVSTA_INT
USE GDATA_INT
INTEGER LDSTAT, LDX, NVAR
PARAMETER (LDSTAT=15, LDX=13, NVAR=5)
!
INTEGER IPRINT, NR, NROW, NV
REAL CONPRM, CONPRV, STAT(LDSTAT,NVAR), X(LDX,NVAR)
! Get data for example.
CALL GDATA (5, X, NR, NV)
! All data are input at once.
NROW = NR
```

```

!           No unequal frequencies or weights
!           are used.
!           Get 95% confidence limits.
!           Delete any row containing a missing
!           value.
!           Print results.
IPRINT = 1
CALL UVSTA (X, STAT, NROW=NROW, IPRINT=IPRINT)
END

```

Output

Univariate Statistics from UVSTA

Variable	Mean	Variance	Std. Dev.	Skewness	Kurtosis
1	7.4615	34.6026	5.8824	0.68768	0.07472
2	48.1538	242.1410	15.5609	-0.04726	-1.32257
3	11.7692	41.0256	6.4051	0.61064	-1.07916
4	30.0000	280.1667	16.7382	0.32960	-1.01406
5	95.4231	226.3136	15.0437	-0.19486	-1.34244

Variable	Minimum	Maximum	Range	Coef. Var.	Count
1	1.0000	21.0000	20.0000	0.7884	13.0000
2	26.0000	71.0000	45.0000	0.3231	13.0000
3	4.0000	23.0000	19.0000	0.5442	13.0000
4	6.0000	60.0000	54.0000	0.5579	13.0000
5	72.5000	115.9000	43.4000	0.1577	13.0000

Variable	Lower CLM	Upper CLM	Lower CLV	Upper CLV
1	3.9068	11.0162	17.7930	94.2894
2	38.7505	57.5572	124.5113	659.8163
3	7.8987	15.6398	21.0958	111.7918
4	19.8852	40.1148	144.0645	763.4335
5	86.3322	104.5139	116.3726	616.6877

Comments

Workspace may be explicitly provided, if desired, by use of U2STA/DU2STA. The reference is

```
CALL U2STA (IDO, NROW, NVAR, X, LDX, IFRQ, IWT, MOPT, CONPRM, CONPRV,
IPRINT, STAT, LDSTAT, NRMISS, WK)
```

The additional argument is

WK — Real work vector of length specified above. **WK** should not be changed between calls to U2STA.

Description

For the data in each column of **X**, except the columns containing frequencies or weights, **UVSTA** computes the sample mean, variance, minimum, maximum, and other basic statistics. It also computes confidence intervals for the mean and variance if the sample is assumed to be from a normal population.

Missing values, that is, values equal to NaN (not a number, the value returned by routine `AMACH(6)`), are excluded from the computations. If `MOPT` is positive, the exclusion is listwise; that is, the entire observation is excluded and no computations are performed even for the variables with valid values. If frequencies or weights are specified, any observation whose frequency or weight is missing is excluded from the computations.

Frequencies are interpreted as multiple occurrences of the other values in the observations. That is, a row of `X` with a frequency variable having a value of 2 has the same effect as two rows with frequencies of 1. The total of the frequencies is used in computing all of the statistics based on moments (mean, variance, skewness, and kurtosis). Weights are not viewed as replication factors. The sum of the weights is used only in computing the mean (of course, then the weighted mean is used in computing the central moments). Both weights and frequencies can be zero, but neither can be negative. In general, a zero frequency means that the row is to be eliminated from the analysis; no further processing, counting of missing values, or error checking is done on the row. Although it is not required that frequencies be integers, the logic of their treatment implicitly assumes that they are. Weights, on the other hand, are allowed to be continuous. A weight of zero results in the row being counted, and updates are made of statistics and of the number of missing values. A missing value for the frequency or a missing value for the weight when the frequency is nonzero results in the row being deleted from the analysis; but even in that case, if one is nonmissing, it is an error for that nonmissing weight or frequency to be negative.

The definitions of some of the statistics are given below in terms of a single variable x . The i -th datum is x_i , with corresponding frequency f_i and weight w_i . If either frequencies or weights are not specified, f_i and/or w_i are identically one. The summation in each case is over the set of valid observations, based on the setting of `MOPT` and the presence of missing values in the data.

Number of nonmissing observations, `STAT(10, *)`

$$n = \sum f_i$$

Mean, `STAT(1, *)`

$$\bar{x}_w = \frac{\sum f_i w_i x_i}{\sum f_i w_i}$$

Variance, `STAT(2, *)`

$$s_w^2 = \frac{\sum f_i w_i (x_i - \bar{x}_w)^2}{n - 1}$$

Skewness, `STAT(4, *)`

$$\frac{\sum f_i w_i (x_i - \bar{x}_w)^3 / n}{[\sum f_i w_i (x_i - \bar{x}_w)^2 / n]^{3/2}}$$

Excess or Kurtosis, STAT(5, *)

$$\frac{\sum f_i w_i (x_i - \bar{x}_w)^4 / n}{[\sum f_i w_i (x_i - \bar{x}_w)^2 / n]^2} - 3$$

Minimum, STAT(6, *)

$$x_{\min} = \min(x_i)$$

Maximum, STAT(7, *)

$$x_{\max} = \max(x_i)$$

Range, STAT(8, *)

$$x_{\max} - x_{\min}$$

Coefficient of Variation, STAT(9, *)

$$\frac{s_w}{\bar{x}_w} \quad \text{for } \bar{x}_w \neq 0$$

The arguments `IDO` and `NROW` allow data to be input a few at a time and even to be deleted after having been included in the analysis. The minima, maxima, and ranges are not updated when observations are deleted.

Additional Examples

Example 2

In this example, we use some simple data to illustrate the use of frequencies, missing values, and the parameters `IDO` and `NROW`. In the data below, “NaN” represents a missing value.

<i>f</i>	<i>x</i>	<i>y</i>
2	3.0	5.0
1	9.0	2.0
3	1.0	NaN

We bring in the data one observation at a time in this example. Also, we bring in one false datum and then delete it on a subsequent call to `UVSTA`.

```
USE IMSL_LIBRARIES
INTEGER    LDSTAT, NVAR
PARAMETER (LDSTAT=15, NVAR=2)
!
INTEGER    IDO, IFRQ, IPRINT, MOPT, NRMIS, NROW
REAL      STAT(LDSTAT,NVAR), X1(1,NVAR+1)
!
!                                     All data are input one observation
!                                     at a time in the vector X1.
NROW = 1
!
!                                     Frequencies are in the first
```

```

!                                     position. No weights are used.
IFRQ = 1
!                                     Get 95% confidence limits.
!                                     Elementwise deletion of missing
!                                     values.
MOPT = 1
!                                     Print results, intermediate as well.
IPRINT = 2
!                                     Bring in the first observation.
IDO = 1
X1(1,1) = 2.0
X1(1,2) = 3.0
X1(1,3) = 5.0
CALL UVSTA (X1, STAT, IDO=IDO, NVAR=NVAR, IFRQ=IFRQ, MOPT=MOPT, &
            IPRINT=IPRINT, NRMISS=NRMISS)
!                                     Bring in the second observation.
IDO = 2
X1(1,1) = 1.0
X1(1,2) = 9.0
X1(1,3) = 2.0
CALL UVSTA (X1, STAT, IDO=IDO, NVAR=NVAR, IFRQ=IFRQ, MOPT=MOPT, &
            IPRINT=IPRINT, NRMISS=NRMISS)
!                                     Bring in a false observation.
X1(1,1) = 3.0
X1(1,2) = 6.0
X1(1,3) = 3.0
CALL UVSTA (X1, STAT, IDO=IDO, NVAR=NVAR, IFRQ=IFRQ, MOPT=MOPT, &
            IPRINT=IPRINT, NRMISS=NRMISS)
!                                     Delete the false observation.
!                                     This may make the minima, maxima,
!                                     and range incorrect.
NROW = -1
X1(1,1) = 3.0
X1(1,2) = 6.0
X1(1,3) = 3.0
CALL UVSTA (X1, STAT, IDO=IDO, NROW=NROW, NVAR=NVAR, IFRQ=IFRQ, &
            MOPT=MOPT, IPRINT=IPRINT, NRMISS=NRMISS)
NROW = 1
!                                     Bring in the final observation.
IDO = 3
X1(1,1) = 3.0
X1(1,2) = 1.0
X1(1,3) = AMACH(6)
CALL UVSTA (X1, STAT, IDO=IDO, NROW=NROW, NVAR=NVAR, IFRQ=IFRQ, &
            MOPT=MOPT, IPRINT=IPRINT, NRMISS=NRMISS)
END

```

Output

Intermediate Statistics from UVSTA					
Variable	Mean	Sum Sqs.	Minimum	Maximum	Count
1	3.0000	0.0000	3.0000	3.0000	2.0000
2	5.0000	0.0000	5.0000	5.0000	2.0000

Intermediate Statistics from UVSTA					
Variable	Mean	Sum Sqs.	Minimum	Maximum	Count

1	5.0000	24.0000	3.0000	9.0000	3.0000
2	4.0000	6.0000	2.0000	5.0000	3.0000
Intermediate Statistics from UVSTA					
Variable	Mean	Sum Sqs.	Minimum	Maximum	Count
1	5.5000	25.5000	3.0000	9.0000	6.0000
2	3.5000	7.5000	2.0000	5.0000	6.0000
Intermediate Statistics from UVSTA					
Variable	Mean	Sum Sqs.	Minimum	Maximum	Count
1	5.0000	24.0000	3.0000	9.0000	3.0000
2	4.0000	6.0000	2.0000	5.0000	3.0000
Univariate Statistics from UVSTA					
Variable	Mean	Variance	Std. Dev.	Skewness	Kurtosis
1	3.0000	9.6000	3.0984	1.4142	0.5000
2	4.0000	3.0000	1.7321	-0.7071	-1.5000
Variable	Minimum	Maximum	Range	Coef. Var.	Count
1	1.0000	9.0000	8.0000	1.0328	6.0000
2	2.0000	5.0000	3.0000	0.4330	3.0000
Variable	Lower CLM	Upper CLM	Lower CLV	Upper CLV	
1	-0.2516	6.2516	3.7405	57.7470	
2	-0.3027	8.3027	0.8133	118.4935	

RANKS

Computes the ranks, normal scores, or exponential scores for a vector of observations.

Required Arguments

X— Vector of length *NOBS* containing the observations to be ranked. (Input)

SCORE— Vector of length *NOBS* containing the rank or a transformation of that rank of each observation. (Output)

X and *SCORE* may occupy the same memory.

Optional Arguments

NOBS— Number of observations. (Input)

Default: *NOBS* = size(*X*,1).

FUZZ— Value used to determine ties. (Input)

If $|x(i) - x(j)|$ is less than or equal to *FUZZ*, then *x(i)* and *x(j)* are said to be tied.

Default: *FUZZ* = 0.0.

ITIE — Option for determining the method used to assign a score to tied observations.
(Input)
Default: `ITIE = 0`.

ITIE	Method
0	The average of the scores of the tied observations is used.
1	The highest score in the group of ties is used.
2	The lowest score in the group of ties is used.
3	The tied observations are to be randomly untied using an IMSL random number generator.

ISCORE — Option for specifying the type of values returned in `SCORE`. (Input)
Default: `ISCORE = 0`.

ISCORE	Type
0	Ranks
1	Blom version of normal scores
2	Tukey version of normal scores
3	Van der Waerdan version of normal scores
4	Expected value of normal order statistics (For tied observations, the average of the expected normal scores are used.)
5	Savage scores (the expected value of exponential order statistics)

FORTRAN 90 Interface

Generic: `CALL RANKS (X, SCORE [,...])`

Specific: The specific interface names are `S_RANKS` and `D_RANKS`.

FORTRAN 77 Interface

Single: `CALL RANKS (NOBS, X, FUZZ, ITIE, ISCORE, SCORE)`

Double: The double precision name is `DRANKS`.

Example

The data for this example, from Hinkley (1977), are the same used in several examples in this chapter. There are 30 observations. Note that the fourth and sixth observations are tied and that the third and twentieth are tied.

```
USE RANKS_INT
USE UMACH_INT
USE RNSET_INT

INTEGER      NOBS
PARAMETER   (NOBS=30)

!
INTEGER      ISCORE, ISEED, ITIE, NOUT
REAL         FUZZ, SCORE(NOBS), X(NOBS)
!
DATA X/0.77, 1.74, 0.81, 1.20, 1.95, 1.20, 0.47, 1.43, 3.37,&
      2.20, 3.00, 3.09, 1.51, 2.10, 0.52, 1.62, 1.31, 0.32, 0.59, &
      0.81, 2.81, 1.87, 1.18, 1.35, 4.75, 2.48, 0.96, 1.89, 0.90, &
      2.05/
!
CALL UMACH (2, NOUT)
!
ISCORE = 0
!
ITIE = 0
FUZZ = 0.0
!
CALL RANKS (X, SCORE, ISCORE=ISCORE, ITIE=ITIE, FUZZ=FUZZ)
WRITE (NOUT,99994) SCORE
99994 FORMAT (' Ranks', /, (1X,10F7.1))
!
ISCORE = 1
!
ITIE = 1
FUZZ = 0.0
!
CALL RANKS (X, SCORE, ISCORE=ISCORE, ITIE=ITIE, FUZZ=FUZZ)
WRITE (NOUT,99995) SCORE
99995 FORMAT (/, ' Blom normal scores', /, (1X,10F7.3))
!
ISCORE = 2
!
ITIE = 2
FUZZ = 0.0
!
CALL RANKS (X, SCORE, ISCORE=ISCORE, ITIE=ITIE, FUZZ=FUZZ)
WRITE (NOUT,99996) SCORE
99996 FORMAT (/, ' Tukey normal scores', /, (1X,10F7.3))
!
ISCORE = 3
!
ITIE = 3
FUZZ = 0.0
!
ISEED = 123457
CALL RNSET (ISEED)
ITIE = 3
FUZZ = 0.0
```

```

!
CALL RANKS (X, SCORE, ISCORE=ISCORE, ITIE=ITIE, FUZZ=FUZZ)
WRITE (NOUT,99997) SCORE
99997 FORMAT (/, ' Van der Waerden scores', /, (1X,10F7.3))
! Expected value of normal O. S.
ISCORE = 4
! Average ties.
ITIE = 0
FUZZ = 0.0
!
CALL RANKS (X, SCORE, ISCORE=ISCORE, ITIE=ITIE, FUZZ=FUZZ)
WRITE (NOUT,99998) SCORE
99998 FORMAT (/, ' Expected values of normal order statistics', /,&
(1X,10F7.3))
! Savage scores.
ISCORE = 5
! Average ties.
ITIE = 0
FUZZ = 0.0
!
CALL RANKS (X, SCORE, ISCORE=ISCORE, ITIE=ITIE, FUZZ=FUZZ)
WRITE (NOUT,99999) SCORE
99999 FORMAT (/, ' Expected values of exponential order statistics', &
/, (1X,10F7.2))
END

```

Output

Ranks

5.0	18.0	6.5	11.5	21.0	11.5	2.0	15.0	29.0	24.0
27.0	28.0	16.0	23.0	3.0	17.0	13.0	1.0	4.0	6.5
26.0	19.0	10.0	14.0	30.0	25.0	9.0	20.0	8.0	22.0

Blom normal scores

-1.024	0.209	-0.776	-0.294	0.473	-0.294	-1.610	-0.041	1.610	0.776
1.176	1.361	0.041	0.668	-1.361	0.125	-0.209	-2.040	-1.176	-0.776
1.024	0.294	-0.473	-0.125	2.040	0.893	-0.568	0.382	-0.668	0.568

Tukey normal scores

-1.020	0.208	-0.890	-0.381	0.471	-0.381	-1.599	-0.041	1.599	0.773
1.171	1.354	0.041	0.666	-1.354	0.124	-0.208	-2.015	-1.171	-0.890
1.020	0.293	-0.471	-0.124	2.015	0.890	-0.566	0.381	-0.666	0.566

Van der Waerden scores

-0.989	0.204	-0.753	-0.287	0.460	-0.372	-1.518	-0.040	1.518	0.753
1.131	1.300	0.040	0.649	-1.300	0.122	-0.204	-1.849	-1.131	-0.865
0.989	0.287	-0.460	-0.122	1.849	0.865	-0.552	0.372	-0.649	0.552

Expected values of normal order statistics

-1.026	0.209	-0.836	-0.338	0.473	-0.338	-1.616	-0.041	1.616	0.777
1.179	1.365	0.041	0.669	-1.365	0.125	-0.209	-2.043	-1.179	-0.836
1.026	0.294	-0.473	-0.125	2.043	0.894	-0.568	0.382	-0.669	0.568

Expected values of exponential order statistics

0.18	0.89	0.24	0.47	1.17	0.47	0.07	0.68	2.99	1.54
2.16	2.49	0.74	1.40	0.10	0.81	0.56	0.03	0.14	0.24

1.91 0.98 0.40 0.61 3.99 1.71 0.35 1.07 0.30 1.28

Comments

1. Workspace may be explicitly provided, if desired, by use `R2NKS/DR2NKS`. The reference is:

```
CALL R2NKS (NOBS, X, FUZZ, ITIE, ISCORE, SCORE, IWK)
```

The additional argument is:

IWK — Integer work vector of length `NOBS`.

2. The routine `RNSET` (see Chapter 18, Random Number Generation) can be used to initialize the seed of the random number generator used to break ties. If the seed is not initialized by `RNSET`; different runs of the same program can yield different results if there are tied observations and `ITIE = 3`.

Description

The routine `RANKS` determines the ranks, or various transformations of the ranks of the data in `X`. Ties in the data can be resolved in four different ways, as specified in `ITIE`.

ISCORE = 0: Ranks

For this option, the values output in `SCORE` are the ordinary ranks of the data in `X`. If `X(I)` has the smallest value among those in `X` and there is no other element in `X` with this value, then `SCORE(I) = 1`. If both `X(I)` and `X(J)` have the same smallest value, then

if `ITIE = 0`, `SCORE(I) = SCORE(J) = 1.5`

if `ITIE = 1`, `SCORE(I) = SCORE(J) = 2.0`

if `ITIE = 2`, `SCORE(I) = SCORE(J) = 1.0`

if `ITIE = 3`, `SCORE(I) = 1.0` and `SCORE(J) = 2.0`
or `SCORE(I) = 2.0` and `SCORE(J) = 1.0`.

When the ties are resolved by use of routine `RNUNF` (see Chapter 18, Random Number Generation) to generate random numbers, different results may occur when running the same program at different times unless the “seed” of the random number generator is set explicitly by use of the routine `RNSET` (see Chapter 18, Random Number Generation). Ordinarily, there is no need to call the routine to set the seed, even if there are ties in the data.

ISCORE = 1: Normal Scores, Blom Version

Normal scores are expected values, or approximations to the expected values, of order statistics from a normal distribution. The simplest approximations are obtained by evaluating the inverse cumulative normal distribution function (routine `ANORIN`, see Chapter 18, Random Number Generation) at the ranks scaled into the open interval (0, 1). In the Blom version (see

Blom 1958), the scaling transformation for the rank r_i ($1 \leq r_i \leq n$, where n is the sample size, NOBS) is $(r_i - 3/8)/(n + 1/4)$. The Blom normal score corresponding to the observation with rank r_i is

$$\Phi^{-1}\left(\frac{r_i - 3/8}{n + 1/4}\right)$$

where $\Phi(\cdot)$ is the normal cumulative distribution function.

Adjustments for ties are made after the normal score transformation. That is, if $x(\mathbb{I})$ equals $x(\mathbb{J})$ (within FUZZ) and their value is the k -th smallest in the data set, the Blom normal scores are determined for ranks of k and $k + 1$, and then these normal scores are averaged or selected in the manner specified by TIE. (Whether the transformations are made first or ties are resolved first makes no difference except when averaging is done.)

IScore = 2: Normal Scores, Tukey Version

In the Tukey version (see Tukey 1962), the scaling transformation for the rank r_i is $(r_i - 1/3)/(n + 1/3)$. The Tukey normal score corresponding to the observation with rank r_i is

$$\Phi^{-1}\left(\frac{r_i - 1/3}{n + 1/3}\right)$$

Ties are handled in the same way as discussed above for the Blom normal scores.

IScore = 3: Normal Scores, Van der Waerden Version

In the Van der Waerden version (see Lehmann 1975, page 97), the scaling transformation for the rank r_i is $r_i/(n + 1)$. The Van der Waerden normal score corresponding to the observation with rank r_i is

$$\Phi^{-1}\left(\frac{r_i}{n + 1}\right)$$

Ties are handled in the same way as discussed above for the Blom normal scores.

IScore = 4: Expected Value of Normal Order Statistics

For this option, the values output in SCORE are the expected values of the normal order statistics from a sample of size NOBS. If the value in $x(\mathbb{I})$ is the k -th smallest, then the value output in SCORE(\mathbb{I}) is $E(Z_k)$, where $E(\cdot)$ is the expectation operator and Z_k is the k -th order statistic in a sample of size NOBS from a standard normal distribution. Such expected values are computed by the routine ENOS (see Chapter 20, Mathematical Support). Ties are handled in the same way as discussed above for the Blom normal scores.

IScore = 5: Savage Scores

For this option, the values output in SCORE are the expected values of the exponential order statistics from a sample of size NOBS. These values are called Savage scores because of their use

in a test discussed by Savage (1956) (see Lehman 1975). If the value in $x(\mathbb{I})$ is the k -th smallest, then the value output in $\text{SCORE}(\mathbb{I})$ is $E(Y_k)$, where Y_k is the k -th order statistic in a sample of size NOBS from a standard exponential distribution. The expected value of the k -th order statistic from an exponential sample of size n (NOBS) is

$$\frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{n-k+1}$$

Ties are handled in the same way as discussed above for the Blom normal scores.

The example uses all of these options with the same data set, which contains some ties. The ties are handled different ways in this example.

LETTR

Produces a letter value summary.

Required Arguments

X — Vector of length NOBS containing the data. (Input)

SUMRY — Vector of length NUM containing the summary letter values. (Output)

If NUM is 5, for example, *SUMRY* contains the minimum, the lower hinge (quartile), the median, the upper hinge, and the maximum, in that order.

NMISS — Number of missing values. (Output)

Optional Arguments

NOBS — Number of observations. (Input)

Default: $\text{NOBS} = \text{size}(X,1)$.

NUM — Number of summary values. (Input)

NUM must be an odd integer greater than or equal to 3. A common value for NUM is 5.

Default: $\text{NUM} = 5$.

FORTRAN 90 Interface

Generic: `CALL LETTR (X, SUMRY, NMISS [, ...])`

Specific: The specific interface names are `S_LETTR` and `D_LETTR`.

FORTRAN 77 Interface

Single: `CALL LETTR (NOBS, X, NUM, SUMRY, NMISS)`

Double: The double precision name is `DLETTR`.

Example

In this example, LETTR is used to compute a letter value summary of the measurements (in inches) of precipitation in Minneapolis/St. Paul during the month of March for 30 consecutive years. These data were studied by Hinkley (1977) and by Velleman and Hoaglin (1981), pages 50–53.

```
USE LETTR_INT
USE UMACH_INT
INTEGER I, NMISS, NOBS, NOUT, NUM
REAL SUMRY(11), X(30)
!
DATA X/0.77, 1.74, 0.81, 1.20, 1.95, 1.20, 0.47, 1.43, 3.37, &
      2.20, 3.00, 3.09, 1.51, 2.10, 0.52, 1.62, 1.31, 0.32, 0.59, &
      0.81, 2.81, 1.87, 1.18, 1.35, 4.75, 2.48, 0.96, 1.89, 0.90, &
      2.05/
!
CALL UMACH (2, NOUT)
NOBS = 30
NUM = 11
!
CALL LETTR (X, SUMRY, NMISS, NUM=NUM)
WRITE (NOUT,99998) SUMRY(6), (SUMRY(6-I),SUMRY(6+I),I=1,5)
99998 FORMAT (' Letter Values', /, ' Lower Upper', &
             /, ' M ', F6.3, /, ' H ', F6.3, 6X, F6.3, /, &
             ' E ', F6.3, 6X, F6.3, /, ' D ', F6.3, 6X, F6.3, /, &
             ' ! ', F6.3, 6X, F6.3, /, ' m/M ', F6.3, 6X, F6.3)
WRITE (NOUT,99999) NMISS
99999 FORMAT (' There are ', I2, ' missing values.')
END
```

Output

```
Letter Values
  Lower      Upper
M          1.470
H  0.900     2.100
E  0.680     2.905
D  0.495     3.230
!  0.395     4.060
m/M 0.320     4.750
There are  0 missing values.
```

Comments

1. Workspace may be explicitly provided, if desired, by use of L2TTR/DL2TTR. The reference is:

```
CALL L2TTR (NOBS, X, NUM, SUMRY, NMISS, WK)
```

The additional argument is:

WK — Work vector of length NOBS.

- | | | | |
|----|----------------------|------|---|
| 2. | Informational errors | | |
| | Type | Code | |
| | 3 | 3 | The results are likely not to be meaningful if <code>NUM</code> is larger than the number of valid observations, (<code>NOBS - NMISS</code>). |
| | 4 | 4 | The number of valid observations (<code>NOBS - NMISS</code>) is not greater than zero. |

Description

The routine `LETTR` computes the median (“*M*”), the minimum, the maximum, and other depths or “letter values”—hinges (“*H*”), eighths (“*E*”), sixteenths (“*D*”), etc.—as specified by `NUM`. If `NUM = 9`, for example, the values in `SUMRY` correspond to min, *D*, *E*, *H*, *M*, *H*, *E*, *D*, and max, in that order. The use of letter values in summarizing a set of data is due to Tukey. Examples and discussion of the use of letter values are given by Tukey (1977, Chapter 2) and by Velleman and Hoaglin (1981, Chapter 2).

ORDST

Determines order statistics.

Required Arguments

X — Vector of length `NOBS` containing the data. (Input)

NOS — Number of order statistics. (Input)
NOS must be greater than or equal to one and less than or equal to `NOBS`.

OS — Vector of length `NOS` containing the order statistics. (Output)

NMISS — Number of missing values. (Output)

Optional Arguments

NOBS — Number of observations. (Input)
NOBS must be greater than or equal to one.
 Default: `NOBS = size(X,1)`.

IOPT — Option to choose the order statistics to be calculated. (Input)
 Default: `IOPT = 1`.

IOPT Action

- | | |
|---|--|
| 0 | Calculate the <code>NOS</code> order statistics listed in <code>IOS</code> . |
| 1 | Calculate the first <code>NOS</code> order statistics. |
| 2 | Calculate the last <code>NOS</code> order statistics. |

IOS — If `IOPT = 0`, `IOS` is a vector of length `NOS` containing the ranks of the order statistics.

(Input)

The elements of `IOS` must be greater than or equal to one and less than or equal to `NOBS`. If `IOPT = 1` or `2`, `IOS` is unreferenced and can be defined as a vector of length 1.

FORTRAN 90 Interface

Generic: `CALL ORDST (X, NOS, OS, NMISS [,...])`

Specific: The specific interface names are `S_ORDST` and `D_ORDST`.

FORTRAN 77 Interface

Single: `CALL ORDST (NOBS, X, NOS, IOPT, IOS, OS, NMISS)`

Double: The double precision name is `DORDST`.

Example 1

The data for these examples are from Hinkley (1977) and Velleman and Hoaglin (1981). They are the measurements (in inches) of precipitation in Minneapolis/St. Paul during the month of March for 30 consecutive years. In the first example, the first five order statistics from a sample of size 30 are obtained. Since `IOPT` is set to 1, `IOS` is not used.

```
USE ORDST_INT
USE UMACH_INT
USE WRRRN_INT
USE AMACH_INT

INTEGER      NOBS, NOS
PARAMETER   (NOBS=30, NOS=5)
!
INTEGER      NMISS, NOUT
REAL         OS(NOS), X(NOBS)
!
DATA X/0.77, 1.74, 0.81, 1.20, 1.95, 1.20, 0.47, 1.43, 3.37, &
      2.20, 3.00, 3.09, 1.51, 2.10, 0.52, 1.62, 1.31, 0.32, 0.59, &
      0.81, 2.81, 1.87, 1.18, 1.35, 4.75, 2.48, 0.96, 1.89, 0.90, &
      2.05/
!
CALL UMACH (2, NOUT)
CALL ORDST (X, NOS, OS, NMISS)
CALL WRRRN ('First five order statistics:', OS, 1, NOS, 1)
WRITE (NOUT,99999) NMISS
99999 FORMAT ('  There are', I2, ' missing values.')
END
```

Output

```
First five order statistics:
   1       2       3       4       5
0.3200  0.4700  0.5200  0.5900  0.7700
There are 0 missing values.
```

Comments

1. Workspace may be explicitly provided, if desired, by use of `O2DST/DO2DST`. The reference is:

```
CALL O2DST (NOBS, X, NOS, IOPT, IOS, OS, NMISS, WK)
```

The additional argument is as follows:

WK — Work vector of length `NOBS`.

2. Informational errors

Type Code

- | | | |
|---|---|--|
| 3 | 1 | All of the observations are missing values. The elements of <code>OS</code> have been set to NaN (not a number). |
| 3 | 2 | <code>NOS</code> order statistics have been requested, but there are only <code>NOBS - NMISS</code> valid observations. Order statistics greater than <code>NOBS - NMISS</code> have been set to NaN (not a number). |
| 3 | 3 | Each value of <code>IOS</code> must be greater than 0 and less than or equal to the number of valid observations. The values of <code>OS</code> that are not defined have been set to NaN. |

3. Missing values (NaN) are excluded from the analysis. Order statistics are based on the `NOBS - NMISS` nonmissing elements of `X`.

Description

The routine `ORDST` determines order statistics from the data in `X` and returns them in the vector `OS`. The routine `ORDST` first checks to see if `X` is sorted, in which case the order statistics are merely picked from `X`. If `X` is not sorted, `ORDST` does either a complete or partial sort, depending on how many order statistics are requested. Since either the largest few order statistics or the smallest few are often of interest, the option parameter `IOPT` allows the user to obtain the largest or the smallest order statistics easily; otherwise (when `IOPT` is set to 0), the user specifies in the vector `IOS` exactly which order statistics are to be returned. If `IOS` is used, the order statistics returned in `OS` are in the same order as the indicators in `IOS`.

Additional Examples

Example 2

In the second example, the last five order statistics from a sample of size 30 are obtained. This example uses the same data as in the first example, but this time the first two observations have been set to a missing value indicator (`AMACH(6)`). Note that since there are two missing values in the data set, the indices of the last five order statistics are numbers 24, 25, 26, 27, and 28. In this example, `NMISS` will be returned with a value of 2. The index of the last order statistic can be determined by `NOBS - NMISS`.

```
USE ORDST_INT  
USE UMACH_INT
```

```

USE WRRRN_INT
USE AMACH_INT

INTEGER IOPT, NOBS, NOS
PARAMETER (IOPT=2, NOBS=30, NOS=5)
!
INTEGER NMISS, NOUT
REAL OS(NOS), X(NOBS)
!
DATA X/0.77, 1.74, 0.81, 1.20, 1.95, 1.20, 0.47, 1.43, 3.37, &
      2.20, 3.00, 3.09, 1.51, 2.10, 0.52, 1.62, 1.31, 0.32, 0.59, &
      0.81, 2.81, 1.87, 1.18, 1.35, 4.75, 2.48, 0.96, 1.89, 0.90, &
      2.05/
!
CALL UMACH (2, NOUT)
X(1) = AMACH(6)
X(2) = AMACH(6)
CALL ORDST (X, NOS, OS, NMISS, IOPT=IOPT)
CALL WRRRN ('Last five order statistics:', OS, 1, NOS, 1)
WRITE (NOUT,99999) NMISS
99999 FORMAT (' There are', I2, ' missing values.')
END

```

Output

```

Last five order statistics:
      1      2      3      4      5
2.810  3.000  3.090  3.370  4.750
There are 2 missing values.

```

Example 3

In this example, we illustrate the use of `IOS` to specify exactly which order statistics are to be computed. We request what would be the last five order statistics from a sample of size 30, that is, order statistics 26, 27, 28, 29, and 30. As in example two, the data set has two missing values. Order statistics 29 and 30 are not defined, but since they are specifically requested, a warning message is issued and `OS` contains two missing values on return.

```

USE ORDST_INT
USE UMACH_INT
USE WRRRN_INT
USE AMACH_INT

INTEGER IOPT, NOBS, NOS
PARAMETER (IOPT=0, NOBS=30, NOS=5)
!
INTEGER IOS(NOS), NMISS, NOUT
REAL OS(NOS), X(NOBS)
!
DATA X/0.77, 1.74, 0.81, 1.20, 1.95, 1.20, 0.47, 1.43, 3.37, &
      2.20, 3.00, 3.09, 1.51, 2.10, 0.52, 1.62, 1.31, 0.32, 0.59, &
      0.81, 2.81, 1.87, 1.18, 1.35, 4.75, 2.48, 0.96, 1.89, 0.90, &
      2.05/
DATA IOS/26, 27, 28, 29, 30/
!
CALL UMACH (2, NOUT)
X(1) = AMACH(6)

```

```

X(2) = AMACH(6)
CALL ORDST (X, NOS, OS, NMISS, IOS=IOS, IOPT=IOPT)
CALL WRRRN ('Last five order statistics:', OS, 1, NOS, 1)
WRITE (NOUT,99999) NMISS
99999 FORMAT ('   There are', I2, ' missing values.')
END

```

Output

```

*** WARNING  ERROR 3 from ORDST.  Each value of IOS must be greater than 0
***          and less than or equal to the number of valid observations,
***          NOBS-NMISS, which is 28.  IOS contains 2 values outside of
***          this range.  The corresponding values of OS have been set to
***          NaN (not a number).

```

```

Last five order statistics:
      1      2      3      4      5
3.090  3.370  4.750   NaN   NaN
There are 2 missing values.

```

EQTIL

Computes empirical quantiles.

Required Arguments

X— Vector of length *NOBS* containing the data. (Input)

NQPROP— Number of quantiles. (Input)
NQPROP must be greater than or equal to one.

QPROP— Vector of length *NQPROP* containing the quantile proportions. (Input)
The elements of *QPROP* must lie in the interval (0, 1).

Q— Vector of length *NQPROP* containing the empirical quantiles. (Output)
 $Q(i)$ corresponds to the empirical quantile at proportion $QPROP(i)$. The quantiles are determined by linear interpolation between adjacent ordered sample values.

XLO— Vector of length *NQPROP* containing the largest element of *X* less than or equal to the desired quantile. (Output)

XHI— Vector of length *NQPROP* containing the smallest element of *X* greater than or equal to the desired quantile. (Output)

NMISS— Number of missing values. (Output)

Optional Arguments

NOBS — Number of observations. (Input)
NOBS must be greater than or equal to one.
Default: NOBS = size (x,1).

FORTRAN 90 Interface

Generic: CALL EQTIL (X, NQPROP, QPROP, Q, XLO, XHI, NMISS [,...])

Specific: The specific interface names are S_EQTIL and D_EQTIL.

FORTRAN 77 Interface

Single: CALL EQTIL (NOBS, X, NQPROP, QPROP, Q, XLO, XHI, NMISS)

Double: The double precision name is DEQTIL.

Example

In this example, five empirical quantiles from a sample of size 30 are obtained. Notice that the 0.5 quantile corresponds to the sample median. The data are from Hinkley (1977) and Velleman and Hoaglin (1981). They are the measurements (in inches) of precipitation in Minneapolis/St. Paul during the month of March for 30 consecutive years.

```
USE EQTIL_INT
USE UMACH_INT
INTEGER NOBS, NQPROP
PARAMETER (NOBS=30, NQPROP=5)
!
INTEGER I, NMISS, NOUT
REAL QPROP(NQPROP), X(NOBS), XEMP(NQPROP), XHI(NQPROP), &
XLO(NQPROP)
!
DATA X/0.77, 1.74, 0.81, 1.20, 1.95, 1.20, 0.47, 1.43, 3.37, &
2.20, 3.00, 3.09, 1.51, 2.10, 0.52, 1.62, 1.31, 0.32, 0.59, &
0.81, 2.81, 1.87, 1.18, 1.35, 4.75, 2.48, 0.96, 1.89, 0.90, &
2.05/
DATA QPROP/0.01, 0.50, 0.90, 0.95, 0.99/
!
CALL UMACH (2, NOUT)
CALL EQTIL (X, NQPROP, QPROP, XEMP, XLO, XHI, NMISS)
WRITE (NOUT,99997)
99997 FORMAT (' Smaller Empirical Larger', /, &
' Quantile Datum Quantile Datum')
DO 10 I=1, NQPROP
WRITE (NOUT,99998) QPROP(I), XLO(I), XEMP(I), XHI(I)
10 CONTINUE
99998 FORMAT (4X, F4.2, 8X, F4.2, 8X, F4.2, 8X, F4.2)
WRITE (NOUT,99999) NMISS
99999 FORMAT (/, ' There are ', I2, ' missing values.')
END
```

Output

Quantile	Smaller Datum	Empirical Quantile	Larger Datum
0.01	0.32	0.32	0.32
0.50	1.43	1.47	1.51
0.90	3.00	3.08	3.09
0.95	3.37	3.99	4.75
0.99	4.75	4.75	4.75

There are 0 missing values.

Comments

1. Workspace may be explicitly provided, if desired, by use of `E2TIL/DE2TIL`. The reference is:

```
CALL E2TIL (NOBS, X, NQPROP, QPROP, Q, XLO, XHI,  
NMISS, WK)
```

The additional argument is:

WK — Workspace of length `NOBS` containing the sorted data. (Output)

If `X` is sorted in ascending order with all missing values at the end of `X`, then `X` and `WK` may share the same storage location.

2. Informational error
Type Code
3 1 All of the observations are missing values. The elements of `Q`, `XLO`, and `XHI` have been set to NaN (not a number).
3. Missing values (NaN) are excluded from the analysis. Empirical quantiles are based on the `NOBS - NMISS` nonmissing elements of `X`.

Description

The routine `EQTIL` determines the empirical quantiles, as indicated in the vector `QPROP`, from the data in `X`. The routine `EQTIL` first checks to see if `X` is sorted; if `X` is not sorted, the routine does either a complete or partial sort, depending on how many order statistics are required to compute the quantiles requested.

The routine `EQTIL` returns the empirical quantiles and, for each quantile, the two order statistics from the sample that are at least as large and at least as small as the quantile. For a sample of size n , the quantile corresponding to the proportion p is defined as

$$Q(p) = (1 - f)x_{(j)} + fx_{(j+1)}$$

where $j = \lfloor p(n + 1) \rfloor$, $f = p(n + 1) - j$, and $x_{(j)}$ is the j -th order statistic, if $1 \leq j < n$; otherwise, the empirical quantile is the smallest or largest order statistic.

TWOMV

Computes statistics for mean and variance inferences using samples from two normal populations.

Required Arguments

X— Vector of length `NROWX` containing observations from the first sample. (Input)

Y— Vector of length `NROWY` containing observations from the second sample. (Input)

STAT— Vector of length 25 containing the statistics.
(Output, if `IDO` = 0 or 1; input/output, if `IDO` = 2 or 3.) These are:

I **STAT(I)**

1 Mean of the first sample.

2 Mean of the second sample.

3 Variance of the first sample.

4 Variance of the second sample.

5 Number of observations in the first sample.

6 Number of observations in the second sample.

(*STAT*(7) through *STAT*(14) depend on the assumption of equal variances.)

7 Pooled variance.

8 *t* value, assuming equal variances.

9 Probability of a larger *t* in absolute value, assuming normality, equal means, and equal variances.

10 Degrees of freedom assuming equal variances.

11 Lower confidence limit for the mean of the first population minus the mean of the second, assuming equal variances.

12 Upper confidence limit for the mean of the first population minus the mean of the second, assuming equal variances.

13 Lower confidence limit for the common variance.

14 Upper confidence limit for the common variance.

(STAT(15) through STAT(19) use approximations that do not depend on an assumption of equal variances.)

- 15 t value, assuming unequal variances.
- 16 Approximate probability of a larger t in absolute value, assuming normality, equal means, and unequal variances.
- 17 Degrees of freedom assuming unequal variances, for Satterthwaite's approximation.
- 18 Approximate lower confidence limit for the mean of the first population minus the mean of the second, assuming equal variances.
- 19 Approximate upper confidence limit for the mean of the first population minus the mean of the second, assuming equal variances.
- 20 F value (greater than or equal to 1.0).
- 21 Probability of a larger F in absolute value, assuming normality and equal variances.
- 22 Lower confidence limit for the ratio of the variance of the first population to the second.
- 23 Upper confidence limit for the ratio of the variance of the first population to the second.
- 24 Number of missing values of first sample.
- 25 Number of missing values of second sample.

Optional Arguments

IDO — Processing option. (Input)
Default: IDO = 0.

IDO Action

- 0 This is the only invocation of TWOMV for this data set, and all the data are input at once.
- 1 This is the first invocation, and additional calls to TWOMV will be made. Initialization and updating are performed. The means are output correctly, but most of the other quantities output in STAT are intermediate quantities.
- 2 This is an intermediate invocation of TWOMV, and updating for the data in X and Y is performed.

- 3 This is the final invocation of this routine. Updating for the data in X and Y and wrap-up computations are performed.

NROWX — Absolute value of ***NROWX*** is the number of observations currently input in X .
(Input)
Default: ***NROWX*** = size (X ,1).
NROWX may be positive, zero, or negative. Negative ***NROWX*** means delete the $-NROWX$ observations in X from the analysis.

NROWY — Absolute value of ***NROWY*** is the number of observations currently input in Y .
(Input)
Default: ***NROWY*** = size (Y ,1).
NROWY may be positive, zero, or negative. Negative ***NROWY*** means delete the $-NROWY$ observations in Y from the analysis.

CONPRM — Confidence level for two-sided interval estimate of the mean of X minus the mean of Y (assuming normality of both populations), in percent. (Input)
Default: ***CONPRM*** = 95.0.
If ***CONPRM*** = 0, no confidence interval for the difference in the means is computed; otherwise, a ***CONPRM*** percent confidence interval is computed, in which case ***CONPRM*** must be between 0.0 and 100.0. ***CONPRM*** is often 90.0, 95.0, or 99.0. For a one-sided confidence interval with confidence level ***ONECL***, set
CONPRM = $100.0 - 2.0 * (100.0 - ONECL)$.

CONPRV — Confidence level for inference on variances. (Input)
Default: ***CONPRV*** = 95.0.
Under the assumption of equal variances, the pooled variance is used to obtain a two-sided ***CONPRV*** percent confidence interval for the common variance in ***STAT***(13) and ***STAT***(14). Without making the assumption of equal variances, the ratio of the variances is of interest. A two-sided ***CONPRV*** percent confidence interval for the ratio of the variance of the first population (X) to that of the second population (assuming normality of both populations) is computed and stored in ***STAT***(22) and ***STAT***(23). The confidence intervals are symmetric in probability. See also the description of ***CONPRM***.

IPRINT — Printing option. (Input)
If ***IPRINT*** = 0, no printing is performed; otherwise, various statistics in ***STAT*** are printed when ***IDO*** = 0 or 3.
Default: ***IPRINT*** = 0.

IPRINT **Action**

- 0 No printing.
- 1 Simple statistics (***STAT*** (1) to ***STAT***(6), ***STAT***(24), and ***STAT***(25)).
- 2 Statistics for means, assuming equal variances.

- 3 Statistics for means, not assuming equal variances.
- 4 Statistics for variances.
- 5 All statistics.

FORTRAN 90 Interface

Generic: CALL TWOMV (X, Y, STAT [,...])

Specific: The specific interface names are S_TWOMV and D_TWOMV.

FORTRAN 77 Interface

Single: CALL TWOMV (IDO, NROWX, X, NROWY, Y, CONPRM, CONPRV, IPRINT, STAT)

Double: The double precision name is DTWOMV.

Example 1

This example is taken from Conover and Iman (1983, page 294). It involves scores on arithmetic tests of two grade school classes. The question is whether a group taught by an experimental method has a higher mean score. The data are shown below.

Scores for Standard Group	Scores for Experimental Group
72	111
75	118
77	128
80	138
104	140
110	150
125	163
	164
	169

It is assumed that the variances of the two populations are equal so the statistics of interest are in STAT(8) and STAT(9). It is seen from the output below that there is strong reason to believe that the two means are different (t -value of -4.804). Since the lower 97.5% confidence limit does not include zero, the null hypothesis that $\mu_x \leq \mu_y$ would be rejected at the 0.05 significance level. (The closeness of the values of the sample variances provides some qualitative substantiation of the assumption of equal variances.)

```
USE TWOMV_INT
INTEGER      IPRINT
REAL        CONPRV, STAT(25), X(7), Y(9)
```

```

!
DATA X/72., 75., 77., 80., 104., 110., 125./Y/111., 118., 128., &
    138., 140., 150., 163., 164., 169./
!

IPRINT = 2
CONPRV = 0.0
CALL TWOMV (X, Y, STAT, IPRINT=IPRINT, CONPRV=CONPRV)
END

```

Output

```

Mean Inferences Assuming Equal Variances
Pooled Variance                434.633
t Value                         -4.804
Probability of a Larger t in Abs. Value  0.000
Degrees of Freedom              14.000
Lower Confidence Limit Difference in Means -73.010
Upper Confidence Limit Difference in Means -27.942

```

Description

The routine `TWOMV` computes the statistics for making inferences about the means and variances of two normal populations, using independent samples in `X` and `Y`. For inferences concerning parameters of a single normal population, see routine `UVSTA` ([page 18](#)). For two samples that are paired, see routine `ATWOB` (see Chapter 3, Correlation), since the pairs can be considered to be blocks.

Let μ_X and

$$\sigma_X^2$$

be the mean and variance, respectively, of the first population, and μ_Y and

$$\sigma_Y^2$$

be the corresponding quantities of the second population. The routine `TWOMV` is used for testing $\mu_X = \mu_Y$ and

$$\sigma_X^2 = \sigma_Y^2$$

or for setting confidence intervals for $\mu_X - \mu_Y$ and

$$\sigma_X^2 / \sigma_Y^2$$

The basic quantities in `STAT(1)` through `STAT(4)` are

$$\bar{x} = \sum_{i=1}^{n_x} x_i / n_x, \quad \bar{y} = \sum_{i=1}^{n_y} y_i / n_y$$

$$s_x^2 = \sum_{i=1}^{n_x} (x_i - \bar{x})^2 / (n_x - 1), \quad \text{and} \quad s_y^2 = \sum_{i=1}^{n_y} (y_i - \bar{y})^2 / (n_y - 1)$$

where n_x and n_y are the respective sample sizes (in `STAT(5)` and `STAT(6)`).

Inferences about the Means

The test for the equality of means of two normal populations depends on whether or not the variances of the two populations can be considered equal. If the variances are equal, the test is the *two-sample t test*, which is equivalent to an analysis of variance test (see Chapter 4). In this case, the statistics returned in STAT(7) through STAT(12) are appropriate for testing $\mu_X = \mu_Y$.

The pooled variance (in STAT(7)) is

$$s^2 = \frac{(n_x - 1)s_x^2 + (n_y - 1)s_y^2}{n_x + n_y - 2}$$

The t statistic (in STAT(8)) is

$$t = \frac{\bar{x} - \bar{y}}{s\sqrt{(1/n_x) + (1/n_y)}}$$

For testing $\mu_X = \mu_Y + c$, for some constant c , the confidence interval for $\mu_X - \mu_Y$ can be used. (If the confidence interval includes c , the null hypothesis would not be rejected at the significance level $1 - \text{CONPRM}/100$.)

If the population variances are not equal, the ordinary t statistic does not have a t distribution; and several approximate tests for the equality of means have been proposed. (See, for example, Anderson and Bancroft 1952, and Kendall and Stuart 1979.) The name *Fisher-Behrens* is associated with this problem, and one of the earliest tests devised for this situation is the Fisher-Behrens test, based on Fisher's concept of *fiducial probability*. Another test is called *Satterthwaite's procedure*. The routine TWOMV computes the statistics for this approximation, which was suggested by H.F. Smith and modified by F.E. Satterthwaite (Anderson and Bancroft 1952, page 83). The test statistic is

$$t' = (\bar{x} - \bar{y}) / s_d$$

where

$$s_d = \sqrt{(s_x^2 / n_x) + (s_y^2 / n_y)}$$

Under the null hypothesis of equal population means, this quantity has an approximate t distribution with degrees of freedom f (in STAT(17)), given by

$$f = \frac{s_d^4}{\frac{(s_x^2 / n_x)^2}{n_x - 1} + \frac{(s_y^2 / n_y)^2}{n_y - 1}}$$

Inferences about the Variances

The F statistic for testing the equality of variances is given by

$$F = s_1^2 / s_2^2, \text{ where } s_1^2$$

is the larger of

s_x^2 and s_y^2 , and s_2^2

is the smaller. If the variances are equal, this quantity has an F distribution with $n_x - 1$ and $n_y - 1$ degrees of freedom.

It is generally not recommended that the results of the F test be used to decide whether to use the regular t test or the modified t' on a single set of data. The more conservative approach is to use the modified t' (Satterthwaite's procedure) if there is doubt about the equality of the variances.

Additional Example

Example 2

For a second example, the same data set is used to illustrate the use of the `IDO` parameter to bring in the data one observation at a time. Since there are more "Y" values than "X" values, `NROWX` is set to zero on the later calls to `TWOMV`.

```
USE TWOMV_INT
INTEGER I, IDO, IPRINT, NROWX, NROWY
REAL STAT(25), X(7), Y(9)
!
DATA X/72., 75., 77., 80., 104., 110., 125./Y/111., 118., 128., &
    138., 140., 150., 163., 164., 169./
!
IPRINT = 5
IDO = 1
NROWX = 1
NROWY = 1
DO 10 I=1, 7
!
!                               Bring in first seven observations
!                               on X and Y, one at a time.
    CALL TWOMV (X(I:), Y(I:), STAT, IDO=IDO, NROWX=NROWX, &
        NROWY=NROWY, IPRINT=IPRINT)
    IDO = 2
10 CONTINUE
!
!                               Now bring in remaining observations
!                               on Y.
NROWX = 0
CALL TWOMV (X(1:), Y(8:), STAT, IDO=IDO, NROWX=NROWX, &
    NROWY=NROWY, IPRINT=IPRINT)
!
!                               Set IDO to indicate last observation.
IDO = 3
CALL TWOMV (X(1:), Y(9:), STAT, IDO=IDO, NROWX=NROWX, &
    NROWY=NROWY, IPRINT=IPRINT)
END
```

Output

```
Statistics from TWOMV
First Sample Mean          91.857
Second Sample Mean        142.333
First Sample Variance     435.810
Second Sample Variance    433.750
```

First Sample Valid Observations	7.000
Second Sample Valid Observations	9.000
First Sample Missing Values	0.000
Second Sample Missing Values	0.000

Mean Inferences Assuming Equal Variances

Pooled Variance	434.63
t Value	-4.80
Probability of a Larger t in Abs. Value	0.00
Degrees of Freedom	14.00
Lower Confidence Limit Difference in Means	-73.01
Upper Confidence Limit Difference in Means	-27.94
Lower Confidence Limit for Common Variance	232.97
Upper Confidence Limit for Common Variance	1081.04

Mean Inferences Assuming Unequal Variances

t Value	-4.8028
Approx. Prob. of a Larger t in Abs. Value	0.0003
Degrees of Freedom	13.0290
Lower Confidence Limit	-73.1758
Upper Confidence Limit	-27.7766

Variance Inferences

F Value	1.00475
Probability of a Larger F in Abs. Value	0.96571
Lower Confidence Limit for Variance Ratio	0.21600
Upper Confidence Limit for Variance Ratio	5.62621

BINES

Estimates the parameter p of the binomial distribution.

Required Arguments

N — Total number of Bernoulli trials. (Input)

N is the parameter *N* in the binomial distribution from which one observation (κ) has been drawn.

K — Number of successes in the *N* trials. (Input)

CONPER — Confidence level for two-sided interval estimate, in percent. (Input)

An approximate *CONPER* percent confidence interval is computed, hence, *CONPER* must be between 0.0 and 100.0. *CONPER* often will be 90.0, 95.0, or 99.0. For a one-sided confidence interval with confidence level *ONECL*, set *CONPER* = 100.0 - 2.0 * (100.0 - *ONECL*).

PHAT — Estimate of p . (Output)

PLOWER — Lower confidence limit for p . (Output)

PUPPER — Upper confidence limit for p . (Output)

FORTRAN 90 Interface

Generic: CALL BINES (N, K, CONPER, PHAT, PLOWER, PUPPER)

Specific: The specific interface names are S_BINES and D_BINES.

FORTRAN 77 Interface

Single: CALL BINES (N, K, CONPER, PHAT, PLOWER, PUPPER)

Double: The double precision name is DBINES.

Example

In this example, we assume that the number of defective microchips in a given lot follows a binomial distribution. We estimate the proportion defective by taking a sample of 50. In this sample, 3 microchips were found to be defective. The routine BINES is used to estimate p and to compute a 95% confidence interval.

```
USE BINES_INT
USE UMACH_INT
INTEGER K, N, NOUT
REAL CONPER, PHAT, PLOWER, PUPPER
!
CALL UMACH (2, NOUT)
N = 50
K = 3
CONPER = 95.0
CALL BINES (N, K, CONPER, PHAT, PLOWER, PUPPER)
WRITE (NOUT,99999) PHAT, PLOWER, PUPPER
99999 FORMAT (' Point estimate of the proportion: ', F5.3, '/', &
' 95% confidence interval: (', F5.3, ',', F5.3, &
')')
END
```

Output

```
Point estimate of the proportion: .060
95% confidence interval: ( .013, .165)
```

Comments

1. Informational errors
Type Code
3 1 CONPER is 100.0 or too large for accurate computations. The confidence limits are set to 0.0 and 1.0.
3 2 CONPER is 0.0 or too small for accurate computations. The confidence limits are both set to PHAT.

2. Since the binomial is a discrete distribution, it is not possible to construct an exact CONPER% confidence interval for all values of CONPER. Let $\alpha = 1 - \text{CONPER}/100$. Then, the approximate lower and upper confidence limits p_L and p_U (PLOWER and PUPPER) are solutions to the equations

$$\sum_{x=K}^N \binom{N}{x} p_L^x (1-p_L)^{N-x} = \alpha/2$$

$$\sum_{x=0}^K \binom{N}{x} p_U^x (1-p_U)^{N-x} = \alpha/2$$

These approximations are not just computational devices. Approximations to the confidence limits are necessary because the binomial distribution is discrete.

Description

The routine `BINES` computes a point estimate and a confidence interval for the parameter, p , of a binomial distribution, using the number of “successes”, K , in a sample of size N from a binomial distribution with probability function

$$f(x) = \binom{N}{x} p^x (1-p)^{N-x} \quad \text{for } x = 0, 1, \dots, N$$

The point estimate for p is merely K/N .

The routine `BINES` makes use of the relationship between the binomial distribution and the beta distribution (see Johnson and Kotz 1969, Chapter 3) by solving the following equations equivalent to those in Comment 2:

$$p_L = \beta_{K, N-K+1, \alpha/2}$$

$$p_U = \beta_{K+1, N-K, 1-\alpha/2}$$

where $\beta_{a, b, \tau}$ is the beta τ critical value with parameters a and b (that is, the inverse beta distribution function evaluated at $1 - \tau$). The routine `BETIN` (see Chapter 17, Probability Distribution Function and Inverses) is used to evaluate the critical values.

POIES

Estimates the parameter of the Poisson distribution.

Required Arguments

IX — Vector of length `NOBS` containing the data. (Input)

The data are assumed to be a random sample from a Poisson distribution; hence, all elements of `IX` must be nonnegative.

CONPER — Confidence level for two-sided interval estimate, in percent. (Input)
 An approximate CONPER percent confidence interval is computed; hence, CONPER must be between 0.0 and 100.0. CONPER often will be 90.0, 95.0, or 99.0. For a one sided confidence interval with confidence level ONECL, set $CONPER = 100.0 - 2.0 * (100.0 - ONECL)$.

THAT — Estimate of the parameter, theta (the mean). (Output)

TLOWER — Lower confidence limit for theta. (Output)

TUPPER — Upper confidence limit for theta. (Output)

Optional Arguments

NOBS — Number of observations. (Input)
 Default: NOBS = size (IX,1).

FORTRAN 90 Interface

Generic: CALL POIES (IX, CONPER, THAT, TLOWER, TUPPER [,...])

Specific: The specific interface names are S_POIES and D_POIES.

FORTRAN 77 Interface

Single: CALL POIES (NOBS, IX, CONPER, THAT, TLOWER, TUPPER)

Double: The double precision name is DPOIES.

Example

It is assumed that flight arrivals at a major airport during the middle of the day follow a Poisson distribution. It is desired to estimate the mean number of arrivals per minute and to obtain an upper one-sided 95% confidence interval for the mean. During a half-hour period, the number of arrivals each minute was recorded. These data are stored in IX, and POIES is used to obtain the estimates.

```

USE POIES_INT
USE UMACH_INT
INTEGER NOBS
PARAMETER (NOBS=30)
!
INTEGER IX(NOBS), NOUT
REAL CONPER, THAT, TLOWER, TUPPER
!
DATA IX/2, 0, 1, 1, 2, 0, 3, 1, 2, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, &
    0, 1, 2, 0, 2, 0, 0, 1, 2, 0, 2/
!
CALL UMACH (2, NOUT)
!
For a 95 percent one-sided !.I.,

```

```

!                                     CONPER = 100.0 - 2.0*(100.0-95.0)
CONPER = 90.0
CALL POIES (IX, CONPER, THAT, TLOWER, TUPPER)
WRITE (NOUT,99999) THAT, TUPPER
99999 FORMAT (' Point estimate of the Poisson mean: ', F5.3, '/', &
             ' Upper one-sided 95% confidence limit: ', F5.3)
END

```

Output

```

Point estimate of the Poisson mean: 0.800
Upper one-sided 95% confidence limit: 1.125

```

Comments

1. Informational error
Type Code
3 1 CONPER is 0.0 or too small for accurate computations. The confidence limits are both set to THAT.
2. Since the Poisson is a discrete distribution, it is not possible to construct an exact CONPER% confidence interval for all values of CONPER. Let $\alpha = 1 - \text{CONPER}/100$, and let k be a single observation. Then, the approximate lower and upper confidence limits θ_L and θ_U (TLOWER and TUPPER) are solutions to the equations

$$\exp(-\theta_L) \sum_{x=k}^{\infty} \theta_L^x / x! = \alpha/2$$

$$\exp(-\theta_U) \sum_{x=0}^k \theta_U^x / x! = \alpha/2$$

Description

The routine POIES computes a point estimate and a confidence interval for the parameter, θ , of a Poisson distribution. It is assumed that the vector IX contains a random sample of size NOBS from a Poisson distribution with probability function

$$f(x) = e^{-\theta} \theta^x / x!, \text{ for } x = 0, 1, 2, \dots$$

The point estimate for θ corresponds to the sample mean.

By exploiting the relationship between the Poisson distribution and the chi-squared distribution (see Johnson and Kotz, 1969, Chapter 4), the equations in Comment 2 can be written as

$$\theta_L = \frac{1}{2} \chi_{2k, \alpha/2}^2$$

$$\theta_U = \frac{1}{2} \chi_{2k+2, 1-\alpha/2}^2$$

where

$$\chi_{\nu, \tau}^2$$

is the chi-squared τ critical value with degrees ν of freedom (that is, the inverse chi-squared distribution function evaluated at $1 - \tau$). The routine `CHIIN` (see Chapter 17, Probability Distribution Functions and Inverses) is used to evaluate the critical values.

For more than one observation, the estimates are obtained as above and then divided by the number of observations, `NOBS`.

NRCES

Computes maximum likelihood estimates of the mean and variance from grouped and/or censored normal data.

Required Arguments

XRT — Vector of length `NOBS` containing either the exact value of the data or the right endpoint of the censoring interval for interval-censored or right-censored data. (Input)
See the argument `ICEN`.

XLT — Vector of length `NOBS` containing the left endpoint of the censoring interval for interval-censored or left-censored data. (Input)
See the argument `ICEN`. *XLT* is not used if there is no left censoring.

ICEN — Vector of length `NOBS` containing the censoring codes. (Input)
The values in `ICEN` indicate the meaning of the values in *XRT* and/or *XLT*.

ICEN(I) Censoring

- 0 Exact response at *XRT*(*I*).
- 1 Right censored. The response is greater than *XRT*(*I*).
- 2 Left censored. The response is less than or equal to *XLT*(*I*).
- 3 Interval censored. The response is greater than *XRT*(*I*), but less than or equal to *XLT*(*I*).

XMEAN — Estimate of the mean. (Input/Output if `INIT = 0`; output otherwise)

XSIGMA — Estimate of the standard deviation. (Input/Output if `INIT = 0`; output otherwise)

VXM — Estimate of the variance of the mean estimate. (Output)

VXS — Estimate of the variance of the variance estimate. (Output)

COVXMS — Estimate of the covariance of the mean and the variance estimates. (Output)

NUMBER — Vector of length 4 containing the numbers of observations having the various censoring properties. (Output)
NUMBER(1) is the number of exact observations. NUMBER(2) is the number of observations specified by a lower bound (right censored). NUMBER(3) is the number of observations specified by an upper bound (left censored). NUMBER(4) is the number of observations specified by an interval.

Optional Arguments

NOBS — Number of observations. (Input)
Default: NOBS = size (XRT,1).

EPSM — Convergence criterion for the mean estimate. (Input)
See the argument EPSSIG. If EPSM is not positive, EPSM = 0.00001 is assumed.
Default: EPSM = .00001.

EPSSIG — Convergence criterion for the variance estimate. (Input)
Convergence is assumed when the relative change in the mean estimate is less than EPSM and the relative change in the variance estimate is less than EPSSIG. If EPSSIG is not positive, EPSSIG = 0.00001 is assumed.
Default: EPSSIG = .00001.

MAXITS — Maximum number of iterations allowed. (Input)
A typical value of MAXITS is 25.
Default: MAXITS = 25.

INIT — Initialization option. (Input)
Default: INIT = 1.

INIT Action

- 0 On input, XMEAN and XSIGMA contain initial estimates of the parameters.
- 1 If there are enough exactly specified data, initial estimates are obtained from it; and, if there are not enough such data, fixed starting values (XRT(1) for the mean and 1.0 for the variance) are used.

FORTRAN 90 Interface

Generic: CALL NRCES (XRT, XLT, ICEN, XMEAN, XSIGMA, VXM, VXS, COVXMS, NUMBER [, ...])

Specific: The specific interface names are S_NRCES and D_NRCES.

FORTRAN 77 Interface

Single: CALL NRCES (NOBS, XRT, XLT, ICEN, EPSM, EPSSIG, MAXITS, INIT, XMEAN, XSIGMA, VXM, VXS, COVXMS, NUMBER)

Double: The double precision name is DNRCES.

Example

This example uses an artificial data set consisting of 18 observations. The first 12 observations are known exactly; the next three are known only by a lower bound; the next two, by an upper bound; and the last one, by two bounds.

```
USE NRCES_INT
USE UMACH_INT
INTEGER NOBS
PARAMETER (NOBS=18)
!
INTEGER ICEN(NOBS), INIT, MAXITS, NOUT, NUMBER(4)
REAL COVXMS, EPSM, EPSSIG, VXM, VXS, XLT(NOBS), XMEAN, &
XRT(NOBS), XSIGMA
!
DATA XRT/4.5, 5.4, 3.9, 5.1, 4.6, 4.8, 2.9, 6.3, 5.5, 4.6, 4.1, &
5.2, 3.2, 4.0, 3.1, 0.0, 0.0, 2.2/
DATA XLT/0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, &
0.0, 0.0, 0.0, 0.0, 5.1, 3.8, 2.5/
DATA ICEN/0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 3/
!
CALL UMACH (2, NOUT)
EPSM = 0.01
EPSSIG = 0.01
MAXITS = 25
INIT = 1
CALL NRCES (XRT, XLT, ICEN, XMEAN, XSIGMA, VXM, VXS, COVXMS, &
NUMBER, EPSM=EPSM, EPSSIG=EPSSIG)
WRITE (NOUT,99999) XMEAN, XSIGMA, VXM, VXS, COVXMS, NUMBER
99999 FORMAT (' Estimate of mean: ', F8.4, &
/, ' Estimate of variance: ', F8.4, &
/, ' Estimate of variance of mean estimate: ', F8.4, &
/, ' Estimate of variance of variance estimate: ', F8.4, &
/, ' Estimate of covariance of mean and variance: ', F8.4, &
/, ' Number of exact observations: ', I4, &
/, ' Number of right-censored observations: ', I4, &
/, ' Number of left-censored observations: ', I4, &
/, ' Number of interval-censored observations: ', I4)
END
```

Output

```
Estimate of mean: 4.4990
Estimate of standard deviation: 1.2304
Estimate of variance of mean estimate: 0.0819
Estimate of variance of variance estimate: -0.0494
Estimate of covariance of mean and variance: -0.0019
Number of exact observations: 12
Number of right-censored observations: 3
Number of left-censored observations: 2
Number of interval-censored observations: 1
```

Description

The routine `NRCES` computes maximum likelihood estimates of the mean and variance of a normal population, using a sample that may be censored. An observation whose value is known exactly is input in `XRT`, and the corresponding element in `ICEN` is set to 0. If an observation is known only by a lower bound, we say the observation is *right censored*; the lower bound is input in `XRT`, and the corresponding element in `ICEN` is set to 1. If an observation is known only by an upper bound, we say the observation is *left censored*; the upper bound is input in `XLT`, and the corresponding element in `ICEN` is set to 2. If an observation is known only by two bounds, we say the observation is *interval censored*; the lower bound is input in `XRT`, the upper bound is input in `XLT`, and the corresponding element in `ICEN` is set to 3.

Newton-Raphson iterations are used to find a stationary point of the likelihood function, and the Hessian at that point is used to estimate the variances and covariance of the estimates of the population mean and variance. If the numerical derivative of the estimate of the variance increases on nine consecutive iterations, the process is deemed divergent and a terminal error is issued. The iterations begin at user-supplied values if `INIT` is set to 0.

GRPES

Computes basic statistics from grouped data.

Required Arguments

TABLE — Vector of length `NGROUP` containing the frequencies within the groups. (Input)
The entries in `TABLE` are interpreted as counts. They must be nonnegative.

CLOW — The center (class mark) of the lowest class interval. (Input)

CWIDTH — The class width. (Input)
`CWIDTH` must be positive.

STAT — Vector of length 13 containing the statistics. (Output)

I **STAT(I)**

1 The sum of the frequencies in `TABLE`.

2 Mean (arithmetic mean, first moment).

3 Sample standard deviation. (Uses `STAT(1) - 1` as divisor).

4 Second moment about the mean, uncorrected for grouping. (Uses `STAT(1)` as divisor.)

5 Second moment about the mean, adjusted using Sheppard's correction.

6 Third moment about the mean, uncorrected for grouping.

- 7 Third moment about the mean, adjusted using Sheppard's correction.
- 8 Fourth moment about the mean, uncorrected for grouping.
- 9 Fourth moment about the mean, adjusted using Sheppard's correction.
- 10 Median.
- 11 Geometric mean; defined only if $CLOW - CWIDTH/2$ is nonnegative.
- 12 Harmonic mean; defined only if $CLOW - CWIDTH/2$ is nonnegative.
- 13 Mode; defined only if one element of `TABLE` is strictly greater than all other elements of `TABLE`.

Optional Arguments

NGROUP— Number of groups. (Input)
 Default: `NGROUP = size(TABLE,1)`.

IPRINT— Printing option. (Input)
 If `IPRINT = 0`, no printing is performed; and if `IPRINT = 1`, the statistics in `STAT` are printed.
 Default: `IPRINT = 0`.

FORTRAN 90 Interface

Generic: `CALL GRPES (TABLE, CLOW, CWIDTH, STAT [, ...])`

Specific: The specific interface names are `S_GRPES` and `D_GRPES`.

FORTRAN 77 Interface

Single: `CALL GRPES (NGROUP, TABLE, CLOW, CWIDTH, IPRINT, STAT)`

Double: The double precision name is `DGRPES`.

Example 1

This example is taken from Conover and Iman (1983, page 119). The objective is to compute some basic statistics relating to test scores, using the following data:

Score	Frequency
91–100	7
81–90	13
71–80	11

61-70	5
≤ 60	4

```

USE GRPES_INT
INTEGER      IPRINT, NGROUP
REAL         CLOW, CWIDTH, STAT(13), TABLE(5)
!
NGROUP      = 5
CLOW        = 55.5
CWIDTH      = 10.0
TABLE(1)    = 4.0
TABLE(2)    = 5.0
TABLE(3)    = 11.0
TABLE(4)    = 13.0
TABLE(5)    = 7.0
IPRINT      = 1
CALL GRPES (TABLE, CLOW, CWIDTH, STAT, IPRINT=IPRINT)
END

```

Output

```

Statistics from GRPES
Sum freqs.      40.0
Mean            79.0
Std. dev.       12.1
2nd moment     142.8
2nd, adj.      134.4
3rd moment     -741.8
3rd, adj.     -2716.8
4th moment     48242.3
4th, adj.     47929.0
Median         80.5
Geometric      78.0
Harmonic       77.0
Mode           85.5

```

Description

The routine `GRPES` computes various statistics using data from equally spaced groups. The second, third, and fourth moments are computed both with and without Sheppard's corrections. These corrections for grouped data are most useful for distributions whose densities tail off smoothly (such as the normal distribution). Kendall, Stuart, and Ord (1987, Chapters 2 and 3) discuss these corrections.

The moments are computed using the sum of the frequencies as the divisor. The standard deviation (`STAT(3)`), on the other hand, is computed using as the divisor the sum of the frequencies minus one.

If any of the class marks are negative, the geometric and harmonic means are not computed, and NaN (not a number) is stored as the value of `STAT(11)`. Likewise, if the mode does not exist (no group has a frequency greater than that of all other groups), NaN is stored as the value of `STAT(13)`.

Additional Example

Example 2

In this example, there are negative values of some class marks, and there is no modal class.

Class Marks	Frequency
-2.0	2
-1.0	5
0.0	7
1.0	7
2.0	2

```
USE GRPES_INT
INTEGER   NGROUP, IPRINT
REAL     TABLE(5), CLOW, CWIDTH, STAT(13)
!
NGROUP = 5
CLOW = -2.0
CWIDTH = 1.0
TABLE(1) = 2.0
TABLE(2) = 5.0
TABLE(3) = 7.0
TABLE(4) = 7.0
TABLE(5) = 2.0
IPRINT = 1
CALL GRPES (TABLE, CLOW, CWIDTH, STAT, IPRINT=IPRINT)
END
```

Output

```
Statistics from GRPES
Sum freqs.      23.0000
Mean            0.0870
Std. dev.       1.1246
2nd moment      1.2098
2nd, adj.       1.1265
3rd moment      -0.2293
3rd, adj.       -0.2510
4th moment      3.3292
4th, adj.       2.7960
Median          0.1429
```

The mode is not defined, since no class has higher frequency than all others.
The geometric and harmonic means are not defined, since the lower bound is negative.

CSTAT

Computes cell frequencies, cell means, and cell sums of squares for multivariate data.

Required Arguments

X — |NROW| by NCOL matrix containing the data. (Input)
Each column of *X* represents either a classification variable, a response variable, a weight, or a frequency.

KMAX — Maximum number of cells. (Input)
This quantity does not have to be exact, but must be at least as large as the actual number of cells, *K*.

CELIF — Matrix with min(*KMAX*, *K*) columns containing cell information.
(Output, if *IDO* = 0 or 1; input/output, if *IDO* = 2 or 3.)
The number of rows in *CELIF* depends on the eight cases tabled below.

Case	Description	Rows in <i>CELIF</i>
1	MOPT ≤ 0, IFRQ = 0 and IWT = 0	NCOL + NR + 1
2	MOPT ≤ 0, IFRQ > 0 and IWT = 0	NCOL + NR
3	MOPT ≤ 0, IFRQ = 0 and IWT > 0	NCOL + NR + 1
4	MOPT ≤ 0, IFRQ > 0 and IWT > 0	NCOL + NR
5	MOPT > 0, IFRQ = 0 and IWT = 0	NCOL + 2 * NR + 1
6	MOPT > 0, IFRQ > 0 and IWT = 0	NCOL + 2 * NR
7	MOPT > 0, IFRQ = 0 and IWT > 0	NCOL + 3 * NR
8	MOPT > 0, IFRQ > 0 and IWT > 0	NCOL + 3 * NR - 1

Each column contains information on each unique combination of values of the *m* classification variables that occurs in the data. The first *m* rows give the values of the classification variables. Row *m* + 1 gives the number of observations that are in this cell. (For cases 2, 4, 6 and 8, this is the sum of the frequencies.) For case 3 and 4, row *m* + 2 contains the sum of the weights. For *NR* greater than zero, the remaining rows (beginning with row *m* + 3 in case 3 and 4 and with row *m* + 2 otherwise) contain information concerning the response variables. For cases 1, 2, 3 and 4, there are 2 * *NR* remaining rows with the cell (weighted) mean and cell (weighted) sum of squares for each of the *NR* response variables. For cases 5 and 6, there are 3 * *NR* remaining rows with the sample size, the mean and sum of squares for each of the *NR* response variables. For case 7 and 8, there are 4 * *NR* remaining rows with the sample size, the sum of weights, weighted means, and weighted sum of squares for each of the *NR* response variables.

Optional Arguments

IDO — Processing option. (Input)
Default: *IDO* = 0.

IDO Action

- 0 This is the only invocation of `CSTAT` for this data set, and all the data are input at once.
- 1 This is the first invocation, and additional calls to `CSTAT` will be made. Initialization and updating for the data in `X` are performed.
- 2 This is an intermediate invocation of `CSTAT`, and updating for the data in `X` is performed.

NROW — The absolute value of `NROW` is the number of rows of data currently input in `X`. (Input)
Default: `NROW = size (X,1)`.
`NROW` may be positive or negative. Negative `NROW` means that the `-NROW` rows of data are to be deleted from some aspects of the analysis, and this should be done only if `IDO` is 2. When a negative value is input for `NROW`, it is assumed that each of the `-NROW` rows of `X` has been input (with positive `NROW`) in previous invocations of `CSTAT`.

NCOL — Number of columns in `X`. (Input)
Default: `NCOL = size (X,2)`.

LDX — Leading dimension of `X` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDX = size (X,1)`.

NR — Number of response variables. (Input)
`NR = 0` means no response variables are input. Otherwise, cell means and sums of squares are computed for the response variables.
Default: `NR = 0`.

IRX — Vector of length `NR`. (Input if `NR` is greater than 0.)
The `IRX(1), ..., IRX(NR)` columns of `X` contain the response variables for which cell means and sums of squares are computed.

IFRQ — Frequency option. (Input)
`IFRQ = 0` means that all frequencies are 1.0. For positive `IFRQ`, column number `IFRQ` of `X` contains the frequencies.
Default: `IFRQ = 0`.

IWT — Weighting option. (Input)
`IWT = 0` means that all weights are 1.0. For positive `IWT`, column `IWT` of `X` contains the weights.
Default: `IWT = 0`.

MOPT — Missing value option. (Input)
If `MOPT` is zero, the exclusion is listwise. If `MOPT` is positive, the following occurs: (1)

if a classification variable's value is missing, the entire case is excluded, (2) if $IFRQ > 0$ and the frequency variable's value is missing, the entire case is excluded, (3) if $IWT > 0$ and the weight variable's value is missing, the case is classified and the cell frequency updated, but no information with regard to the response variables is computed, and (4) if only some response variables' values are missing, all computations are performed except those pertaining to the response variables with missing values.
 Default: $MOPT = 0$.

K — Number of cells or an upper bound for this number. (Input/Output)
 On the first call K must be input $K = 0$. It should not be changed between calls to `CSTAT`. K is incremented by one for each new cell up to $KMAX$ cells. Once $KMAX$ cells are encountered, K is incremented by one for each observation that does not fall into one of the $KMAX$ cells. In this case, K is an upper bound on the number of cells and can be used for $KMAX$ in a subsequent run.
 Default: $K = 0$.

LDCELI — Leading dimension of `CELIF` exactly as specified in the dimension statement in the calling program. (Input)
 Default: $LDCELI = \text{size}(\text{CELIF}, 1)$.

FORTRAN 90 Interface

Generic: `CALL CSTAT (X, KMAX, CELIF [, ...])`

Specific: The specific interface names are `S_CSTAT` and `D_CSTAT`.

FORTRAN 77 Interface

Single: `CALL CSTAT (IDO, NROW, NCOL, X, LDX, NR, IRX, IFRQ, IWT, MOPT, KMAX, K, CELIF, LDCELI)`

Double: The double precision name is `DCSTAT`.

Example 1

In this example, there are two classification variables, C_1 and C_2 , and two response variables, R_1 and R_2 . Their values are shown below.

		C_1			
		1			2
		R_1	R_2	R_1	R_2
C_2	1	5.0	3.4	3.8	2.4
		7.0	2.6	5.2	6.3
				4.9	1.2

2

R_1	R_2	R_1	R_2
4.3	9.8	6.5	3.4
3.2	7.1	3.1	5.1
1.7	6.3		

```

USE CSTAT_INT
USE WRRRL_INT

INTEGER    KMAX, LDCELI, LDX, NR, NCOL
PARAMETER (KMAX=4, LDCELI=15, LDX=10, NR=2, NCOL=4)
!
INTEGER    IDO, IFRQ, IRX(NR), IWT, K, MIN0, MOPT, NROW
REAL       CELIF(LDCELI,KMAX), X(LDX,NCOL)
CHARACTER  CLABEL(1)*6, FMT*7, RLABEL(7)*6
INTRINSIC  MIN0
!
                                Get data for example
DATA X/1.0, 1.0, 1.0, 1.0, 1.0, 2.0, 2.0, 2.0, 2.0, 2.0, 1.0, &
      1.0, 2.0, 2.0, 2.0, 1.0, 1.0, 1.0, 2.0, 2.0, 5.0, 7.0, 4.3, &
      3.2, 1.7, 3.8, 5.2, 4.9, 6.5, 3.1, 3.4, 2.6, 9.8, 7.1, 6.3, &
      2.4, 6.3, 1.2, 3.4, 5.1/
!
                                All data are input at once
IDO = 0
NROW = 10
K = 0
!
                                No unequal frequencies or weights
                                are used
IFRQ = 0
IWT = 0
!
                                Response variables are in 3rd and 4th
                                columns
IRX(1) = 3
IRX(2) = 4
!
                                Delete any row containing a missing
                                value
MOPT = 0
!
CALL CSTAT (X, KMAX, CELIF, NR=NR, IRX=IRX, K=K)
!
                                Print the results
CLABEL(1) = 'NONE'
RLABEL(1) = ' '
RLABEL(2) = ' '
RLABEL(3) = 'Freq.'
RLABEL(4) = 'Mean 1'
RLABEL(5) = 'SS 1'
RLABEL(6) = 'Mean 2'
RLABEL(7) = 'SS 2'
FMT = ' (W10.4) '
CALL WRRRL ('Statistics for the Cells', CELIF, &
            RLABEL, CLABEL, NRA=(NCOL+NR+1), &
            NCA=MIN0(KMAX, K), FMT=FMT)
END

```

Output

		Statistics for the Cells			
		1.00	1.00	2.00	2.00
		1.00	2.00	1.00	2.00
Freq.		2.00	3.00	3.00	2.00
Mean 1		6.00	3.07	4.63	4.80
SS 1		2.00	3.41	1.09	5.78
Mean 2		3.00	7.73	3.30	4.25
SS 2		0.32	6.73	14.22	1.44

Comments

1. If no nonmissing observations with positive weights or frequencies exist in a cell for a particular response variable, the mean and sum of squares are set to NaN (not a number).
2. In cases 3 and 6, if a zero weight is encountered, there is no contribution to the means or sums of squares, but the sample sizes are implemented by one for that observation.

Description

The routine `CSTAT` computes cell frequencies, cell means, and cell sums of squares for multivariate data in `X`. The columns of `X` can contain data for four types of variables: classification variables, a frequency variable, a weight variable, and response variables. The frequency variable, the weight variable, and the response variables are all designated by indicators in `IFRQ`, `IWT`, and `IRX`. All other variables are considered to be classification variables; hence, there are m classification variables, where $m = \text{NCOL} - \text{NR}$ if there is no weight or frequency variable, $m = \text{NCOL} - \text{NR} - 1$ if there is a weight or frequency variable but not both, and $m = \text{NCOL} - \text{NR} - 2$ if there are weight and frequency variables.

Each combination of values of the classification variables is stored in the first m rows of `CELIF`. For each combination of values of the classification variables, the frequencies are stored in the next row of `CELIF`. Then, for each combination, means and sums of squares for each of the response variables are computed and stored in the remaining rows of `CELIF`. If a weighting variable is specified, the sum of the weights for each combination is computed and stored. If missing values are deleted elementwise (that is, if `MOPT` is positive), the frequencies and sums of weights for each of the response variables are stored in the rows of `CELIF`.

Additional Example

Example 2

This example uses the same data as in the first example, except some of the data are set to missing values. Also, a frequency variable is used. It is in the fourth column of `X`. The frequency variable indicates that the values of the classification and response variables in the first observation occur 3 times and that all other frequencies are 1. Since `MOPT` is greater than zero, statistics on one response variable are accumulated even if the other response variable has a missing value. If the frequency variable has a missing value, however, the entire observation is omitted.

The missing value is NaN (not a number) that can be obtained with the argument of 6 in the routine `AMACH` (Reference Material). For this example, we set the first response variable in the first cell ($C_1 = 1, C_2 = 1$) to a missing value; we set the second response variable in the (2, 1) cell to a missing value; and we set the frequency variable in the (1, 2) cell to a missing value. The data are now as shown below, with “NaN” in place of the missing values.

		C₁			
		1		2	
		<i>R₁</i>	<i>R₂</i>	<i>R₁</i>	<i>R₂</i>
C₂	1	NaN	3.4	3.8	NaN
		NaN	3.4	5.2	6.3
		NaN	3.4	4.9	1.2
		7.0	2.6		
		<i>R₁</i>	<i>R₂</i>	<i>R₁</i>	<i>R₂</i>
2	NaN	NaN	6.5	3.4	
	3.2	7.1	3.1	5.1	
	1.7	6.3			

The first two rows output in `CELIF` are the values of the classification variables, and the third row is the frequencies of the cells, as before. The next three rows correspond to the first response variable, and the last three rows correspond to the second response variable. (This is “case 6” above, where the argument `CELIF` is described.)

```

USE CSTAT_INT
USE WRRRN_INT

INTEGER      KMAX, LDCELI, LDX, NR, NCOL
PARAMETER   (KMAX=4, LDCELI=15, LDX=10, NR=2, NCOL=5)
!
INTEGER      IDO, IFRQ, IRX(NR), IWT, K, MIN0, MOPT, NR
REAL         CELIF(LDCELI,KMAX), X(LDX,NCOL)
INTRINSIC   MIN0
!
                                Get data for example.
DATA X/1.0, 1.0, 1.0, 1.0, 1.0, 2.0, 2.0, 2.0, 2.0, 2.0, 1.0, &
      1.0, 2.0, 2.0, 2.0, 1.0, 1.0, 1.0, 2.0, 2.0, 5.0, 7.0, 4.3, &
      3.2, 1.7, 3.8, 5.2, 4.9, 6.5, 3.1, 3.4, 2.6, 9.8, 7.1, 6.3, &
      2.4, 6.3, 1.2, 3.4, 5.1, 3.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, &
      1.0, 1.0, 1.0/
!
                                All data are input at once.
IDO = 0
NROW = 10
K = 0
!
                                Frequencies are in the 5th column.
!
                                All weights are equal
IFRQ = 5
IWT = 0
!
                                Response variables are in 3rd and 4th
!
                                columns.
IRX(1) = 3
IRX(2) = 4
!
                                Set some values to "missing" for
!
                                this example. Specify elementwise
!
                                deletion of missing values of the

```

```

!                                     response variables.
MOPT   = 1
X(1,3) = AMACH(6)
X(6,4) = AMACH(6)
X(3,5) = AMACH(6)
!
CALL CSTAT (X, KMAX, CELIF, NR=NR, IRX=IRX, MOPT=MOPT, IFRQ=IFRQ, &
           K=K)
!                                     Print the results.
CALL WRRRN ('Statistics for the Cells', CELIF, NRA=(NCOL+2*NR), &
           NCA=MIN0(KMAX, K))
END

```

Output

```

Statistics for the Cells
      1      2      3      4
1  1.00  1.00  2.00  2.00
2  1.00  2.00  1.00  2.00
3  4.00  2.00  3.00  2.00
4  1.00  2.00  3.00  2.00
5  7.00  2.45  4.63  4.80
6  0.00  1.12  1.09  5.78
7  4.00  2.00  2.00  2.00
8  3.20  6.70  3.75  4.25
9  0.48  0.32 13.01  1.44

```

MEDPL

Computes a median polish of a two-way table.

Required Arguments

TABLE — $NROW$ by $NCOL$ matrix containing the table. (Input)

MAXIT — Maximum number of polishing iterations to be performed. (Input)
 An iteration is counted each time the rows or the columns are polished. The iterations begin by polishing the rows.

PTABLE — $(NROW + 1)$ by $(NCOL + 1)$ matrix containing the cell residuals from the fitted table and, in the last row and column, the marginal residuals. (Output)

Optional Arguments

NROW — Number of rows in the table. (Input)
 Default: $NROW = \text{size}(\text{TABLE}, 1)$.

NCOL — Number of columns in the table. (Input)
 Default: $NCOL = \text{size}(\text{TABLE}, 2)$.

LDTABL — Leading dimension of `TABLE` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDTABL = size (TABLE,1)`.

LDPTAB — Leading dimension of `PTABLE` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDPTAB = size (PTABLE,1)`.

ITER — Number of iterations actually performed. (Output)

FORTRAN 90 Interface

Generic: `CALL MEDPL (TABLE, MAXIT, PTABLE [,...])`

Specific: The specific interface names are `S_MEDPL` and `D_MEDPL`.

FORTRAN 77 Interface

Single: `CALL MEDPL (NROW, NCOL, TABLE, LDTABL, MAXIT, PTABLE, LDPTAB, ITER)`

Double: The double precision name is `DMEDPL`.

Example

This example is taken from Emerson and Hoaglin (1983, page 168). It involves data on infant mortality in the United States, classified by father's education and by region of the country. In order to show the difference between making only one polishing pass over the rows and polishing until convergence, on the first invocation `MAXIT` is set to one. On a second call, it is set large enough to have reasonable assurance of execution until convergence. In the first case, the last row and column of `PTABLE` are printed. The values in these are the medians before any polishing. These values approach zero as the polishing continues.

```
USE MEDPL_INT
USE UMACH_INT
USE WRRRL_INT

INTEGER NCOL, NROW
PARAMETER (NCOL=5, NROW=4)
!
INTEGER ITER, LDPTAB, LDTABL, MAXIT, NOUT
REAL PTABLE(NROW+1,NCOL+1), TABLE(NROW,NCOL)
CHARACTER CLABEL(1)*5, RLABEL(1)*5
!
DATA CLABEL/'NONE'/
DATA RLABEL/'NONE'/
DATA TABLE/25.3, 32.1, 38.8, 25.4, 25.3, 29.0, 31.0, 21.1, 18.2, &
18.8, 19.3, 20.3, 18.3, 24.3, 15.7, 24.0, 16.3, 19.0, 16.8, &
17.5/
!
CALL UMACH (2, NOUT)
```

```

MAXIT = 1
LDTABL = 4
LDPTAB = 5
CALL MEDPL (TABLE, MAXIT, PTABLE, ITER=ITER)
CALL WRRRL ('Fitted table after one iteration over the rows', &
           PTABLE, CLABEL, RLABEL, FMT='(W10.4)')
MAXIT = 15
CALL MEDPL (TABLE, MAXIT, PTABLE, ITER=ITER)
CALL WRRRL ('%/Fitted table and marginal residuals', PTABLE, &
           CLABEL, RLABEL, FMT='(W10.4)')
WRITE (NOUT,99999) ITER
99999 FORMAT (/, ' Iterations taken: ', I2)
END

```

Output

```

Fitted table after one iteration over the rows
  7.0      7.0      -0.1      0.0      -2.0      18.3
  7.8      4.7      -5.5      0.0      -5.3      24.3
 19.5     11.7      0.0     -3.6     -2.5      19.3
  4.3      0.0     -0.8      2.9     -3.6      21.1
  0.0      0.0      0.0      0.0      0.0      0.0

```

```

Fitted table and marginal residuals
-1.55      0.00      0.00     -1.15      0.60     -1.45
 1.55      0.00     -3.10      1.15     -0.40      2.25
10.85      4.60      0.00     -4.85      0.00     -0.35
-3.25     -6.00      0.30      2.75      0.00      0.35
 8.10      6.55     -0.55      0.70     -3.05     20.20

```

Iterations taken: 15

Comments

Workspace may be explicitly provided, if desired, by use of M2DPL/DM2DPL.
The reference is:

```
CALL M2DPL (NROW, NCOL, TABLE, LDTABL, MAXIT, PTABLE,
           LDPTAB, ITER, WK)
```

The additional argument is:

WK — Work vector of length $\max(\text{NROW}, \text{NCOL})$.

Description

The routine MEDPL performs a median polish on a two-way table. It first copies TABLE into PTABLE and fills the last row and last column of PTABLE with zeroes. It then computes the row-wise medians, adds these to the values in the last column and corresponding row, and subtracts them from the other entries in the corresponding row. Similar computations are then performed for all $\text{NCOL} + 1$ columns. The whole procedure is then repeated (using $\text{NROW} + 1$ rows) until convergence is achieved (until no changes occur), or until MAXIT iterations are performed. Convergence is known to have occurred if ITER is less than MAXIT.

As Emerson and Hoaglin (1983) discuss, it is not necessarily desirable to continue until convergence. If `MAXIT` is set to twice the maximum of the number of rows and columns plus five, it is likely that convergence will occur.

As Emerson and Hoaglin point out, median polish starting with rows can lead to a different fit from that obtained by starting with columns. Although `MEDPL` does not make provision for choosing which dimension to start with, `TABLE` can be transposed by use of routine `TRNRR` (IMSL MATH/LIBRARY). Use of the transposed table in `MEDPL` would result in the iterations beginning with the columns of the original table. Further descriptions of median polish, which was first proposed by John Tukey, and examples of its use can be found in Tukey (1977, Chapter 11) and in Velleman and Hoaglin (1981, Chapter 8).

Chapter 2: Regression

Routine

2.1.	Simple Linear Regression		
	Straight line fit	RLINE	86
	Simple linear regression analysis	RONE	89
	Response control by a fitted line.....	RINCF	99
	Inverse prediction by a fitted line	RINPF	103
2.2.	Multivariate General Linear Model Analysis		
2.2.1	Model Fitting		
	From raw data for a single dependent variable	RLSE	107
	From covariances	RCOV	113
	From raw data without classification variables	RGIVN	117
	From raw data with classification variables	RGLM	128
	With linear equality restrictions	RLEQU	142
2.2.2	Statistical Inference and Diagnostics		
	Summary statistics for a fitted regression.....	RSTAT	152
	Variance-covariance		
	matrix of the estimated coefficients	RCOVB	164
	Construction of a completely testable hypothesis	CESTI	170
	Sums of crossproducts for a multivariate hypothesis	RHPSS	176
	Tests for the multivariate linear hypothesis	RHPTE	183
	Test for lack of fit based on exact replicates	RLOFE	189
	Test for lack of fit based on near replicates.....	RLOFN	195
	Intervals and diagnostics for individual cases	RCASE	204
	Diagnostics for outliers and influential cases.....	ROTIN	215
2.2.3	Utilities for Classification Variables		
	Getting unique values of classification variables	GCLAS	222
	Generation of regressors for a general linear model.....	GRGLM	225
2.3.	Variable Selection		
	All best regressions via leaps-and-bounds algorithm	RBEST	231
	Stepwise regression	RSTEP	237
	Generalized sweep of a nonnegative definite matrix.....	GSWEP	247
	Retrieval of a symmetric submatrix		
	from a symmetric matrix	RSUBM	251

2.4.	Polynomial Regression and Second-Order Models		
2.4.1	Polynomial Regression Analysis		
	Polynomial fit of known degree	RCURV	255
	Polynomial regression analysis	RPOLY	260
2.4.2	Second-Order Model Design		
	Generation of an orthogonal central composite design	RCOMP	268
2.4.3	Utility Routines for Polynomial Models and Second-Order Models		
	Polynomial regression fit	RFORP	272
	Summary statistics for a fitted polynomial model	RSTAP	279
	Case statistics for a fitted polynomial model	RCASP	285
	Generation of orthogonal polynomials	OPOLY	292
	Centering of variables and generation of crossproducts.....	GCSCP	296
	Transforming coefficients for a second order model.....	TCSCP	301
2.5.	Nonlinear Regression Analysis		
	Nonlinear regression fit	RNLIN	304
2.6.	Fitting Linear Models Based on Criteria Other Than Least Squares		
	Least absolute value regression	RLAV	317
	Least L_p norm regression	RLLP	322
	Least maximum value regression	RLMV	333

Usage Notes

Simple Linear Regression

The simple linear regression model is

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad i = 1, 2, \dots, n$$

where the observed values of the y_i 's constitute the responses or values of the dependent variable, the x_i 's are the settings of the independent (explanatory) variable, β_0 and β_1 are the intercept and slope parameters, respectively, and the ε_i 's are independently distributed normal errors each with mean zero and variance σ^2 .

Routine `RLINE` (page 86) fits a straight line and computes summary statistics for the simple linear regression model. There are no options with this routine.

Routine `RONE` (page 89) analyzes a simple linear regression model. Routine `RONE` requires a data matrix as input. There is an option for excluding the intercept β_0 from the model. The variables x , y , weights (optional), and frequencies (optional) must correspond to columns in this matrix. The simple linear regression model is fit, summary statistics are computed (including a test for lack of fit), and confidence intervals and diagnostics for individual cases are computed. There are options for printing and plotting the results.

Routines `RINCF` (page 99) and `RINPF` (page 103) solve the inverse regression (calibration) problem using a fitted simple linear regression. Routines `RLINE` (page 86) or `RONE` can be used to compute the fit. Routine `RINCF` estimates settings of the independent variable that restrict, at a

specified confidence percentage, y to a given specified range. Routine `RINPF` computes a confidence interval on the setting of the independent variable for a given response y_0 .

Multiple Linear Regression

The multiple linear regression model is

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik} + \varepsilon_i \quad i = 1, 2, \dots, n$$

where the observed values of the y_i 's constitute the responses or values of the dependent variable, the x_{i1} 's, x_{i2} 's, ..., x_{ik} 's are the settings of the k independent (explanatory) variables, $\beta_0, \beta_1, \dots, \beta_k$ are the regression coefficients, and the ε_i 's are independently distributed normal errors each with mean zero and variance σ^2 .

Routine `RLSE` ([page 107](#)) fits the multiple linear regression model. There is an option for excluding the intercept β_0 . There are no other options. The responses are input in a one-dimensional array `Y`, and the independent variables are input in a two-dimensional array `X` that contains the individual cases as the rows and the variables as the columns.

By specifying a single dependent variable, either `RGIVN` ([page 117](#)) or `RCOV` ([page 113](#)) can also be used to fit the multiple linear regression. (These routines are designed to fit any number of dependent variables simultaneously. See the section "Multivariate General Linear Model" [on page 75](#).)

Routine `RGIVN` fits the model using fast Givens transformations. For large data sets that cannot be stored in a single array, `RGIVN` is designed to allow multiple invocations. In this case, only some of the rows from the entire data set are input at any one time. Alternatively, the data set can be input in a single array.

Routine `RCOV` fits the multiple linear regression model from the sum of squares and crossproducts matrix for the data $(x_1, x_2, \dots, x_k, y)$. Routine `CORVC` in Chapter 3, "Correlation," can compute the required sums of squares and crossproducts matrix for input into `RCOV`. Routine `RORDM` in Chapter 19, "Utilities," can reorder this matrix, if required.

Three routines in the IMSL MATH/LIBRARY can be used for fitting the multiple linear regression model. Routine `LSQRR` (IMSL MATH/LIBRARY) computes the fit via the Householder QR decomposition. Routine `LSBRR` (IMSL MATH/LIBRARY) computes the fit via iterative refinement. Routine `LSVRR` (IMSL MATH/LIBRARY) computes the singular value decomposition of a matrix. Routines `LSQRR` and `LSBRR` use the regressors and dependent variable as two input arrays. Routine `LSVRR` computes the singular value decomposition of the matrix of regressors, from which the regression coefficients can be obtained. Kennedy and Gentle (1980, section 8.1) discuss some of the computational advantages and disadvantages of the various methods for least-squares computations.

No Intercept Model

Several routines provide the option for excluding the intercept from a model. In most practical applications, the intercept should be included in the model. For routines that use the sums of squares and crossproducts matrix as input, the no-intercept case can be handled by using the raw sums of squares and crossproducts matrix as input in place of the corrected sums of squares and crossproducts. The raw sum of squares and crossproducts matrix can be computed as

$(x_1, x_2, \dots, x_k, y)^T(x_1, x_2, \dots, x_k, y)$ using the matrix multiplication routine `MXTXF` (IMSL MATH/LIBRARY).

Variable Selection

Variable selection can be performed by `RBEST` (page 231), which does all best subset regressions, or by `RSTEP` (page 237), which does stepwise regression. In either case, the sum of squares and crossproducts matrix must first be formed. The method used by `RBEST` is generally preferred over that used by `RSTEP` because `RBEST` implicitly examines all possible models in the search for a model that optimizes some criterion while stepwise does not examine all possible models. However, the computer time and memory requirements for `RBEST` can be much greater than that for `RSTEP` when the number of candidate variables is large.

Two utility routines `GSWEP` (page 247) and `RSUBM` (page 251) are provided also for variable selection. Routine `GSWEP` performs a generalized sweep of a nonnegative definite matrix. Routine `RSUBM` can be invoked after either `GSWEP` or `RSTEP` in order to extract the symmetric submatrix whose rows and columns have been swept, i.e., whose rows and columns have entered the stepwise model. Routines `GSWEP` and `RSUBM` can be invoked prior to `RBEST` in order to force certain variables into all the models considered by `RBEST`.

Polynomial Model

The polynomial model is

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \dots + \beta_k x_i^k + \varepsilon_i \quad i = 1, 2, \dots, n$$

where the observed values of the y_i 's constitute the responses or values of the dependent variable, the x_i 's are the settings of the independent (explanatory) variables, $\beta_0, \beta_1, \dots, \beta_k$ are the regression coefficients, and the ε_i 's are independently distributed normal errors each with mean zero and variance σ^2 .

Routine `RCURV` (page 255) fits a specified degree polynomial. Routine `RPOLY` (page 260) determines the degree polynomial to fit and analyzes this model. If only a decomposition of sum of squares for first, second, ..., k -th degree effects in a polynomial model is required, either `RCURV` or the service routine `RFORP` (page 272) can be used to compute this decomposition. The other service routines (`RSTAP`, page 279, `RCASE`, page 285, `OPOLY`, page 292) can be used to perform other parts of the full analysis.

Multivariate General Linear Model

Routines for the multivariate general linear model use the model

$$Y = XB + \varepsilon$$

where Y is the $n \times q$ matrix of responses, X is the $n \times p$ matrix of regressors, B is the $p \times q$ matrix of regression coefficients, and ε is the $n \times q$ matrix of errors whose q -dimensional rows are identically and independently distributed multivariate normal with mean vector 0 and variance-covariance matrix Σ .

Specification of X for the General Linear Model

Variables used in the general linear model are either continuous or classification variables. Typically, multiple regression models use continuous variables, whereas analysis of variance models use classification variables. Although the notation used to specify analysis of variance models and multiple regression models may look quite different, the models are essentially the same. The term *general linear model* emphasizes that a common notational scheme is used for specifying a model that may contain both continuous and classification variables.

A general linear model is specified by its effects (sources of variation). We refer to an effect as a single variable or a product of variables. (The term *effect* is often used in a narrower sense, referring only to a single regression coefficient.) In particular, an effect is composed of one of the following:

1. a single continuous variable
2. a single classification variable
3. several different classification variables
4. several continuous variables, some of which may be the same
5. continuous variables, some of which may be the same, and classification variables, which must be distinct

Effects of the first type are common in multiple regression models. Effects of the second type appear as main effects in analysis of variance models. Effects of the third type appear as interactions in analysis of variance models. Effects of the fourth type appear in polynomial models and response surface models as powers and crossproducts of some basic variables. Effects of the fifth type appear in one-way analysis of covariance models as regression coefficients that indicate lack of parallelism of a regression function across the groups.

The specification of a general linear model is through arguments `INTCEP`, `NCLVAR`, `INDCL`, `NEF`, `NVEF`, and `INDEF`, whose meanings are as follows:

INTCEP — Intercept option. (Input)

INTCEP	Action
0	An intercept is not in the model.
1	An intercept is in the model.

NCLVAR — Number of classification variables. (Input)

INDCL — Index vector of length $NCLVAR$ containing the column numbers of X that are the classification variables. (Input)

NEF — Number of effects (sources of variation) in the model excluding error. (Input)

NVEF — Vector of length NEF containing the number of variables associated with each effect in the model. (Input)

INDEF — Index vector of length $NVEF(1) + NVEF(2) + \dots + NVEF(NEF)$. (Input)

The first $NVEF(1)$ elements give the column numbers of X for each variable in the first effect. The next $NVEF(2)$ elements give the column numbers for each variable in the second effect. ... The last $NVEF(NEF)$ elements give the column numbers for each variable in the last effect.

Suppose the data matrix has as its first 4 columns two continuous variables in columns 1 and 2 and two classification variables in columns 3 and 4. The data might appear as follows:

Column 1	Column 2	Column 3	Column 4
11.23	1.23	1.0	5.0
12.12	2.34	1.0	4.0
12.34	1.23	1.0	4.0
4.34	2.21	1.0	5.0
5.67	4.31	2.0	4.0
4.12	5.34	2.0	1.0
4.89	9.31	2.0	1.0
9.12	3.71	2.0	1.0

Each distinct value of a classification variable determines a level. The classification variable in column 3 has two levels. The classification variable in column 4 has three levels. (Integer values are recommended, but not required, for values of the classification variables. If real numbers are used, the values of the classification variables corresponding to the same level must be identical.) Some examples of regression functions and their specifications are as follows:

	INTCEP	NCLVAR	INDCL	NEF	NVEF	INDEF
$\beta_0 + \beta_1 x_1$	1	0		1	1	1
$\beta_0 + \beta_1 x_1 + \beta_2 x_1^2$	1	0		2	1,2	1,1,1
$\mu + \alpha_I$	1	1	3	1	1	3
$\mu + \alpha_i + \beta_j + \gamma_{ij}$	1	2	3,4	3	1,1,2	3,4,3,4
μ_{ij}	0	2	3,4	1	2	3,4
$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$	1	0		3	1,1,2	1,2,1,2
$\mu + \alpha_i + \beta x_{1i} + \beta_i x_{1i}$	1	1	3	3	1,1,2	3,1,1,3

Routines for Fitting the Model

Routine `RGLM` (page 128) fits a multivariate general linear model. If the data set is too large to be stored in a single array, `RGLM` is designed so that multiple invocations can be made. In this case, one or more rows of the entire data set can be input at each invocation. Alternatively, the data set can be input all at once in a single array. Index vectors are used to specify the column numbers of the data matrix used as classification variables, effects, and dependent variables. This is useful if several models with different effects need to be fit from the same data matrix.

Routine `RLEQU` (page 142) can be called after `RGIVN` (page 117) or `RGLM` to impose linear equality restrictions $AB = Z$ on the regression parameters. `RLEQU` checks consistency of the restrictions. Routine `RLEQU` is useful for fitting spline functions where restrictions on the regression parameters arise from continuity and differentiability conditions on the regression function.

Routine `RLEQU` can be used to test the multivariate general linear hypothesis $AB = Z$ by fitting the restricted model after the full model is fit. The additional degrees of freedom for error (and the additional sum of squares and crossproducts for error) gained in the restricted model can be used for computing a test statistic. However, a more efficient approach for computing the sum of squares and crossproducts for a multivariate general linear hypothesis is provided by `RHPSS` (page 176). See the next section entitled “Multivariate General Linear Hypothesis” for a brief description of the problem and related routines.

Two utility routines `GCLAS` (page 222) and `GRGLM` (page 225) are provided to determine the values of the classification variables and then to use those values and the specified general linear model to generate the regressors in the model. These routines would not be required if you use `RGLM` to fit the model since `RGLM` does this automatically. However, if other routines in this chapter are used that require the actual regressors and not the classification variables, then these routines could be used.

Linear Dependence and the R Matrix

Linear dependence of the regressors frequently arises in regression models—sometimes by design and sometimes by accident. The routines in this chapter are designed to handle linear dependence of the regressors, i.e., the $n \times p$ matrix X (the matrix of regressors) in the general linear model can have rank less than p . Often, the models are referred to as nonfull rank models.

As discussed in Searle (1971, Chapter 5) some care must be taken to use correctly the results of the fitted nonfull rank regression model for estimation and hypothesis testing. In the nonfull rank case, not all linear combinations of the regression coefficients can be estimated. Those linear combinations that can be estimated are called “estimable functions.” If routines in this chapter are used to attempt to estimate linear combinations that cannot be estimated, error messages are issued. A good general discussion of estimable functions is given by Searle (1971, pages 180–188).

The check used by routines in this chapter for linear dependence is sequential. The j -th regressor is declared linearly dependent on the preceding regressors

$j - 1$ regressors if

$$\sqrt{1 - R_{j-1,2,\dots,j-1}^2}$$

is less than or equal to `TOL`. Here, $R_{j-1,2,\dots,j-1}$ is the multiple correlation coefficient of the j -th regressor with the first $j - 1$ regressors. Also, `TOL` is a tolerance that must be input by the user. When a routine declares the j -th regressor to be linearly dependent on the first $j - 1$ regressors, the j -th regression coefficient is set to zero. Essentially, this removes the j -th regressor from the model.

The reason a sequential check is used is that frequently practitioners include the variables that they prefer to remain in the model first. Also, the sequential check is based on many of the computations already performed as this does not degrade the overall efficiency of the routines. There is no perfect test for linear dependence when finite precision arithmetic is used. The input of the tolerance `TOL` allows the user some control over the check for linear dependence. If you know your model is full rank, you can input `TOL = 0.0`. However, generally `TOL` should be input as approximately 100 times the machine epsilon. The machine epsilon is `AMACH(4)` in single precision and `DMACH(4)` in double precision. (See routines `AMACH` and `DMACH` (Reference Material))

Routines in this chapter performing least squares are based on QR decomposition of X or on a Cholesky factorization $R^T R$ of $X^T X$. Maindonald (1984, chapters 1–5) discusses these methods extensively. The R matrix used by the regression routines is taken to be a $p \times p$ upper triangular matrix, i.e., all elements below the diagonal are zero. The signs of the diagonal elements of R are used as indicators of linearly dependent regressors and as indicators of parameter restrictions imposed by fitting a restricted model. The rows of R can be partitioned into three classes by the sign of the corresponding diagonal element:

1. A positive diagonal element means the row corresponds to data.
2. A negative diagonal element means the row corresponds to a linearly independent restriction imposed on the regression parameters by $AB = Z$ in a restricted model.
3. A zero diagonal element means a linear dependence of the regressors was declared. The regression coefficients in the corresponding row of \hat{B} are set to zero. This represents an arbitrary restriction which is imposed to obtain a solution for the regression coefficients. The elements of the corresponding row of R are also set to zero.

Multivariate General Linear Hypothesis

Routine `RHPSS` (page 176) computes the matrix of sums of squares and crossproducts for the general linear hypothesis $HB = G$ for the multivariate general linear model $Y = XB + \varepsilon$ with possible linear equality restrictions $AB = Z$. The R matrix and \hat{B} from the routines that fit the model are required for input to `RHPSS`.

The rows of H must be linear combinations of the rows of R , i.e., $HB = G$ must be completely testable. If the hypothesis is not completely testable, routine `CESTI` (page 170) can be used to construct an equivalent completely testable hypothesis.

Routine `RHPTE` (page 183) computes several test statistics and approximate p -values for the multivariate general linear hypothesis. The test statistics computed included are Wilks' lambda, Roy's maximum root, Hotelling's trace, and Pillai's trace. Seber (1984, pages 409–416) and Morrison (1976, pages 222–224) discuss the procedures and compare the test statistics. The error sum of squares and crossproducts matrix (`SCPE`) output from the fit of the model is required for input to `RHPTE`. In addition, the hypothesis sum of squares and crossproducts matrix (`SCPH`), which can be computed using `RHPSS`, is required for input to `RHPTE`.

Nonlinear Regression Model

The nonlinear regression model is

$$y_i = f(x_i; \theta) + \varepsilon_i \quad i = 1, 2, \dots, n$$

where the observed values of the y_i 's constitute the responses or values of the dependent variable, the x_i 's are the known vectors of values of the independent (explanatory) variables, f is a known function of an unknown regression parameter vector θ , and the ε_i 's are independently distributed normal errors each with mean zero and variance σ^2 .

Routine `RNLIN` (page 304) performs the least-squares fit to the data for this model. The routine `RCOVB` (page 164) can be used to compute the large sample variance-covariance matrix of the estimated nonlinear regression parameters from the output of `RNLIN`.

Weighted Least Squares

Routines throughout the chapter generally allow weights to be assigned to the observations. The argument `IWT` is used throughout to specify the weighting option. (`IWT = 0` means ordinary least squares; a positive `IWT` means weighted least squares with weights in column `IWT` of the data set.) All of the weights must be nonnegative. For routines requiring a sum of squares and crossproducts matrix for input, a weighted analysis can be performed by using as input a weighted sum of squares and crossproducts matrix. Routine `CORVC` in Chapter 3, "Correlation," can compute the required weighted sum of squares and crossproducts matrix.

Computations that relate to statistical inference, e.g., t tests, F tests, and confidence intervals, are based on the multiple regression model except that the variance of ε_i is assumed to equal σ^2 (or Σ in the multivariate case) times the reciprocal of the corresponding weight.

If a single row of the data matrix corresponds to n_i observations, the argument `IFRQ` can be used to specify the frequency option. `IFRQ = 0` means that for all rows, $n_i = 1$; a positive `IFRQ` means

the frequencies are entered into column `IFRQ` of the data matrix. Degrees of freedom for error are affected by frequencies, but are unaffected by weights.

Summary Statistics

Summary statistics for a single dependent variable are computed by several routines in the regression chapter. The routines `RONE` (page 89), `RLSE` (page 107), `RSTEP` (page 237), and `RPOLY` (page 260) output some summary statistics with the fit of the the model. For additional summary statistics, the routines `RSTAT` (page 152) and `RSTAP` (page 279) can be used.

Routine `RSTAT` can be used to compute and print statistics related to a regression for each of the q dependent variables fitted by `RGIVN` (page 117), `RGLM` (page 128), `RLEQU` (page 142), or `RCOV` (page 113). Routine `RSTAT` computes summary statistics that include the model analysis of variance table, sequential sums of squares and F -statistics, coefficient estimates, estimated standard errors, t -statistics, variance inflation factors, and estimated variance-covariance matrix of the estimated regression coefficients. If only the variance-covariance matrix of the estimated regression coefficients in needed, routine `RCOV` (page 164) can be used.

The summary statistics are computed under the model $y = X\beta + \varepsilon$, where y is the $n \times 1$ vector of responses, X is the $n \times p$ matrix of regressors with $\text{rank}(X) = r$, β is the $p \times 1$ vector of regression coefficients, and ε is the $n \times 1$ vector of errors whose elements are independently normally distributed with mean 0 and variance σ^2/w_i .

Given the results of a weighted least-squares fit of this model (with the w_i 's as the weights), most of the computed summary statistics are output in the following variables:

AOV — a one-dimensional array usually of length 15. In `RSTEP`, `AOV` is of length 13 because the last last two elements of the array cannot be computed from the input. The array contains statistics related to the analysis of variance. The sources of variation examined are the regression, error, and total. The first 10 elements of `AOV` and the notation frequently used for these is described in the following table:

Model Analysis of Variance Table

Source of Variation	Degrees of Freedom	Sum of Squares	Mean Square	F	p -value
Regression	DFR=AOV(1)	SSR=AOV(4)	MSR=AOV(7)	AOV(9)	AOV(10)
Error	DFE=AOV(2)	SSE=AOV(5)	$s^2 = \text{AOV}(8)$		
Total	DFT=AOV(3)	SST=AOV(6)			

In the case an intercept is indicated (`INTCEP = 1`), the total sum of squares is the sum of squares of the deviations of y_i from its (weighted) mean

$$\bar{y}$$

—the so-called *corrected total sum of squares*, it is denoted by

$$\text{SST} = \sum_{i=1}^n w_i (y_i - \bar{y})^2$$

In the case an intercept is not indicated (`INTCEP=0`), the total sum of squares is the sum of squares of y_i —the so-called *corrected total sum of squares*, it is denoted by

$$\text{SST} = \sum_{i=1}^n w_i y_i^2$$

The error sum of squares is given by

$$\text{SSE} = \sum_{i=1}^n w_i (y_i - \hat{y}_i)^2$$

The error degrees of freedom is defined by

$$\text{DFE} = n - r$$

The estimate of σ^2 is given by

$$s^2 = \text{SSE}/\text{DFE}$$

which is the error mean square.

The computed F statistic for the null hypothesis $H_0 : \beta_1 = \beta_2 = \dots = \beta_k = 0$ versus the alternative that at least one coefficient is nonzero is given by

$$F = \text{MSR}/s^2$$

The p -value associated with the test is the probability of an F larger than that computed under the assumption of the model and the null hypothesis. A small p -value (less than 0.05) is customarily used to indicate that there is sufficient evidence from the data to reject the null hypothesis.

The remaining 5 elements in `AOV` frequently are displayed together with the actual analysis of variance table. The quantities R -squared ($R^2 = \text{AOV}(11)$) and adjusted R -squared

$$R_a^2 = \text{AOV}(12)$$

are expressed as a percentage and are defined by

$$R^2 = 100(\text{SSR}/\text{SST}) = 100(1 - \text{SSE}/\text{SST})$$

$$R_a^2 = 100 \max \left\{ 0, 1 - \frac{s^2}{\text{SST}/\text{DFT}} \right\}$$

The square root of s^2 ($s = \text{AOV}(13)$) is frequently referred to as the estimated standard deviation of the model error.

The overall mean of the responses

$$\bar{y}$$

is output in (`AOV(14)`).

The coefficient of variation ($\text{CV} = \text{AOV}(15)$) is expressed as a percentage and is defined by

$$\text{CV} = 100s / \bar{y}$$

COEF — a two dimensional array containing the regression coefficient vector

$$\hat{\beta}$$

as one column and associated statistics (including the estimated standard error, t statistic and p -value) in the remaining columns.

SQSS — a two dimensional array containing sequential sums of squares as one column and associated statistics (including degrees of freedom, F statistic, and p -value) in the remaining columns.

COVB — the estimated variance-covariance matrix of the estimated regression coefficients.

Tests for Lack of Fit

Tests for lack of fit are computed for simple linear regression by `RONE` (page 89), for the polynomial regression by routines `RPOLY` (page 260) and `RSTAP` (page 279) and for multiple regression by routines `RLOFE` (page 189) and `RLOFN` (page 195).

In the case of polynomial regression, the two-dimensional output array `TLOF` contains the lack of fit F tests for each degree polynomial 1, 2, ..., k , that is fit to the data. These tests are useful for indicating the degree of the polynomial required to fit the data well.

In the case of simple and multiple regression, the one-dimensional output array `TESTLF` of length 10 contains the analysis of variance table for the test of lack of fit. Two routines `RLOFE` and `RLOFN` can be used to compute a test for lack of fit. Routine `RLOFE` requires exact replicates of the independent variables, i.e., there must be at least two cases in the data set that have the same settings of all the independent variables, while `RLOFN` does not require exact replicates. Customarily, one would require there to be several sets of duplicate settings of the independent variables in order to use `RLOFE`.

For `RLOFE`, the 10 elements of `TESTLF` and the notation frequently used for these is described in the following table:

Lack of Fit Analysis of Variance Table

Source of Variation	Degrees of Freedom	Sum of Squares	Mean Square	F	p -value
Lack of Fit	<code>TESTLF(1)</code>	<code>TESTLF(4)</code>	<code>TESTLF(7)</code>	<code>TESTLF(9)</code>	<code>TESTLF(10)</code>
Error	<code>DFPE = TESTLF(2)</code>	<code>SSPE = TESTLF(5)</code>	<code>TESTLF(8)</code>		
Pure Error	<code>DFE = TESTLF(3)</code>	<code>SSE = TESTLF(6)</code>			

For `RLOFN`, the 10 elements of `TESTLF` are similar to those in the previous table. However, since there may not be exact replicates in the data, the data are grouped into sets of near replicates. Then, instead of computing a pure error (or within) sum of squares using a one-way analysis of variance model, an expanded one-way analysis of covariance model using the clusters of near replicates as the groups is computed. The error from this expanded model replaces the pure error

in the preceding table in order to compute an exact F test for lack of fit conditional on the selected clusters.

Diagnostics for Individual Cases

Diagnostics for individual cases (observations) are computed by several routines in the regression chapter. Routines `RONE` (page 89), and `RPOLY` (page 260) output diagnostics for individual cases with the fit. If the fit of the model is done by other routines, `RCASE` (page 204) and `RCASP` (page 285) can be used to compute the diagnostics.

Routine `RCASE` computes confidence intervals and diagnostics for individual cases in the data matrix. The cases can be stored in a single data matrix or multiple invocations can be made in which one or more rows of the entire data set are input at any one time. Statistics computed by `RCASE` include predicted values, confidence intervals, and diagnostics for detecting outliers and cases that greatly influence the fitted regression.

If not all of the statistics computed by `RCASE` are needed, `ROTIN` (page 215) can be used to obtain some of the statistics.

The diagnostics are computed under the model $y = X\beta + \varepsilon$, where y is the $n \times 1$ vector of responses, X is the $n \times p$ matrix of regressors with $\text{rank}(X) = r$, β is the $p \times 1$ vector of regression coefficients, and ε is the $n \times 1$ vector of errors whose elements are independently normally distributed with mean 0 and variance σ^2/w_i .

Given the results of a weighted least-squares fit of this model (with the w_i 's as the weights), the following five diagnostics are computed: (1) leverage, (2) standardized residual, (3) jackknife residual, (4) Cook's distance, and (5) DFFITS. These diagnostics are stored in the FORTRAN matrix `CASE`. The definition of these terms is given in the discussion that follows:

Let x_i be a column vector containing the elements of the i -th row of X . A case could be unusual either because of x_i or because of the response y_i . The *leverage* h_i is a measure of unusualness of the x_i . The leverage is defined by

$$h_i = \left[x_i^T (X^T W X)^{-} x_i \right] w_i$$

where $W = \text{diag}(w_1, w_2, \dots, w_n)$ and $(X^T W X)^{-}$ denotes a generalized inverse of $X^T W X$. The average value of the h_i 's is r/n . Regression routines declare x_i unusual if $h_i > 2r/n$. A row label x is printed beside a case that is unusual because of x_i . Hoaglin and Welsch (1978) call a data point highly influential (i.e., a leverage point) when this occurs.

Let e_i denote the residual

$$y_i - \hat{y}_i$$

for the i -th case. The estimated variance of e_i is $(1 - h_i)s^2/w_i$ where s^2 is the residual mean square from the fitted regression. The i -th *standardized residual* (also called the internally studentized residual) is by definition

$$r_i = e_i \sqrt{\frac{w_i}{s^2(1-h_i)}}$$

and r_i follows an approximate standard normal distribution in large samples.

The i -th *jackknife residual* or *deleted residual* involves the difference between y_i and its predicted value based on the data set in which the i -th case is deleted. This difference equals $e_i/(1 - h_i)$. The jackknife residual is obtained by standardizing this difference. The residual mean square for the regression in which the i -th case is deleted is

$$s_i^2 = \frac{(n-r)s^2 - w_i e_i^2 / (1-h_i)}{n-r-1}$$

The jackknife residual is defined to be

$$t_i = e_i \sqrt{\frac{w_i}{s_i^2 (1-h_i)}}$$

and t_i follows a t distribution with $n - r - 1$ degrees of freedom. The regression routines declare y_i unusual (an outlier) if a jackknife residual greater than 2.0 in absolute value is computed. A row label γ is printed beside a case that is unusual because of y_i .

Cook's distance for the i -th case is a measure of how much an individual case affects the estimated regression coefficients. It is given as

$$D_i = \frac{w_i h_i e_i^2}{rs^2 (1-h_i)^2}$$

Weisberg (1985) states that if D_i exceeds the 50-th percentile of the $F(r, n - r)$ distribution, it should be considered large. (This value is about 1. This statistic does not have an F distribution.)

DFFITS, like Cook's distance, is also a measure of influence. For the i -th case, DFFITS is computed by the formula

$$\text{DFFITS}_i = e_i \sqrt{\frac{w_i h_i}{s_i^2 (1-h_i)^2}}$$

Hoaglin and Welsch (1978) suggest that DFFITS_i is greater than

$$2\sqrt{r/n}$$

is large.

Transformations

Transformations of the independent variables are sometimes useful in order to satisfy the regression model. The inclusion of squares and crossproducts of the variables

$$(x_1, x_2, x_1^2, x_2^2, x_1 x_2)$$

is often needed. Logarithms of the independent variables are also often used. (See Draper and Smith, 1981, pages 218–222, Box and Tidwell, 1962, Atkinson, 1985, pages 177–180, Cook and Weisberg, 1982, pages 78–86.)

When the responses are described by a nonlinear function of the parameters, a transformation of the model equation can often be selected so that the transformed model is linear in the regression parameters. For example, the exponential model

$$y = e^{\beta_0 + \beta_1 x_i} \varepsilon$$

by taking natural logarithms on both sides of the equation, can be transformed to a model that satisfies the linear regression model provided the ε_i 's have a log normal distribution (Draper and Smith, pages 222–225).

When the responses are nonnormal and their distribution is known, a transformation of the responses can often be selected so that the transformed responses closely satisfy the regression model assumptions. The square root transformation for counts with a Poisson distribution and the arc-sine transformation for binomial proportions are common examples (Snedecor and Cochran, 1967, pages 325–330, Draper and Smith, pages 237–239).

If the distribution of the responses is not known, the data can be used to select a transformation so that the transformed responses may more closely obey the regression model. For a positive response variable $y > 0$, the family of power transformations indexed by λ

$$y^{(\lambda)} = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \ln y & \text{if } \lambda = 0 \end{cases}$$

and generalizations of this family are useful. Routine `BCTR` (See Chapter 8, “Time Series Analysis and Forecasting”) can be used to perform the transformation. A method to estimate and to compute an approximate test for $\lambda = 1$ is given by Atkinson (1973). Also, Atkinson (1986) discusses transformation deletion statistics for computing the estimate and test leaving out a single observation since the evidence for a transformation of the response may sometimes depend crucially on one or a few observations.

Alternatives to Least Squares

The method of least squares has desirable characteristics when the errors are normally distributed, e.g., a least-squares solution produces maximum likelihood estimates of the regression parameters. However, when errors are not normally distributed, least squares may yield poor estimators. The least absolute value (LAV, L_1) criterion yields the maximum likelihood estimate when the errors follow a Laplace distribution. Routine `RLAV` (page 317) is often used when the errors have a heavy tailed distribution or when a fit is needed that is resistant to outliers.

A more general approach, minimizing the L_p norm ($p \geq 1$), is given by routine `RLLP` (page 322). Although the routine requires about 30 times the CPU time for the case $p = 1$ than would the use of `RLAV`, the generality of `RLLP` allows the user to try several choices for $p \geq 1$ by simply changing the input value of p in the calling program. The CPU time decreases as p gets larger. Generally, choices of p between 1 and 2 are of interest. However, the L_p norm solution for values of p larger than 2 can also be computed.

The minimax (LMV, L_∞ , Chebyshev) criterion is used by `RLMV` (page 333). Its estimates are very sensitive to outliers, however, the minimax estimators are quite efficient if the errors are uniformly distributed.

Missing Values

NaN (not a number) is the missing value code used by the regression routines. Use function `AMACH(6)` (or function `DMACH(6)` with double precision regression routines) to retrieve NaN. (See the section “Machine-Dependent Constants” in Reference Material.) Any element of the data matrix that is missing must be set to `AMACH(6)` (or `DMACH(6)` for double precision). In fitting regression models, any row of the data matrix containing NaN for the independent, dependent, weight, or frequency variables is omitted from the computation of the regression parameters.

Often predicted values and confidence intervals are desired for combinations of settings of the independent variables not used in computing the regression fit. This can be accomplished by including additional rows in the data matrix. These additional rows should contain the desired settings of the independent variables along with the responses set equal to NaN. The cases with NaN will not be used in determining the estimates of the regression parameters, and a predicted value and confidence interval will be computed from the given settings of the independent variables.

RLINE

Fits a line to a set of data points using least squares.

Required Arguments

XDATA — Vector of length `NOBS` containing the *x*-values. (Input)

YDATA — Vector of length `NOBS` containing the *y*-values. (Input)

B0 — Estimated intercept of the fitted line. (Output)

B1 — Estimated slope of the fitted line. (Output)

Optional Arguments

NOBS — Number of observations. (Input)

Default: `NOBS = size(XDATA,1)`.

STAT — Vector of length 12 containing the statistics described below. (Output)

I	STAT(I)
1	Mean of <i>XDATA</i>
2	Mean of <i>YDATA</i>
3	Sample variance of <i>XDATA</i>
4	Sample variance of <i>YDATA</i>

- I STAT(I)**
- 5 Correlation
- 6 Estimated standard error of B0
- 7 Estimated standard error of B1
- 8 Degrees of freedom for regression
- 9 Sum of squares for regression
- 10 Degrees of freedom for error
- 11 Sum of squares for error
- 12 Number of (x, y) points containing NaN (not a number) as either the x or y value

FORTRAN 90 Interface

Generic: CALL RLINE (XDATA, YDATA, B0, B1 [,...])

Specific: The specific interface names are S_RLINE and D_RLINE.

FORTRAN 77 Interface

Single: CALL RLINE (NOBS, XDATA, YDATA, B0, B1, STAT)

Double: The double precision name is DRLINE.

Example

This example fits a line to a set of data discussed by Draper and Smith (1981, Table 1.1, pages 9–33). The response y is the amount of steam used per month (in pounds), and the independent variable x is the average atmospheric temperature (in degrees Fahrenheit).

```

USE RLINE_INT
USE UMACH_INT
USE WRRRL_INT
INTEGER NOBS
PARAMETER (NOBS=25)
!
INTEGER NOUT
REAL B0, B1, STAT(12), XDATA(NOBS), YDATA(NOBS)
CHARACTER CLABEL(13)*15, RLABEL(1)*4
!
DATA XDATA/35.3, 29.7, 30.8, 58.8, 61.4, 71.3, 74.4, 76.7, 70.7, &
      57.5, 46.4, 28.9, 28.1, 39.1, 46.8, 48.5, 59.3, 70.0, 70.0, &
      74.5, 72.1, 58.1, 44.6, 33.4, 28.6/
DATA YDATA/10.98, 11.13, 12.51, 8.4, 9.27, 8.73, 6.36, 8.5, &

```

```

7.82, 9.14, 8.24, 12.19, 11.88, 9.57, 10.94, 9.58, 10.09, &
8.11, 6.83, 8.88, 7.68, 8.47, 8.86, 10.36, 11.08/
DATA RLABEL/'NONE'/, CLABEL/' ', 'Mean of X', 'Mean of Y', &
'Variance X', 'Variance Y', 'Corr.', 'Std. Err. B0', &
'Std. Err. B1', 'DF Reg.', 'SS Reg.', 'DF Error', &
'SS Error', 'Pts. with NaN'/
!
CALL RLINE (XDATA, YDATA, B0, B1, STAT=STAT)
!
CALL UMACH (2, NOUT)
WRITE (NOUT,99999) B0, B1
99999 FORMAT (' B0 = ', F7.2, ' B1 = ', F9.5)
CALL WRRRL ('%/STAT', STAT, RLABEL, CLABEL, 1, 12, 1, &
FMT='(12W10.4)')
!
END

```

Output

B0 = 13.62 B1 = -0.07983

		STAT			
Mean of X	Mean of Y	Variance X	Variance Y	Corr.	Std. Err. B0
52.6	9.424	298.1	2.659	-0.8452	0.5815
Std. Err. B1	DF Reg.	SS Reg.	DF Error	SS Error	Pts. with NaN
0.01052	1	45.59	23	18.22	0

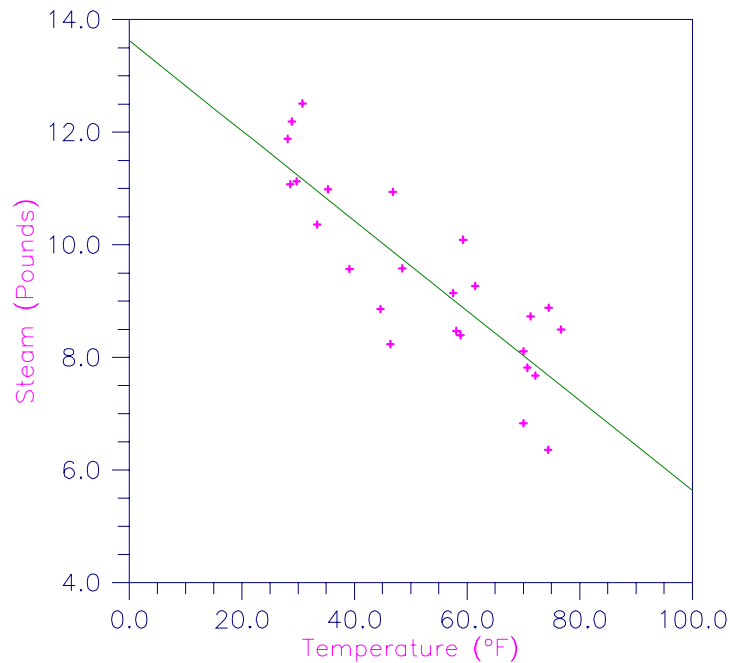


Figure 2-1 Plot of the Data and the Least Squares Line

Comments

Informational error

Type Code

4 1 Each (x, y) point contains NaN (not a number). There are no valid data.

Description

Routine `RLINE` fits a line to a set of (x, y) data points using the method of least squares. Draper and Smith (1981, pages 1–69) discuss the method. The fitted model is

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

where $\hat{\beta}_0$ (stored in `B0`) is the estimated intercept and $\hat{\beta}_1$ (stored in `B1`) is the estimated slope. In addition to the fit, `RLINE` produces some summary statistics, including the means, sample variances, correlation, and the error (residual) sum of squares. The estimated standard errors of $\hat{\beta}_0$ and $\hat{\beta}_1$ are computed under the simple linear regression model. The errors in the model are assumed to be uncorrelated and with constant variance.

If the x values are all equal, the model is degenerate. In this case, `RLINE` sets $\hat{\beta}_1$ to zero and $\hat{\beta}_0$ to the mean of the y values.

RONE

Analyzes a simple linear regression model.

Required Arguments

X — `NOBS` by `NCOL` matrix containing the data. (Input)

IRSP — Column number `IRSP` of *X* contains the data for the response (dependent) variable. (Input)

IND — Column number `IND` of *X* contains the data for the independent (explanatory) variable. (Input)

AOV — Vector of length 15 containing statistics relating to the analysis of variance. (Output)

I **AOV(I)**

1 Degrees of freedom for regression

2 Degrees of freedom for error

3 Total degrees of freedom

I	AOV (I)
4	Sum of squares for regression
5	Sum of squares for error
6	Total sum of squares
7	Regression mean square
8	Error mean square
9	<i>F</i> -statistic
10	<i>p</i> -value
11	R^2 (in percent)
12	Adjusted R^2 (in percent)
13	Estimated standard deviation of the model error
14	Mean of the response (dependent) variable
15	Coefficient of variation (in percent)

If `INTCEP = 1`, the regression and total are corrected for the mean. If `INTCEP = 0`, the regression and total are not corrected for the mean, and `AOV(14)` and `AOV(15)` are set to NaN (not a number).

COEF — `INTCEP + 1` by 5 matrix containing statistics relating the regression coefficients.
(Output)

If `INTCEP = 1`, the first row corresponds to the intercept. Row `INTCEP + 1` corresponds to the coefficient for the slope. The statistics in the columns are

Col. Description

1	Coefficient estimate
2	Estimated standard error of the coefficient estimate
3	<i>t</i> -statistic for the test that the coefficient is zero
4	<i>p</i> -value for the two-sided <i>t</i> test
5	Variance inflation factor

COVB — $\text{INTCEP} + 1$ by $\text{INTCEP} + 1$ matrix that is the estimated variance-covariance matrix of the estimated regression coefficients. (Output)

TESTLF — Vector of length 10 containing statistics relating to the test for lack of fit of the model. (Output)

Elem. Description

- 1 Degrees of freedom for lack of fit
- 2 Degrees of freedom for pure error
- 3 Degrees of freedom for error ($\text{TESTLF}(1) + \text{TESTLF}(2)$)
- 4 Sum of squares for lack of fit
- 5 Sum of squares for pure error
- 6 Sum of squares for error
- 7 Mean square for lack of fit
- 8 Mean square for pure error
- 9 F statistic
- 10 p -value

If there are no replicates in the data set, a test for lack of fit cannot be performed. In this case, elements 7, 8, 9, and 10 of **TESTLF** are set to NaN (not a number).

CASE — NOBS by 12 matrix containing case statistics. (Output)
Columns 1 through 12 contain the following:

Col. Description

- 1 Observed response
- 2 Predicted response
- 3 Residual
- 4 Leverage
- 5 Standardized residual
- 6 Jackknife residual

Col. Description

- 7 Cook's distance
- 8 DFFITS
- 9, 10 Confidence interval on the mean
- 11, 12 Prediction interval

Optional Arguments

NOBS — Number of observations. (Input)
Default: NOBS = size (x,1).

NCOL — Number of columns in x. (Input)
Default: NCOL = size (x,2).

LDX — Leading dimension of x exactly as specified in the dimension statement in the calling program. (Input)
Default: LDX = size (x,1).

INTCEP — Intercept option. (Input)
Default: INTCEP = 1.

INTCEP Action

- 0 An intercept is not in the model.
- 1 An intercept is in the model.

IFRQ — Frequency option. (Input)
IFRQ = 0 means that all frequencies are 1.0. For positive IFRQ, column number IFRQ of x contains the frequencies. If $x(I, IFRQ) = 0.0$, none of the remaining elements of row I of x are referenced, and updating of statistics is skipped for row I.
Default: IFRQ = 0.

IWT — Weighting option. (Input)
IWT = 0 means that all weights are 1.0. For positive IWT, column number IWT of x contains the weights.
Default: IWT = 0.

IPRED — Prediction interval option. (Input)
IPRED = 0 means that prediction intervals are computed for a single future response. For positive IPRED, a prediction interval is computed on the average of future responses, and column number IPRED of x contains the number of future responses in each average.
Default: IPRED = 0.

CONPCM — Confidence level for two-sided interval estimates on the mean, in percent.

(Input)

CONPCM percent confidence intervals are computed, hence, CONPCM must be greater than or equal to 0.0 and less than 100.0. CONPCM often will be 90.0, 95.0, or 99.0. For one-sided intervals with confidence level ONECL, where ONECL is greater than or equal to 50.0 and less than 100.0, set $CONPCM = 100.0 - 2.0 * (100.0 - ONECL)$.

Default: CONPCM = 95.0.

CONPCP — Confidence level for two-sided prediction intervals, in percent. (Input)

CONPCP percent prediction intervals are computed, hence, CONPCP must be greater than or equal to 0.0 and less than 100.0. CONPCP often will be 90.0, 95.0, or 99.0. For one-sided intervals with confidence level ONECL, where ONECL is greater than or equal to 50.0 and less than 100.0, set $CONPCP = 100.0 - 2.0 * (100.0 - ONECL)$.

Default: CONPCP = 95.0.

IPRINT — Printing option. (Input)

Default: IPRINT = 0.

IPRINT Action

- 0 No printing is performed.
- 1 AOV, COEF, TESTLF, and unusual rows of CASE are printed.
- 2 AOV, COEF, TESTLF, and unusual rows of CASE are printed. A plot of the data with the regression line is printed.
- 3 All printing is performed. A plot of the data with the regression line, a plot of the standardized residuals versus the independent variable, and a half-normal probability plot of the standardized residuals are printed.

LDCOEF — Leading dimension of COEF exactly as specified in the dimension statement in the calling program. (Input)

Default: LDCOEF = size (COEF,1).

LDCOVB — Leading dimension of COVB exactly as specified in the dimension statement in the calling program. (Input)

Default: LDCOVB = size (COVB,1).

LDCASE — Leading dimension of CASE exactly as specified in the dimension statement in the calling program. (Input)

Default: LDCASE = size (CASE,1).

NRMIS — Number of rows of data encountered containing missing values for the independent, dependent, weight, or frequency variables. (Output)

NaN (not a number) is used as the missing value code. Any row of x containing NaN as a value of the independent, dependent, weight, or frequency variables is omitted from the computations for fitting the model.

FORTRAN 90 Interface

Generic: CALL RONE (X, IRSP, IND, AOV, COEF, COVB, TESTLF, CASE [,...])

Specific: The specific interface names are S_RONE and D_RONE.

FORTRAN 77 Interface

Single: CALL RONE (NOBS, NCOL, X, LDX, INTCEP, IRSP, IND, IFRQ,
 IWT, IPRED, CONPCM, CONPCP, IPRINT, AOV, COEF, LDCOEF,
 COVB, LDCOVB, TESTLF, CASE, LDCASE, NRMISS)

Double: The double precision name is DRONE.

Example 1

This example fits a line to a set of data discussed by Draper and Smith (1981, pages 9–33). The response y is the amount of steam used per month (in pounds), and the independent variable x is the average atmospheric temperature (in degrees Fahrenheit). The `IPRINT = 1` option is selected. Hence, plots are not produced and only unusual cases are printed. Note in the case analysis, with the default page width, the observation number and the associated 12 statistics require two lines of output. (Routine `PGOPT`, Chapter 19, Utilities, can be invoked to increase the page width to put all 12 statistics on the same line.) Also note that observation 11 is labeled with a “Y” to indicate an unusual y (response). The residual for this case is about 2 standard deviations from zero.

```
USE RONE_INT
INTEGER  INTCEP, LDCASE, LDCOEF, LDCOVB, LDX, NCOEF, NCOL, NOBS
PARAMETER (NOBS=25, LDX=25, LDCASE=25, INTCEP=1, NCOEF=INTCEP+1, &
           LDCOEF=NCOEF, LDCOVB=NCOEF, NCOL=2)
!
INTEGER  IND, IPRINT, IRSP, NRMISS
REAL    AOV(15), CASE(LDCASE,12), COEF(LDCOEF,5), CONPCP, &
        COVB(LDCOVB,NCOEF), TESTLF(10), X(LDX,NCOL)
!
DATA (X(1,J),J=1,2) /35.3, 10.98/
DATA (X(2,J),J=1,2) /29.7, 11.13/
DATA (X(3,J),J=1,2) /30.8, 12.51/
DATA (X(4,J),J=1,2) /58.8,  8.40/
DATA (X(5,J),J=1,2) /61.4,  9.27/
DATA (X(6,J),J=1,2) /71.3,  8.73/
DATA (X(7,J),J=1,2) /74.4,  6.36/
DATA (X(8,J),J=1,2) /76.7,  8.50/
DATA (X(9,J),J=1,2) /70.7,  7.82/
DATA (X(10,J),J=1,2) /57.5,  9.14/
DATA (X(11,J),J=1,2) /46.4,  8.24/
DATA (X(12,J),J=1,2) /28.9, 12.19/
DATA (X(13,J),J=1,2) /28.1, 11.88/
DATA (X(14,J),J=1,2) /39.1,  9.57/
DATA (X(15,J),J=1,2) /46.8, 10.94/
DATA (X(16,J),J=1,2) /48.5,  9.58/
DATA (X(17,J),J=1,2) /59.3, 10.09/
```

```

DATA (X(18,J),J=1,2) /70.0, 8.11/
DATA (X(19,J),J=1,2) /70.0, 6.83/
DATA (X(20,J),J=1,2) /74.5, 8.88/
DATA (X(21,J),J=1,2) /72.1, 7.68/
DATA (X(22,J),J=1,2) /58.1, 8.47/
DATA (X(23,J),J=1,2) /44.6, 8.86/
DATA (X(24,J),J=1,2) /33.4, 10.36/
DATA (X(25,J),J=1,2) /28.6, 11.08/
!
IRSP = 2
IND = 1
CONPCP = 99.0
IPRINT = 1
CALL RONE (X, IRSP, IND, AOV, COEF, COVB, TESTLF, CASE, &
CONPCP=CONPCP, IPRINT=IPRINT, NRMISS=NRMISS)
!
END

```

Output

R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
71.444	70.202	0.8901	9.424	9.445

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Regression	1	45.59	45.59	57.543	0.0000
Residual	23	18.22	0.79		
Corrected Total	24	63.82			

* * * Inference on Coefficients * * *

Coef.	Estimate	Standard Error	t-statistic	Prob. of Larger t	Variance Inflation
1	13.62	0.5815	23.43	0.0000	10.67
2	-0.08	0.0105	-7.59	0.0000	1.00

* * * Test for Lack of Fit * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Lack of fit	22	17.40	0.7911	0.966	0.6801
Pure error	1	0.82	0.8192		
Residual	23	18.22			

* * * Case Analysis * * *

Obs.	Observed Cook's D	Predicted DFFITs	Residual 95.0% CI	Leverage 95.0% CI	Std. Res. 99.0% PI	Jack Res. 99.0% PI
Y 11	8.2400 0.0886	9.9189 -0.4497	-1.6789 9.5267	0.0454 10.3112	-1.9305 7.3640	-2.0625 12.4739

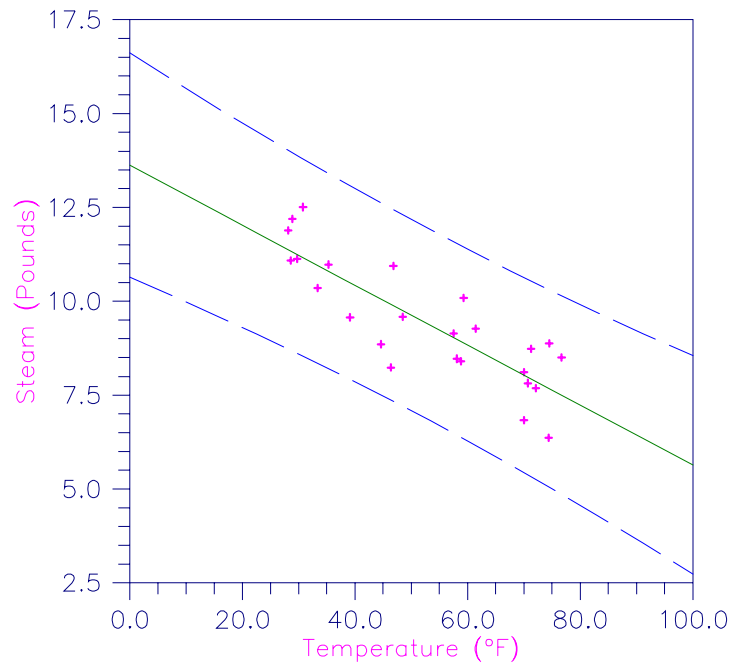


Figure 2-2 Plot of Line and 99% One-at-a-Time Prediction Intervals

Comments

1. Workspace may be explicitly provided, if desired, by use of `R2NE/DR2NE`. The reference is:

```
CALL R2NE (NOBS, NCOL, X, LDX, INTCEP, IRSP, IND, IFRQ, IWT,
IPRED, CONPCM, CONPCP, IPRINT, AOV, COEF, LDCOEF, COVB, LDCOVB,
TESTLF, CASE, LDCASE, NRMISS, IWK, WK)
```

The additional arguments are as follows:

IWK — Work vector of length `NOBS`.

WK — Work vector of length `3 * NOBS`.

2.	Informational errors	
	Type	Code
	3	5 CONPCM is less than 50.0. Confidence percentages commonly used are 90.0, 95.0, and 99.0.
	3	6 CONPCP is less than 50.0. Confidence percentages commonly used are 90.0, 95.0, and 99.0.
	4	1 Negative weight encountered.
	4	2 Negative frequency encountered.
	4	7 Each row of X contains NaN.

Description

Routine RONE performs an analysis for the simple linear regression model. In addition to the fit, summary statistics (analysis of variance, t tests, lack-of-fit test), and confidence intervals and diagnostics for individual cases are computed. With the printing option, diagnostic plots can also be produced. Draper and Smith (1981, chapter 1) give formulas for many of the statistics computed by RONE. For definitions of the case diagnostics (stored in CASE), see the “Usage Notes” of this chapter ([page 72](#)).

Additional Example

Example 2

This example fits a line to a data set discussed by Draper and Smith (1981, pages 38–40). The data set contains several repeated x values in order to assess lack of fit of the straight line. The IPRINT = 1 option is selected. Hence, plots are not produced and only unusual cases are printed. Note in the case analysis that observations 1 and 2 are labeled with an “x” to indicate an unusual x value. Each have leverage 0.1944 that exceeds the average leverage of $p/n = 2/24$ by a factor of 2.

```

USE RONE_INT
INTEGER   LDCASE, LDCEOF, LDCOV, LDX, NCOEF, NCOL, NOBS
PARAMETER (INTCEP=1, NCOL=2, NOBS=24, LDCASE=NOBS, LDX=NOBS, &
           NCOEF=INTCEP+1, LDCEOF=NCOEF, LDCOV=NCOEF)
!
INTEGER   IFRQ, IND, IPRED, IPRINT, IRSP
REAL      AOV(15), CASE(LDCASE,12), COEF(LDCEOF,5), &
           COV(LDCOV,NCOEF), TESTLF(10), X(LDX,NCOL)
!
DATA (X(1,J),J=1,2) /2.3, 1.3/
DATA (X(2,J),J=1,2) /1.8, 1.3/
DATA (X(3,J),J=1,2) /2.8, 2.0/
DATA (X(4,J),J=1,2) /1.5, 2.0/
DATA (X(5,J),J=1,2) /2.2, 2.7/
DATA (X(6,J),J=1,2) /3.8, 3.3/
DATA (X(7,J),J=1,2) /1.8, 3.3/
DATA (X(8,J),J=1,2) /3.7, 3.7/
DATA (X(9,J),J=1,2) /1.7, 3.7/
DATA (X(10,J),J=1,2) /2.8, 4.0/
DATA (X(11,J),J=1,2) /2.8, 4.0/
DATA (X(12,J),J=1,2) /2.2, 4.0/

```

```

DATA (X(13,J),J=1,2) /5.4, 4.7/
DATA (X(14,J),J=1,2) /3.2, 4.7/
DATA (X(15,J),J=1,2) /1.9, 4.7/
DATA (X(16,J),J=1,2) /1.8, 5.0/
DATA (X(17,J),J=1,2) /3.5, 5.3/
DATA (X(18,J),J=1,2) /2.8, 5.3/
DATA (X(19,J),J=1,2) /2.1, 5.3/
DATA (X(20,J),J=1,2) /3.4, 5.7/
DATA (X(21,J),J=1,2) /3.2, 6.0/
DATA (X(22,J),J=1,2) /3.0, 6.0/
DATA (X(23,J),J=1,2) /3.0, 6.3/
DATA (X(24,J),J=1,2) /5.9, 6.7/

```

!

```

IRSP = 1
IND = 2
IPRINT = 1
CALL RONE (X, IRSP, IND, AOV, COEF, COVB, TESTLF, CASE, &
          IPRINT=IPRINT, NRMISS=NRMISS)
END

```

Output

R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
22.983	19.483	0.9815	2.858	34.34

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Regression	1	6.32	6.325	6.565	0.0178
Residual	22	21.19	0.963		
Corrected Total	23	27.52			

* * * Inference on Coefficients * * *

Coef.	Estimate	Standard Error	t-statistic	Prob. of Larger t	Variance Inflation
1	1.436	0.5900	2.435	0.0235	8.672
2	0.338	0.1319	2.562	0.0178	1.000

* * * Test for Lack of Fit * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Lack of fit	11	8.72	0.793	0.700	0.7183
Pure error	11	12.47	1.134		
Residual	22	21.19			

* * * Case Analysis * * *

Obs.	Observed Cook's D	Predicted DFFITS	Residual 95.0% CI	Leverage 95.0% CI	Std. Res. 95.0% PI	Jack Res. 95.0% PI	
X	1	2.3000 0.0280	1.8756 0.2324	0.4244 0.9783	0.1944 2.7730	0.4817 -0.3489	0.4731 4.1002
X	2	1.8000 0.0009	1.8756 -0.0412	-0.0756 0.9783	0.1944 2.7730	-0.0859 -0.3489	-0.0839 4.1002
Y	13	5.4000 0.1481	3.0245 0.6264	2.3755 2.5877	0.0460 3.4612	2.4780 0.9426	2.8515 5.1063
Y	24	5.9000	3.7002	2.1998	0.1537	2.4363	2.7855

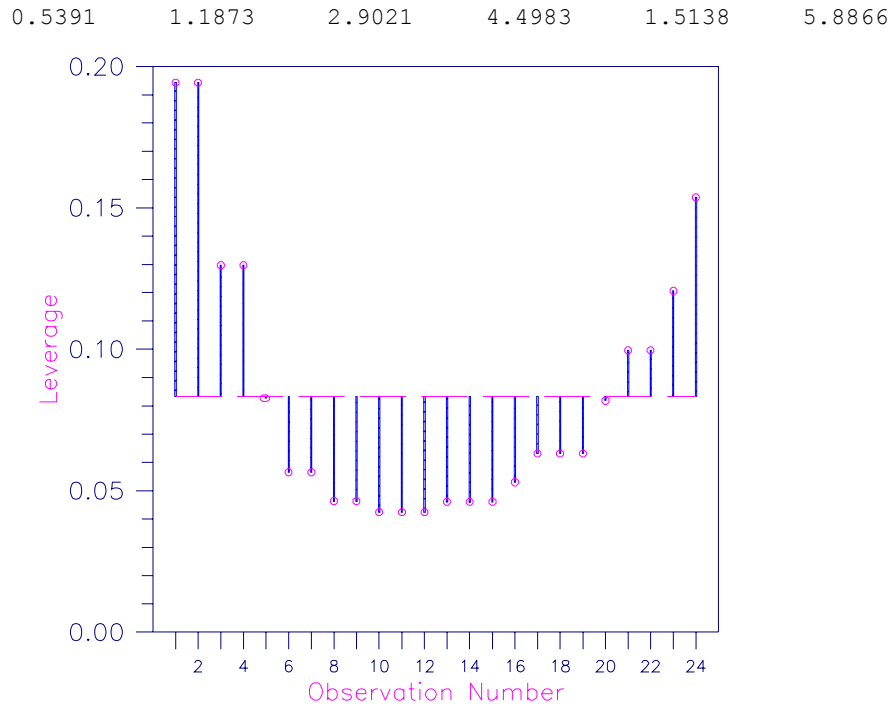


Figure 2-3 Plot of Leverages h_i and the Average ($p/n = 2/24$)

RINCF

Performs response control given a fitted simple linear regression model.

Required Arguments

SUMWTF — Sum of products of weights with frequencies from the fitted regression.
(Input, if **INTCEP** = 1)

In the ordinary case when weights and frequencies are all one, **SUMWTF** equals the number of observations.

DFE — Degrees of freedom for error from the fitted regression. (Input)

B — Vector of length **INTCEP** + 1 containing a least-squares solution for the intercept and slope. (Input)

INTCEP	Intercept	Slope
0		B(1)
1	B(1)	B(2)

XYMEAN — Vector of length 2 containing the variable means. (Input)
XYMEAN(1) is the independent variable mean. XYMEAN(2) is the dependent variable mean. If INTCEP = 0, XYMEAN is not referenced and can be a vector of length one.

SSX — Sum of squares for the independent variable. (Input)
If INTCEP = 1, SSX is the sums of squares of deviations of the independent variable from its mean. Otherwise, SSX is not corrected for the mean.

S2 — s^2 , the estimate of σ^2 from the fitted regression. (Input)

YLOWER — Lower limit for the response. (Input)

YUPPER — Upper limit for the response. (Input)

XLOWER — Lower limit on the independent variable for controlling the response. (Output)

XUPPER — Upper limit on the independent variable for controlling the response. (Output)

Optional Arguments

INTCEP — Intercept option. (Input)
Default: INTCEP = 1.

INTCEP Action

0 An intercept is not in the model.

1 An intercept is in the model.

SWTFY0 — $S2/SWTFY0$ is the estimated variance of the future response (or future response mean) that is to be controlled. (Input)
In the ordinary case, when weights and frequencies are all one, SWTFY0 is the number of observations in the response mean that is to be controlled. SWTFY0 = 0.0 means the true response mean is to be controlled.
Default: SWTFY0 = 0.0.

CONPER — Confidence level for a two-sided response control, in percent. (Input)
CONPER percent limits are computed; hence, CONPER must be greater than or equal to 0.0 and less than 100.0. CONPER often will be 90.0, 95.0, or 99.0. For one-sided control with confidence level ONECL, where ONECL is greater than or equal to 50.0 and less than 100.0, set CONPCM = $100.0 - 2.0 * (100.0 - ONECL)$.
Default: CONPER = 95.0.

FORTRAN 90 Interface

Generic: CALL RINCF (SUMWTF, DFE, B, XYMEAN, SSX, S2, YLOWER,
YUPPER, XLOWER, XUPPER [,...])

Specific: The specific interface names are S_RINCF and D_RINCF.

FORTRAN 77 Interface

Single: CALL RINCF (SUMWTF, DFE, INTCEP, B, XYMEAN, SSX, S2, SWTFY0, CONPER, YLOWER, YUPPER, XLOWER, XUPPER)

Double: The double precision name is DRINCF.

Example

This example estimates the settings of the independent variable that restrict, at 97.5% confidence, the true mean response to an upper bound of -4.623, using a fitted simple linear regression model. The fitted model excludes the intercept term. To accomplish one-sided control, CONPER is set to $100 - 2(100 - 97.5) = 95$, and YLOWER is set to an arbitrary value less than YUPPER. The output for XLOWER furnishes the lower bound for x necessary to control y .

```
USE RINCF_INT
USE UMACH_INT

INTEGER    INTCEP
PARAMETER (INTCEP=0)

!
INTEGER    NOUT
REAL       B(INTCEP+1), CONPER, DFE, ONECL, S2, SSX, SUMWTF, &
           SWTFY0, XLOWER, XUPPER, XYMEAN(1), YLOWER, YUPPER
!
DATA B/-.079829/
!
SUMWTF = 25.0
DFE    = 24.0
SSX    = 76323.0
S2     = 0.7926
SWTFY0 = 0.0
ONECL  = 97.5
CONPER = 100.0 - 2*(100.0-ONECL)
YUPPER = -4.623
YLOWER = -9.0
CALL RINCF (SUMWTF, DFE, B, XYMEAN, SSX, S2, YLOWER, YUPPER, &
           XLOWER, XUPPER, INTCEP=INTCEP, CONPER=CONPER)
CALL UMACH (2, NOUT)
WRITE (NOUT,*) 'XLOWER = ', XLOWER, '    XUPPER = ', XUPPER
END
```

Output

```
XLOWER = 63.1747    XUPPER = 104.07
```

Comments

Informational errors

Type Code

- | | | |
|---|---|--|
| 4 | 1 | The slope is not significant at the $(100 - \text{CONPER})$ percent level. Control limits cannot be obtained. |
| 4 | 2 | The computed lower limit, XLOWER, exceeds the computed upper limit, XUPPER. No satisfactory settings of the independent variable exist to control the response as specified. |

Description

Routine RINCF estimates settings of the independent variable that restrict, at a specified confidence percentage, the average of k randomly drawn responses to a given acceptable range (or the true mean response to a given acceptable range), using a fitted simple linear regression model. The results of routine RLINE (page 86) or RONE (page 89) can be used for input into RINCF. The simple linear regression model is assumed:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad i = 1, 2, \dots, n + k$$

where the ε_i 's are independently distributed normal errors with mean zero and variance σ^2/w_i . Here, n is the total number of observations used in the fit of the line, i.e., $n = \text{DFE} + \text{INTCEP} + 1$. Also, k is the number of additional responses whose average is to be restricted to the specified range. The w_i 's are the weights.

The methodology is based on Graybill (1976, pages 280–283). The estimate of σ^2 , s^2 (stored in S2), is the usual estimate of σ^2 from the fitted regression based on the first n observations. First, a test of the hypothesis $H_0: \beta_1 = 0$ vs. $H_a: \beta_1 \neq 0$ at level $\alpha = 1 - \text{CONPER}/100$ is performed. If H_0 is accepted, the model becomes $y_i = \beta_0 + \varepsilon_i$, and limits for x to control the response are meaningless since x is no longer in the model. In this case, a type 4 fatal error is issued. If H_0 is rejected and $\hat{\beta}_1$ is positive, a lower limit (upper limit) for x stored in XLOWER(XUPPER) is computed for the case where SWTFY0 is positive by

$$\bar{x} + \frac{\hat{\beta}_1 (y_0 - \bar{y})}{a} \pm \frac{ts}{a} \sqrt{\frac{a}{\sum_{i=1}^n w_i} + \frac{a}{\sum_{i=n+1}^{n+k} w_i} + \frac{(y_0 - \bar{y})^2}{\sum_{i=1}^n w_i (x_i - \bar{x})^2}}$$

where y_0 is the value stored in YLOWER(YUPPER) and where

$$a = \hat{\beta}_1^2 - \frac{t^2 s^2}{\sum_{i=1}^n w_i (x_i - \bar{x})^2}$$

and t is the $50 + \text{CONPER}/2$ percentile of the t distribution with DFE degrees of freedom. In the formula, the symbol \pm is used to indicate that $+$ is used to compute XLOWER with $y_0 = \text{YLOWER}$, and $-$ is used to compute XUPPER with $y_0 = \text{YUPPER}$. If H_0 is rejected and $\hat{\beta}_1$ is negative, a lower limit (upper limit) for x stored in XLOWER(XUPPER) is computed for the case where SWTFY0 is positive by a small modification. In particular, the symbol \pm is then taken so that $+$ is used to compute

X_{LOWER} with $y_0 = Y_{UPPER}$, and $-$ is used to compute X_{UPPER} with $y_0 = Y_{LOWER}$. These limits actually have a confidence coefficient less than that specified by $CONPER$.

In the weighted case, which was discussed earlier, the means (stored in $XYMEAN$) and the sum of squares for x (stored in SSX) are all weighted. When the variances of the ϵ_i 's are all equal, ordinary least squares must be used, this corresponds to all $w_i = 1$.

The previous discussion can be generalized to the case where an intercept is not in the model. The necessary modifications are to let $\beta_0 = 0, \hat{\beta}_0 = 0$ and to replace the first term under the square root symbol by zero, \bar{x} by zero, and \bar{y} by zero.

In order to restrict the true mean response to a specified range, i.e, when $SWTFY0$ is zero, the formulas are modified by replacing the second term under the square root symbol with zero.

RINPF

Performs inverse prediction given a fitted simple linear regression model.

Required Arguments

SUMWTF — Sum of products of weights with frequencies from the fitted regression.

(Input, if $INTCEP = 1$)

In the ordinary case when weights and frequencies are all one, $SUMWTF$ equals the number of observations used in the fit of the model.

DFS2 — Degrees of freedom for estimate of σ^2 . (Input)

If $IY0 = 1$, $DFS2$ is the degrees of freedom for error from the fitted regression. If $IY0 = 0$, $DFS2$ is the pooled degrees of freedom from the estimate of sigma-squared based on the fitted regression and the additional responses used to compute the mean $Y0$.

B — Vector of length $INTCEP + 1$ containing a least-squares solution for the intercept and slope. (Input)

INTCEP	Intercept	Slope
0		B(1)
1	B(1)	B(2)

XYMEAN — Vector of length 2 with the mean of the independent and dependent variables, respectively. (Input, if $INTCEP = 1$)

If $INTCEP = 0$, $XYMEAN$ is not referenced and can be a vector of length 1.

SSX — Sum of squares for x . (Input)

If $INTCEP = 1$, SSX is the sum of squares of deviations of x from its mean. If $INTCEP = 0$, SSX must not be corrected for the mean.

S2 — s^2 , the estimate of the variance of the error in the model. (Input)

If $IY0 = 1$, $S2$ is the estimate of σ^2 from the fitted regression. If $IY0 = 0$, $S2$ is the

pooled estimate of σ^2 based on the fitted regression, and the additional responses used to compute the mean Y_0 .

Y0 — Value of the response variable for which an interval estimate of the corresponding independent variable value is desired. (Input)

X0HAT — Point estimate of the independent variable. (Output)

XLOWER — Lower limit of the interval estimate for the independent variable. (Output)

XUPPER — Upper limit of the interval estimate for the independent variable. (Output)

Optional Arguments

INTCEP — Intercept option. (Input)

Default: `INTCEP = 1`.

INTCEP Action

0 An intercept is not in the model.

1 An intercept is in the model.

CONPER — Confidence level for the interval estimation. (Input)

`CONPER` must be expressed as a percentage between 0.0 and 100.0. `CONPER` often will be 90.0, 95.0, 99.0. For one-sided confidence intervals with confidence level `ONECL`, set `CONPER = 100.0 - 2.0 * (100.0 - ONECL)`.

Default: `CONPER = 95.0`.

IY0 — Option for Y_0 . (Input)

Default: `IY0 = 1`.

IY0 Meaning

0 Y_0 is a sample mean of one or more responses.

1 Y_0 is the true mean response.

SWTFY0 — Sum of products of weights with frequencies for Y_0 . (Input, if `IY0 = 0`)

In the ordinary case, when weights and frequencies are all one, `SWTFY0` is the number of observations used to obtain the mean Y_0 . If `IY0 = 1`, `SWTFY0` is not referenced.

FORTRAN 90 Interface

Generic: `CALL RINPF (SUMWTF , DFS2 , B , XYMEAN , SSX , S2 , Y0 , X0HAT , XLOWER , XUPPER [, ...])`

Specific: The specific interface names are S_RINPF and D_RINPF.

FORTRAN 77 Interface

Single: CALL RINPF (SUMWTF, DFS2, INTCEP, B, XYMEAN, SSX, S2, CONPER, IY0, SWTFY0, Y0, XOHAT, XLOWER, XUPPER)

Double: The double precision name is DRINPF.

Example

This example fits a line to a set of data discussed by Draper and Smith (1981, Table 1.1, page 9). The response y is the amount of steam used per month (in pounds), and the independent variable x is the average atmospheric temperature (in degrees Fahrenheit). A 95% confidence interval for the temperature x_0 is computed given a single response of $y_0 = 10$.

```
USE RINPF_INT
USE RLINE_INT
USE UMACH_INT

INTEGER      NOBS
PARAMETER   (NOBS=25)

!
INTEGER      INTCEP, IY0, NOUT
REAL         B(2), B0, B1, CONPER, DFS2, S2, SSX, STAT(12), &
             SUMWTF, SWTFY0, XOHAT, XDATA(NOBS), XLOWER, XUPPER, &
             XYMEAN(2), Y0, YDATA(NOBS)
!
DATA XDATA/35.3, 29.7, 30.8, 58.8, 61.4, 71.3, 74.4, 76.7, 70.7, &
      57.5, 46.4, 28.9, 28.1, 39.1, 46.8, 48.5, 59.3, 70.0, 70.0, &
      74.5, 72.1, 58.1, 44.6, 33.4, 28.6/
DATA YDATA/10.98, 11.13, 12.51, 8.4, 9.27, 8.73, 6.36, 8.5, &
      7.82, 9.14, 8.24, 12.19, 11.88, 9.57, 10.94, 9.58, 10.09, &
      8.11, 6.83, 8.88, 7.68, 8.47, 8.86, 10.36, 11.08/
!
CALL RLINE (XDATA, YDATA, B0, B1, STAT=STAT)
SUMWTF     = NOBS
DFS2       = STAT(10)
INTCEP     = 1
B(1)       = B0
B(2)       = B1
XYMEAN(1) = STAT(1)
XYMEAN(2) = STAT(2)
SSX        = STAT(3) * (NOBS-1)
S2         = STAT(11) / STAT(10)
CONPER     = 95.0
IY0        = 0
SWTFY0     = 1.0
Y0         = 10.0
CALL RINPF (SUMWTF, DFS2, B, XYMEAN, SSX, S2, Y0, XOHAT, XLOWER, &
           XUPPER, IY0=IY0, SWTFY0=SWTFY0)
CALL UMACH (2, NOUT)
```

```

WRITE (NOUT,*) 'XOHAT = ', XOHAT
WRITE (NOUT,*) '(XLOWER,XUPPER) = (', XLOWER, ', ', XUPPER, ') '
END

```

Output

```

XOHAT = 45.3846
(XLOWER,XUPPER) = (20.2627,69.347)

```

Comments

Informational errors

Type Code

3	2	The slope is not significant at the $(100 - \text{CONPER})\%$ level. Confidence limits XLOWER and XUPPER cannot be obtained.
---	---	--

Description

Routine RINPF computes a confidence interval on the independent variable setting x_0 for a given response y_0 from the results of a straight line fit. Here, y_0 may represent the mean of k responses or the true mean response. The results of routine RLINE (page 86) or RONE (page 89) can be used for input into RINPF. The simple linear regression model is assumed,

$$y_i = \beta_0 + \beta_1 x + \varepsilon_i \quad i = 1, 2, \dots, n+k$$

where the ε_i 's are independently distributed normal errors with mean zero and variance σ^2/w_i . Here, n is the total number of observations used in the fit of the line, i.e., $n = \text{DFE} + \text{INTCEP} + 1$ where DFE is the degrees of freedom from the fitted regression. Also, k is the number of additional responses used to determine y_0 . The w_i 's are the weights that must be used in the fit of the model. The methodology is discussed by Graybill (1976, pages 280–283). For the case when $\text{IY0} = 1$, the estimate of σ^2 , s^2 (stored in S2), is the usual estimate of σ^2 from the fitted regression based on the first n observations. If $\text{IY0} = 0$, the estimate of σ^2 is a pooled estimator based on the fitted regression and the k responses that produce \bar{y}_0 .

This pooled estimator (stored in S2) is given by

$$s^2 = \frac{\sum_{i=1}^n w_i (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 + \sum_{i=n+1}^{n+k} w_i (y_i - \bar{y}_0)^2}{(n-2) + (k-1)}$$

where $(n-2) + (k-1)$ (stored in DFS2) is the pooled degrees of freedom for s^2 .

First, a point estimate \hat{x}_0 (stored in XOHAT) is computed by

$$\hat{x}_0 = \frac{y_0 - \hat{\beta}_0}{\hat{\beta}_1}$$

Then, a test of the hypothesis $H_0 : \beta_1 = 0$ vs. $H_a : \beta_1 \neq 0$ is performed. If H_0 is accepted, the model becomes $y_i = \beta_0 + \varepsilon_i$, and therefore no confidence interval exists for x_0 because it is no

longer in the model. In this case, a type 3 warning error is issued. If H_0 is rejected, a confidence interval exists and is computed for the case $\text{IY0} = 1$ by

$$\bar{x} + \frac{\hat{\beta}_1(y_0 - \bar{y})}{a} \pm \frac{ts}{a} \sqrt{\frac{a}{\sum_{i=1}^n w_i} + \frac{a}{\sum_{i=n+1}^{n+k} w_i} + \frac{(y_0 - \bar{y})^2}{\sum_{i=1}^n w_i (x_i - \bar{x})^2}}$$

where

$$a = \hat{\beta}_1^2 - \frac{t^2 s^2}{\sum_{i=1}^n w_i (x_i - \bar{x})^2}$$

and t is the $50 + \text{CONPER}/2$ percentile of the t distribution with DFS2 degrees of freedom. The interval actually has a confidence coefficient less than that specified by CONPER .

In the weighted case, which was discussed earlier, the means (stored in XYMEAN) and the sum of squares for x (stored in SSX) are all weighted. When the variances of the ε_i 's are all equal, ordinary least squares must be used, this corresponds to all $w_i = 1$.

Modifications are necessary to the preceding discussion for other cases. For the case when an intercept is not in the model, let $\beta_0 = 0$, $\hat{\beta}_0 = 0$ the pooled degrees of freedom of s^2 equal to $(n - 1) + (k - 1)$, and replace the first term under the square root symbol with zero, \bar{x} with zero, and \bar{y} with zero.

For the case of the true response mean, i.e., when $\text{IY0} = 1$, replace the second term under the square root symbol by zero.

RLSE

Fits a multiple linear regression model using least squares.

Required Arguments

Y— Vector of length NOBS containing the dependent (response) variable. (Input)

X— NOBS by NIND matrix containing the independent (explanatory) variables. (Input)

B— Vector of length $\text{INTCEP} + \text{NIND}$ containing a least-squares solution $\hat{\beta}$ for the regression coefficients. (Output)

For $\text{INTCEP} = 0$, the fitted value for observation I is $\text{B}(1) * \text{X}(I, 1) + \text{B}(2) * \text{X}(I, 2) + \dots + \text{B}(\text{NIND}) * \text{X}(I, \text{NIND})$.

For $\text{INTCEP} = 1$, the fitted value for observation I is $\text{B}(1) + \text{B}(2) * \text{X}(I, 1) + \dots + \text{B}(\text{NIND} + 1) * \text{X}(I, \text{NIND})$.

Optional Arguments

NOBS — Number of observations. (Input)

Default: NOBS = size (Y,1).

NIND — Number of independent (explanatory) variables. (Input)

Default: NIND = size (X,2).

LDX — Leading dimension of X exactly as specified in the dimension statement in the calling program. (Input)

Default: LDX = size (X,1).

INTCEP — Intercept option. (Input)

Default: INTCEP = 1.

INTCEP Action

0 An intercept is not in the model.

1 An intercept is in the model.

SST — Total sum of squares. (Output)

If INTCEP = 1, the total sum of squares is corrected for the mean.

SSE — Sum of squares for error. (Output)

FORTRAN 90 Interface

Generic: CALL RLSE (Y, X, B [, ...])

Specific: The specific interface names are S_RLSE and D_RLSE.

FORTRAN 77 Interface

Single: CALL RLSE (NOBS, Y, NIND, X, LDX, INTCEP, B, SST, SSE)

Double: The double precision name is DRLSE.

Example 1

A regression model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \varepsilon_i \quad i = 1, 2, \dots, 9$$

is fitted to data taken from Maindonald (1984, pages 203–204).

```
USE RLSE_INT
USE WRRRN_INT
USE UMACH_INT
INTEGER INTCEP, LDX, NCOEF, NIND, NOBS
```

```

PARAMETER (INTCEP=1, NIND=3, NOBS=9, LDX=NOBS, &
           NCOEF=INTCEP+NIND)
!
INTEGER   NOUT
REAL      B(NCOEF), SSE, SST, X(LDX,NIND), Y(NOBS)
!
DATA (X(1,J),J=1,NIND)/ 7.0, 5.0, 6.0/, Y(1)/ 7.0/
DATA (X(2,J),J=1,NIND)/ 2.0, -1.0, 6.0/, Y(2)/-5.0/
DATA (X(3,J),J=1,NIND)/ 7.0, 3.0, 5.0/, Y(3)/ 6.0/
DATA (X(4,J),J=1,NIND)/-3.0, 1.0, 4.0/, Y(4)/ 5.0/
DATA (X(5,J),J=1,NIND)/ 2.0, -1.0, 0.0/, Y(5)/ 5.0/
DATA (X(6,J),J=1,NIND)/ 2.0, 1.0, 7.0/, Y(6)/-2.0/
DATA (X(7,J),J=1,NIND)/-3.0, -1.0, 3.0/, Y(7)/ 0.0/
DATA (X(8,J),J=1,NIND)/ 2.0, 1.0, 1.0/, Y(8)/ 8.0/
DATA (X(9,J),J=1,NIND)/ 2.0, 1.0, 4.0/, Y(9)/ 3.0/
!
CALL RLSE (Y, X, B, SST=SST, SSE=SSE)
CALL WRRRN ('B', B)
CALL UMACH (2, NOUT)
WRITE (NOUT,*)
WRITE (NOUT,99999) 'SST = ', SST, ' SSE = ', SSE
99999 FORMAT (A7, F7.2, A7, F7.2)
END

```

Output

```

B
1  7.733
2 -0.200
3  2.333
4 -1.667

SST = 156.00  SSE = 4.00

```

Comments

1. Workspace may be explicitly provided, if desired, by use of R2SE/DR2SE. The reference is:

```
CALL R2SE (NOBS, Y, NIND, X, LDX, INTCEP, B, SST, SSE, R, LDR,
          DFE, NRMISS, WK)
```

The additional arguments are as follows:

R — INTCEP + NIND by INTCEP + NIND upper triangular matrix containing the *R* matrix from a *QR* decomposition of the matrix of regressors. (Output)
All of the diagonal element of *R* are taken to be nonnegative. The rank of the matrix of regressors is the number of positive diagonal elements, which equals NOBS - NRMISS - DFE.

LDR — Leading dimension of *R* exactly as specified in the dimension statement in the calling program. (Input)

DFE — Degrees of freedom for error. (Output)

NRMISS — Number of rows in the augmented matrix (X , Y) containing NaN (not a number). (Output)

If a row contains NaN, that row is excluded from all other computations.

WK — Work vector of length $5 * NIND + 4 * INTCEP + 2$.

2. Informational error

Type Code

3 1 The model is not full rank. There is not a unique least-squares solution. If the I -th diagonal element of R is zero, $B(I)$ is set to zero in order to compute a solution.

Description

Routine `RLSE` fits a multiple linear regression model with or without an intercept. If `INTCEP = 1`, the multiple linear regression model is

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik} + \varepsilon_i \quad i = 1, 2, \dots, n$$

where the observed values of the y_i 's (input in Y) constitute the responses or values of the dependent variable, the x_{i1} 's, x_{i2} 's, ..., x_{ik} 's (input in X) are the settings of the k (input in `NIND`) independent variables, $\beta_0, \beta_1, \dots, \beta_k$ are the regression coefficients whose estimated values are output in B , and the ε_i 's are independently distributed normal errors each with mean zero and variance σ^2 . Here, n is the number of valid rows in the augmented matrix (X , Y), i.e. n equals `NOBS - NRMISS` (the number of rows that do not contain NaN). If `INTCEP = 0`, β_0 is not included in the model.

Routine `RLSE` computes estimates of the regression coefficients by minimizing the sum of squares of the deviations of the observed response y_i from the fitted response

$$\hat{y}_i$$

for the n observations. This minimum sum of squares (the error sum of squares) is output and denoted by

$$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

In addition, the total sum of squares is output. For the case, `INTCEP = 1`, the total sum of squares is the sum of squares of the deviations of y_i from its mean

$$\bar{y}$$

—the so-called *corrected total sum of squares*; it is denoted by

$$SST = \sum_{i=1}^n (y_i - \bar{y})^2$$

For the case `INTCEP = 0`, the total sum of squares is the sum of squares of y_i —the so-called *uncorrected total sum of squares*; it is denoted by

$$\text{SST} = \sum_{i=1}^n y_i^2$$

See Draper and Smith (1981) for a good general treatment of the multiple linear regression model, its analysis, and many examples.

In order to compute a least-squares solution, `RLSE` performs an orthogonal reduction of the matrix of regressors to upper triangular form. If the user needs the upper triangular matrix output for subsequent computing, the routine `R2SE` can be invoked in place of `RLSE`. (See the description of `R` in Comment 1). The reduction is based on one pass through the rows of the augmented matrix (X, Y) using fast Givens transformations. (See routines `SROTMG` and `SROTM` Golub and Van Loan, 1983, pages 156-162, Gentleman, 1974.) This method has the advantage that the loss of accuracy resulting from forming the crossproduct matrix used in the normal equations is avoided.

With `INTCEP` = 1, the current means of the dependent and independent variables are used to internally center the data for improved accuracy. Let x_j be a column vector containing the j -th row of data for the independent variables. Let \bar{x}_i represent the mean vector for the independent variables given the data for rows 1, 2, ..., i . The current mean vector is defined to be

$$\bar{x}_i = \frac{\sum_{j=1}^i x_j}{i}$$

The i -th row of data has \bar{x}_i subtracted from it and is then weighted by $i/(i-1)$. Although a crossproduct matrix is not computed, the validity of this centering operation can be seen from the following formula for the sum of squares and crossproducts matrix:

$$\sum_{i=1}^n (x_i - \bar{x}_n)(x_i - \bar{x}_n)^T = \sum_{i=2}^n \frac{i}{i-1} (x_i - \bar{x}_i)(x_i - \bar{x}_i)^T$$

An orthogonal reduction on the centered matrix is computed. When the final computations are performed, the first row of `R` and the first element of `B` are updated so that they reflect the statistics for the original (uncentered) data. This means that the estimate of the intercept and the `R` matrix are for the uncentered data.

As part of the final computations, `RLSE` checks for linearly dependent regressors. If the i -th regressor is a linear combination of the first $i-1$ regressors, the i -th diagonal element of `R` is close to zero (exactly zero if infinite precision arithmetic could be used) prior to the final computations. In particular, linear dependence of the regressors is declared if any of the following three conditions is satisfied:

- A regressor equals zero.
- Two or more regressors are constant.
- The result of

$$\sqrt{1 - R_{i,1,2,\dots,i-1}^2}$$

is less than or equal to $100 \times \epsilon$ where ϵ is the machine epsilon. (For `RLSE`, $\epsilon = \text{AMACH}(4)$ and for `DRLSE`, $\epsilon = \text{DMACH}(4)$. See routines `AMACH` and `DMACH` (Reference Material)). Here, $R_{i,1,2,\dots,i-1}$ is

the multiple correlation coefficient of the i -th independent variable with the first $i - 1$ independent variables. If no intercept is in the model (`INTCEP = 0`), the “multiple correlation” coefficient is computed without adjusting for the mean.

On completion of the final computations, if the i -th regressor is declared to be linearly dependent upon the previous $i - 1$ regressors, then the i -th element of `B` and all elements in the i -th row of `R` are set to zero. Finally, if a linear dependence is declared, an informational (error) message, code 1, is issued indicating the model is not full rank.

Additional Example

Example 2

A weighted least-squares fit is computed using the model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i \quad i = 1, 2, \dots, 4$$

and weights $1/i^2$ discussed by Maindonald (1984, pages 67 - 68). In order to compute the weighted least-squares fit, using an ordinary least squares routine (`RLSE`), the regressors (including the column of ones for the intercept term as well as the independent variables) and the responses must be transformed prior to invocation of `RLSE`. The transformed regressors and responses can be computed by using routine `SHPROD` (IMSL MATH/LIBRARY). For the i -th case the corresponding response and regressors are multiplied by a square root of the i -th weight. Because the column of ones corresponding to the intercept term in the untransformed model, is transformed by the weights, this transformed column of ones must be input to the least squares subroutine as an additional independent variable along with the option `INTCEP = 0`.

In terms of the original, untransformed regressors and responses, the minimum sum of squares for error output in `SSE` is

$$SSE = \sum_{i=1}^n w_i (y_i - \hat{y}_i)^2$$

where here the weight $w_i = 1/i^2$. Also, since `INTCEP = 0`, the uncorrected total sum of squares is output in `SST`. In terms of the original untransformed responses,

$$SST = \sum_{i=1}^n w_i y_i^2$$

```

USE RLSE_INT
USE SHPROD_INT
USE WRRRN_INT
USE UMACH_INT

INTEGER    INTCEP, LDX, NCOEF, NIND, NOBS
PARAMETER (INTCEP=0, NIND=3, NOBS=4, LDX=NOBS, &
           NCOEF=INTCEP+NIND)
!

INTEGER    I, NOUT
REAL       B(NCOEF), SQRT, SSE, SST, W(NOBS), X(LDX,NIND), &
           Y(NOBS)
INTRINSIC  SQRT
!
```

```

DATA (X(1,J),J=1,NIND)/1.0, -2.0, 0.0/, Y(1)/-3.0/
DATA (X(2,J),J=1,NIND)/1.0, -1.0, 2.0/, Y(2)/ 1.0/
DATA (X(3,J),J=1,NIND)/1.0, 2.0, 5.0/, Y(3)/ 2.0/
DATA (X(4,J),J=1,NIND)/1.0, 7.0, 3.0/, Y(4)/ 6.0/
!
DO 10 I=1, NOBS
!
!           Assign weights
      W(I) = 1.0/I**2
!
!           Store square roots of weights
      W(I) = SQRT(W(I))
10 CONTINUE
!
!           Transform regressors
DO 20 J=1, NIND
      CALL SHPROD (NOBS, W, 1, X(:,J), 1, X(:,J), 1)
20 CONTINUE
!
!           Transform response
CALL SHPROD (NOBS, W, 1, Y, 1, Y, 1)
!
CALL RLSE (Y, X, B, INTCEP=INTCEP, SST=SST, SSE=SSE)
!
CALL WRRRN ('B', B)
CALL UMACH (2, NOUT)
WRITE (NOUT,*)
WRITE (NOUT,99999) 'SST = ', SST, ' SSE = ', SSE
99999 FORMAT (A7, F7.2, A7, F7.2)
END

```

Output

```

B
1 -1.431
2 0.658
3 0.748

SST = 11.94 SSE = 1.01

```

RCOV

Fits a multivariate linear regression model given the variance-covariance matrix.

Required Arguments

COV — $NIND + NDEP$ by $NIND + NDEP$ matrix containing the variance-covariance matrix or sum of squares and crossproducts matrix. (Input)

Only the upper triangle of **COV** is referenced. The first $NIND$ rows and columns correspond to the independent variables, and the last $NDEP$ rows and columns correspond to the dependent variables. If $INTCEP = 0$, **COV** contains raw sums of squares and crossproducts. If $INTCEP = 1$, **COV** contains sums of squares and crossproducts corrected for the mean. If weighting is desired, **COV** contains weighted sums of squares and crossproducts.

XYMEAN — Vector of length $NIND + NDEP$ containing variable means. (Input, if $INTCEP = 1$)
The first $NIND$ elements of **XYMEAN** are for the independent variables in the same order in which they appear in **COV**. The last $NDEP$ elements of **XYMEAN** are for the dependent variables in the same order in which they appear in **COV**. If weighting is desired, **XYMEAN** contains weighted means. If $INTCEP = 0$, **XYMEAN** is not referenced and can be a vector of length one.

SUMWTF — Sum of products of weights with frequencies. (Input, if $INTCEP = 1$)
In the ordinary case when weights and frequencies are all one, **SUMWTF** equals the number of observations.

B — $INTCEP + NIND$ by $NDEP$ matrix containing a least-squares solution \hat{B} for the regression coefficients. (Output)
Column j is for the j -th dependent variable. If $INTCEP = 1$, row 1 is for the intercept. Row $INTCEP + i$ is for the i -th independent variable. Elements of the appropriate row(s) of \hat{B} are set to 0.0 if linear dependence of the regressors is declared.

Optional Arguments

INTCEP — Intercept option. (Input)
Default: $INTCEP = 1$.

INTCEP **Action**

0 An intercept is not in the model.

1 An intercept is in the model.

NIND — Number of independent (explanatory) variables. (Input)
Default: $NIND = \text{size}(B,1) - INTCEP$.

NDEP — Number of dependent (response) variables. (Input)
Default: $NDEP = \text{size}(B,2)$.

LDCOV — Leading dimension of **COV** exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDCOV = \text{size}(COV,1)$.

TOL — Tolerance used in determining linear dependence. (Input)
For **RCOV**, $TOL = 100 * AMACH(4)$ is a common choice. See documentation for routine **AMACH** (Reference Material).
Default: $TOL = 1.e-5$ for single precision and $2.d-14$ for double precision.

LDB — Leading dimension of **B** exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDB = \text{size}(B,1)$.

R — INTCEP + NIND by INTCEP + NIND upper triangular matrix containing the R matrix from a Cholesky factorization $R^T R$ of the matrix of sums of squares and crossproducts of the regressors. (Output)
Elements of the appropriate row(s) of R are set to 0.0 if linear dependence of the regressors is declared.

LDR — Leading dimension of R exactly as specified in the dimension statement in the calling program. (Input)
Default: LDR = size (R,1).

IRANK — Rank of R . (Output)
IRANK less than INTCEP + NIND indicates that linear dependence of the regressors was declared. In this case, some rows of \hat{B} are set to zero.

SCPE — NDEP by NDEP matrix containing the error (residual) sums of squares and crossproducts. (Output)

LDSCPE — Leading dimension of SCPE exactly as specified in the dimension statement in the calling program. (Input)
Default: LDSCPE = size (SCPE,1).

FORTRAN 90 Interface

Generic: CALL RCOV (COV, XYMEAN, SUMWTF, B [,...])

Specific: The specific interface names are S_RCOV and D_RCOV.

FORTRAN 77 Interface

Single: CALL RCOV (INTCEP, NIND, NDEP, COV, LDCOV, XYMEAN, SUMWTF, TOL, B, LDB, R, LDR, IRANK, SCPE, LDSCPE)

Double: The double precision name is DRCOV.

Example

This example uses a data set from Draper and Smith (1981, pages 629 – 630). This data set is put into the matrix X by routine GDATA (See Chapter 19, Utilities). The first four columns are for the independent variables, and the last column is for the dependent variable. Routine CORVC in Chapter 3, “Correlation,” is invoked to compute the corrected sum of squares and crossproducts matrix. Then, RCOV is invoked to compute the regression coefficient estimates, the R matrix, and the sum of squares for error.

```
USE RCOV_INT
USE GDATA_INT
USE CORVC_INT
USE UMACH_INT
USE WRRRN_INT
```

```

PARAMETER (LDX=13, NDX=5, NIND=4, NDEP=1, LDCOV=NIND+NDEP, &
           LDSCPE=NDEP)
PARAMETER (INTCEP=1, LDB=INTCEP+NIND, LDR=INTCEP+NIND)
REAL      XYMEAN(NIND+NDEP)
REAL      X(LDX,NDX), B(LDB,NDEP), R(LDR,INTCEP+NIND)
REAL      COV(LDCOV,NIND+NDEP), SCPE(LDSCPE,NDEP), SUMWTF
INTEGER    INCD(1,1), ICOPT
!
CALL GDATA (5, X, NROW, NVAR)
!
ICOPT = 1
CALL CORVC (NVAR, X, COV, ICOPT=ICOPT, XMEAN=XYMEAN, SUMWT=SUMWTF)
!
CALL RCOV (COV, XYMEAN, SUMWTF, B, R=R, IRANK=IRANK, &
          SCPE=SCPE)
!
CALL UMACH (2, NOUT)
WRITE (NOUT,*) 'IRANK = ', IRANK, ' SCPE(1,1) = ', SCPE(1,1)
CALL WRRRN ('B', B, 1, INTCEP+NIND, 1)
CALL WRRRN ('R', R)
END

```

Output

```
IRANK = 5 SCPE(1,1) = 47.8638
```

```

          B
      1      2      3      4      5
62.40    1.55    0.51    0.10   -0.14

```

```

          R
      1      2      3      4      5
1     3.6    26.9   173.6   42.4   108.2
2     0.0    20.4    12.3   -18.3  -14.2
3     0.0     0.0    52.5     1.1  -54.6
4     0.0     0.0     0.0    12.5  -12.9
5     0.0     0.0     0.0     0.0    3.4

```

Comments

1. Informational error
Type Code

```

      3      1 COV is not a variance-covariance matrix within the tolerance defined
              by TOL.

```

Description

Routine `RCOV` fits a multivariate linear regression model given the variance-covariance matrix (or sum of squares and crossproducts matrix) for the independent and dependent variables. Typically, an intercept is to be in the model, and the corrected sum of squares and crossproducts matrix is input for `COV`. Routine `CORVC` in Chapter 3, “Correlation,” can be invoked to compute the corrected sum of squares and crossproducts matrix. Routine `RORDM` in Chapter 19, “Utilities,” can reorder this matrix, if required. If an intercept is not to be included in the model,

a raw (uncorrected) sum of squares and crossproducts matrix must be input for `COV`; and `SUMWTF` and `XYMEAN` are not used in the computations. Routine `MXTXF` (IMSL MATH/LIBRARY) can be used to compute the raw sum of squares and crossproducts matrix.

Routine `RCOV` is based on a Cholesky factorization of `COV`. Let k (input in `NIND`) be the number of independent variables, and d (input in `SUMWTF`) the denominator used in computing the x means (input in the first k locations of `XYMEAN`). The matrix R is formed by computing a Cholesky factorization of the first k rows and columns of `COV`. If `INTCEP` equals one, the k rows from this factorization are appended to the initial row

$$\sqrt{d}, \sqrt{d}\bar{x}_1, \dots, \sqrt{d}\bar{x}_k$$

The resulting R matrix is the Cholesky factor of the $X^T X$ matrix where X contains a column of ones as its first column and the independent variable settings as its remaining k columns.

Maindonald (1984, Chapter 3) discusses the Cholesky factorization as it applies to regression computations.

The routine `RCOV` checks sequentially for linear dependent regressors. Linear dependence of the regressors is declared if

$$1 - R_{i-1,2,\dots,i-1}^2$$

is less than or equal to `TOL`. Here, $R_{i-1,2,\dots,i-1}$ is the multiple correlation coefficient of the i -th independent variable with the first $i - 1$ independent variables. If no intercept is in the model (`INTCEP = 0`), the “multiple correlation” coefficient is computed without adjusting for the mean. When a dependence is declared, elements of the corresponding rows of R and B are set to zero. Maindonald (1984, Sections 3.3, 3.4, and 3.9) discusses these implementation details of the Cholesky factorization in regression problems.

RGIVN

Fits a multivariate linear regression model via fast Givens transformations.

Required Arguments

X — `|NROW|` by `NCOL` matrix containing the data. (Input)

`IIND` — Independent variable option. (Input)

`IIND` Meaning

< 0 The first `-IIND` columns of X contain the independent (explanatory) variables.

> 0 The `IIND` independent variables are specified by the column numbers in `INDIND`.

= 0 There are no independent variables.

The regressors are the intercept (if $INTCEP = 1$) and the independent variables. There are $INTCEP + |IIND|$ regression coefficients for each dependent variable.

INDIND — Index vector of length $IIND$ containing the column numbers of X that are the independent variables. (Input, if $IIND$ is positive)

If $IIND$ is nonpositive, $INDIND$ is not referenced and can be a vector of length one.

IDEP — Dependent variable option. (Input)

IDEP Meaning

< 0 The last $-IDEP$ columns of X contain the dependent (response) variables. That is, columns $NCOL + IDEP + 1, NCOL + IDEP + 2, \dots, NCOL$ contain the dependent variables.

> 0 The $IDEP$ dependent (response) variables are specified by the column numbers in $INDDEP$.

$= 0$ There are no dependent variables. (Generally, this option is not used. The R matrix from a QR decomposition of a matrix of regressors is computed.)

INDDEP — Index vector of length $IDEP$ containing the column numbers of X that are the dependent variables. (Input, if $IDEP$ is positive)

If $IDEP$ is nonpositive, $INDDEP$ is not referenced and can be a vector of length one.

B — $INTCEP + |IIND|$ by $|IDEP|$ matrix containing a least-squares solution

$$\hat{B}$$

for the regression coefficients on return from the final invocation of this routine. (Output, if $IDO = 0$ or 1 ; input/output, if $IDO = 2$ or 3)

If $INTCEP = 1$, row 1 is for the intercept. Row $INTCEP + I$ is for the I -th independent variable. Column j is for the j -th dependent variable.

IDO Action

1 or 2 A current least-squares solution is given by a solution x to the equation $Rx = B$.

0 or 3 A least-squares solution for the regression coefficients is returned in B . Elements of the appropriate row(s) of B are set to 0.0 if linear dependence of the regressors is declared.

If $IDEP = 0$, B is not referenced and can be a vector of length 1.

Optional Arguments

IDO — Processing option. (Input)

Default: IDO = 0.

IDO Action

- | | |
|---|--|
| 0 | This is the only invocation of <code>RGIVN</code> for this data set, and all the data are input at once. |
| 1 | This is the first invocation, and additional calls to <code>RGIVN</code> will be made. Initialization and updating for the data in <code>X</code> are performed. |
| 2 | This is an intermediate invocation of <code>RGIVN</code> , and updating for the data in <code>X</code> is performed. |
| 3 | This is the final invocation of this routine. Updating for the data in <code>X</code> and wrap-up computations are performed. |

NROW — The absolute value of `NROW` is the number of rows of data currently input in `X`.

(Input)

`NROW` may be positive, zero, or negative. Negative `NROW` means that the $-NROW$ rows of data are to be deleted from some aspects of the analysis, and this should be done only if `IDO` is 2 or 3 and the wrap-up computations have not been performed. When a negative value is input for `NROW`, it is assumed that each of the $-NROW$ rows of `X` has been input (with positive `NROW`) in previous invocations of `RGIVN`. Use of negative values of `NROW` should be made with care and with the understanding that `XMIN` and `XMAX` cannot be updated properly in this case. It is also possible that a constant variable in the remaining data will not be recognized as such.

Default: `NROW` = size (`X`,1).

NCOL — Number of columns in `X`. (Input)

Default: `NCOL` = size (`X`,2).

LDX — Leading dimension of `X` exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDX` = size (`X`,1).

INTCEP — Intercept option. (Input)

Default: `INTCEP` = 1.

INTCEP Action

- | | |
|---|-----------------------------------|
| 0 | An intercept is not in the model. |
| 1 | An intercept is in the model. |

IFRQ — Frequency option. (Input)

$IFRQ = 0$ means that all frequencies are 1.0. For positive $IFRQ$, column number $IFRQ$ of X contains the frequencies. If $X(I, IFRQ) = 0.0$, none of the remaining elements of row I of X are referenced, and updating of statistics is skipped for row I .

Default: $IFRQ = 0$.

IWT — Weighting option. (Input)

$IWT = 0$ means that all weights are 1.0. For positive IWT , column number IWT of X contains the weights.

Default: $IWT = 0$.

ICEN — Data centering option. (Input)

If $INTCEP = 0$, $ICEN$ must equal 0.

Default: $ICEN = 1$.

ICEN Action

0 No centering. This option should be used when (1) the data are already centered; (2) there is no intercept in the model; or (3) the independent variables for a large percentage of the data are zero, and sparsity of the problem needs to be preserved in order that the Givens rotations are performed quickly.

1 Variables are centered using the method of provisional means for improved accuracy of the computations. The final estimate for the intercept and the R matrix are given for the uncentered data. This option is generally recommended.

TOL — Tolerance used in determining linear dependence. (Input)

For $RGIVN$, $TOL = 100 * AMACH(4)$ is a common choice. See the documentation for routine *AMACH* (Reference Material).

Default: $TOL = 1.e-5$ for single precision and 2.D-14 for double precision.

LDB — Leading dimension of B exactly as specified in the dimension statement in the calling program. (Input)

Default: $LDB = \text{size}(B, 1)$.

R — $INTCEP + |IIND|$ by $INTCEP + |IIND|$ upper triangular matrix containing the R matrix from a QR decomposition of the matrix of regressors on return from the final invocation of this routine. (Output, if $IDO = 0$ or 1; input/output, if $IDO = 2$ or 3)

IDO Action

1 or 2 The current matrix of raw sums of squares and crossproducts for the regressors can be found as $R^T \cdot \text{diag}(D) \cdot R$ where $\text{diag}(D)$ is the diagonal matrix whose diagonal elements are the elements of the vector D .

0 or 3 The matrix of raw sums of squares and crossproducts for the regressors can be found as $R^T R$. Elements of the appropriate row(s) of R are set to 0.0 if linear dependence of the regressors is declared.

LDR — Leading dimension of R exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDR = \text{size}(R, 1)$.

D — Vector of length $\text{INTCEP} + |\text{IIND}|$ containing scale factors for fast Givens transformations. (Output, if $\text{IDO} = 0$ or 1; input/output, if $\text{IDO} = 2$ or 3)

IDO Action

1 or 2 D contains the current scale factors associated with the fast Givens transformations.

0 or 3 Each element of D is set to 1.0.

IRANK — Rank of R . (Output, if $\text{IDO} = 0$ or 3)
 IRANK less than $\text{INTCEP} + |\text{IIND}|$ indicates linear dependence of the regressors was declared.

DFE — Degrees of freedom for error on return from the final invocation of this routine. (Output, if $\text{IDO} = 0$ or 1; input/output, if $\text{IDO} = 2$ or 3)
Prior to the final invocation of RGIVN , DFE is the sum of the frequencies.

SCPE — $|\text{IDEP}|$ by $|\text{IDEP}|$ matrix containing error (residual) sums of squares and crossproducts. (Output, if $\text{IDO} = 0$ or 1; input/output, if $\text{IDO} = 2$ or 3)
 $\text{SCPE}(m, n)$ contains the current sum of crossproducts of residuals for the m -th and n -th dependent variables. If $\text{IDEP} = 0$, SCPE is not referenced and can be a 1 by 1 array.

LDSCPE — Leading dimension of SCPE exactly as specified in the dimension statement in the calling program. (Input)
Default: $\text{LDSCPE} = \text{size}(\text{SCPE}, 1)$.

NRMIS — Number of rows of data encountered in calls to RGIVN that contain any missing values for the independent, dependent, weight, or frequency variables. (Output, if $\text{IDO} = 0$ or 1; input/output, if $\text{IDO} = 2$ or 3)
NaN (not a number) is used as the missing value code. Any row of X containing NaN as a value of the independent, dependent, weight, or frequency variables is omitted from the analysis.

XMIN — Vector of length $\text{INTCEP} + |\text{IIND}|$ containing the minimum values for each of the regressors. (Output, if $\text{IDO} = 0$ or 1; input/output, if $\text{IDO} = 2$ or 3)

XMAX — Vector of length $\text{INTCEP} + |\text{IIND}|$ containing the maximum values for each of the regressors. (Output, if $\text{IDO} = 0$ or 1; input/output, if $\text{IDO} = 2$ or 3)

FORTRAN 90 Interface

Generic: CALL RGIVN (X, IIND, INDIND, IDEP, INDDEP, B [,...])

Specific: The specific interface names are S_RGIVN and D_RGIVN.

FORTRAN 77 Interface

Single: CALL RGIVN (IDO, NROW, NCOL, X, LDX, INTCEP, IIND, INDIND, IDEP, INDDEP, IFRQ, IWT, ICEN, TOL, B, LDB, R, LDR, D, IRANK, DFE, SCPE, LDSCPE, NRMISS, XMIN, XMAX)

Double: The double precision name is DRGIVN.

Example 1

The first example uses a data set from Draper and Smith (1981, pages 629-630). This data set is put into the matrix *X* by routine *GDATA* in Chapter 19, "Utilities". There is 1 dependent variable and 4 independent variables. *RGIVN* is invoked to fit the regression model with the *IDO* = 0 option, so all computations are performed in one call.

```
USE RGIVN_INT
USE GDATA_INT
USE WRRRN_INT
USE UMACH_INT
INTEGER LDB, LDcoef, LDR, LDSCPE, LDX, NCOEF, NCOL, NDEP, NRX
PARAMETER (LDSCPE=1, NCOEF=5, NCOL=5, NDEP=1, NRX=13, &
           LDB=NCOEF, LDcoef=NCOEF, LDR=NCOEF, LDX=NRX)
!
INTEGER I, IDEP, IIND, INDDEP(1), INDIND(1), &
        IRANK, NOBS, NOUT, NRMISS, NVAR
REAL B(LDB,NDEP), D(NCOEF), DFE, R(LDR,NCOEF), &
     SCPE(LDSCPE,NDEP), X(LDX,NCOL), XMAX(NCOEF), &
     XMIN(NCOEF)
!
CALL GDATA (5, X, NOBS, NVAR)
!
IIND = -4
IDEP = -1
CALL RGIVN (X, IIND, INDIND, IDEP, INDDEP, B, R=R, D=D, &
           IRANK=IRANK, DFE=DFE, SCPE=SCPE, NRMISS=NRMISS, &
           XMIN=XMIN, XMAX=XMAX)
!
CALL WRRRN ('B', B)
CALL WRRRN ('R', R)
CALL UMACH (2, NOUT)
WRITE (NOUT,*)
WRITE (NOUT,*) 'Regressor XMIN XMAX'
DO 10 I=1, NCOEF
   WRITE (NOUT,'(1X,I5,2X,2F9.1)') I, XMIN(I), XMAX(I)
10 CONTINUE
WRITE (NOUT,*) ' '
WRITE (NOUT,*) 'IRANK = ', IRANK
```



```

WRITE (NOUT,*) 'DFE = ', DFE, ' SCPE(1,1) = ', SCPE(1,1)
WRITE (NOUT,*) 'NRMISS = ', NRMISS
END

```

Output

```

B
1  62.41
2   1.55
3   0.51
4   0.10
5  -0.14

R
      1      2      3      4      5
1  3.6  26.9  173.6  42.4  108.2
2   0.0  20.4  12.3  -18.3  -14.2
3   0.0   0.0  52.5   1.1  -54.6
4   0.0   0.0   0.0  12.5  -12.9
5   0.0   0.0   0.0   0.0   3.4

Regressor      XMIN      XMAX
      1          1.0      1.0
      2          1.0     21.0
      3         26.0     71.0
      4          4.0     23.0
      5          6.0     60.0

IRANK =      5
DFE =      8.00000  SCPE(1,1) =      47.8637
NRMISS =      0

```

Comments

1. Workspace may be explicitly provided, if desired, by use of R2IVN/DR2IVN. The reference is:

```

CALL R2IVN (IDO, NROW, NCOL, X, LDX, INTCEP, IIND, INDIND, IDEP,
INDDEP, IFRQ, IWT, ICEN, TOL, B, LDB, R, LDR, D, IRANK, DFE,
SCPE, LDSCPE, NRMISS, XMIN, XMAX, WK)

```

The additional argument is:

WK — Work vector of length $\text{INTCEP} + |\text{IIND}| + |\text{IDEP}|$

2. Informational errors
Type Code

```

4      1  Negative weight encountered.
4      2  Negative frequency encountered.

```

Description

Routine RGIVN fits a multivariate linear regression model. (See the chapter introduction for a description of the multivariate linear regression model.) The routine RGIVN is designed so that

multiple invocations can be made. In this case, zero, one, or several rows of the data set can be input for each invocation of `RGIVN` (with `IDO = 1, 2, 2, ..., 2, 3`). Alternatively, one invocation of `RGIVN` (with `IDO = 0`) can be made with the entire data set contained in `X`. Routine `RSTAT` (page 152) can be invoked after the wrap-up computations are performed by `RGIVN` to compute and print summary statistics related to the fitted regression.

Routine `RGIVN` performs an orthogonal reduction of the matrix of regressors to upper triangular form. The reduction is based on fast Givens transformations. (See routines `SROTMG` and `SROTM`, Golub and Van Loan 1983, pages 156-162, Gentleman 1974.) This method has two main advantages: (1) the loss of accuracy resulting from forming the crossproduct matrix used in the normal equations is avoided, (2) data can be conveniently added or deleted to take advantage of the previous computations performed.

With `ICEN = 1`, the current means of the independent and dependent variables are used to center the data for improved accuracy. Let x_i be a column vector containing the i -th row of data for the independent variables. Let \bar{x}_i represent the mean vector for the independent variables given the data for observations 1, 2, ..., i . The mean vector is defined to be

$$\bar{x}_i = \frac{\sum_{j=1}^i w_j f_j x_j}{\sum_{j=1}^i w_j f_j}$$

where the w_j 's and f_j 's are the weights and frequencies, respectively. The i -th row of data has \bar{x}_i subtracted from it, and then $w_i f_i$ is multiplied by the factor a_i/a_{i-1} where

$$a_i = \sum_{j=1}^i w_j f_j$$

Although a crossproduct matrix is not computed, the validity of this centering operation can be seen from the following formula for the sum of squares and crossproducts matrix:

$$\sum_{i=1}^n w_i f_i (x_i - \bar{x}_n)(x_i - \bar{x}_n)^T = \sum_{i=2}^n \frac{a_i}{a_{i-1}} w_i f_i (x_i - \bar{x}_i)(x_i - \bar{x}_i)^T$$

An orthogonal reduction on the centered matrix is computed. When wrap-up computations (`IDO = 3` or `IDO = 0`) are performed, the first rows of `R` and `B` are updated so that they reflect the statistics for the original (uncentered) data. This means that the estimate of the intercept and the `R` matrix are for the uncentered data.

If the i -th regressor is a linear combination of the first $i - 1$ regressors, the i -th diagonal element of `R` will be close to zero (exactly zero if infinite precision arithmetic could be used) prior to the wrap-up computations. When performing the wrap-up computations, `RGIVN` checks sequentially for linear dependent regressors. Linear dependence of the regressors is declared if any of the following three conditions is satisfied:

- A regressor equals zero, as determined from `XMIN` and `XMAX`.
- Two or more regressors are constant, as determined from `XMIN` and `XMAX`.

$$\sqrt{1 - R_{i,1,2,\dots,i-1}^2}$$

is less than or equal to TOL. Here, $R_{i_1, 2, \dots, i-1}$ is the multiple correlation coefficient of the i -th independent variable with the first $i - 1$ independent variables. If no intercept is in the model (INTCEP = 0) the “multiple correlation” coefficient is computed without adjusting for the mean.

When a dependence is declared, R is changed in the wrap-up computations so as to reflect the deletion of the i -th regressor from the model. On completion of the wrap-up computations, if the i -th regressor is declared to be dependent upon the previous $i - 1$ regressors, then the R and \hat{B} matrices will have all elements in their i -th rows set to zero.

Additional Examples

Example 2

The data for the second example are taken from Maindonald (1984, pages 203–204). The data are saved in the matrix x . Here, the data are input into RGIVN a row at a time. The data set is small for clarity. However, the approach is generally useful when the data set is large and the entire data set cannot be stored in x . A multivariate regression model containing two dependent variables and three independent variables is fit.

```

USE RGIVN_INT
USE WRRRN_INT
USE UMACH_INT
INTEGER INTCEP, LDB, LDR, LDSCPE, LDX, NCOEF, NCOL, NDEP, &
        NIND, NOBS
PARAMETER (INTCEP=1, NCOL=5, NDEP=2, NIND=3, NOBS=9, &
        LDSCPE=NDEP, LDX=NOBS, NCOEF=INTCEP+NIND, LDB=NCOEF, &
        LDR=NCOEF)
!
INTEGER I, IDEP, IDO, IIND, INDDEP(1), INDIND(1), IRANK, &
        NOUT, NRMISS, NROW
REAL B(LDB, NDEP), D(NCOEF), DFE, R(LDR, NCOEF), &
        SCPE(LDSCPE, NDEP), TOL, X(LDX, NCOL), XMAX(NCOEF), &
        XMIN(NCOEF)
!
DATA (X(1, J), J=1, NCOL) / 7.0, 5.0, 6.0, 7.0, 1.0 /
DATA (X(2, J), J=1, NCOL) / 2.0, -1.0, 6.0, -5.0, 4.0 /
DATA (X(3, J), J=1, NCOL) / 7.0, 3.0, 5.0, 6.0, 10.0 /
DATA (X(4, J), J=1, NCOL) / -3.0, 1.0, 4.0, 5.0, 5.0 /
DATA (X(5, J), J=1, NCOL) / 2.0, -1.0, 0.0, 5.0, -2.0 /
DATA (X(6, J), J=1, NCOL) / 2.0, 1.0, 7.0, -2.0, 4.0 /
DATA (X(7, J), J=1, NCOL) / -3.0, -1.0, 3.0, 0.0, -6.0 /
DATA (X(8, J), J=1, NCOL) / 2.0, 1.0, 1.0, 8.0, 2.0 /
DATA (X(9, J), J=1, NCOL) / 2.0, 1.0, 4.0, 3.0, 0.0 /
!
NROW = 1
IIND = -NIND
IDEP = -NDEP
DO 10 I=1, 9
  IF (I .EQ. 1) THEN
    IDO = 1
  ELSE IF (I .EQ. 9) THEN
    IDO = 3
  ELSE

```

```

        IDO = 2
      END IF
      CALL RGIVN (X(I:I, 1:NCOL), IIND, INDIND, IDEP, INDDEP, &
        B, IDO=IDO, R=R, D=D, IRANK=IRANK, DFE=DFE, &
        SCPE=SCPE, NRMISS=NRMISS, XMIN=XMIN, XMAX=XMAX)

10 CONTINUE
!
      CALL WRRRN ('B', B)
      CALL WRRRN ('R', R)
      CALL WRRRN ('SCPE', SCPE)
      CALL UMACH (2, NOUT)
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Regressor   XMIN       XMAX'
      DO 20 I=1, NCOEF
        WRITE (NOUT, '(1X,I5,2X,2F9.1)') I, XMIN(I), XMAX(I)
20 CONTINUE
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'IRANK = ', IRANK
      WRITE (NOUT,*) 'DFE = ', DFE
      WRITE (NOUT,*) 'NRMISS = ', NRMISS
      END

```

Output

```

      B
      1      2
1  7.733  -1.633
2  -0.200  0.400
3  2.333  0.167
4  -1.667  0.667

      R
      1      2      3      4
1  3.00  6.00  3.00  12.00
2  0.00  10.00  4.00  2.00
3  0.00  0.00  4.00  2.00
4  0.00  0.00  0.00  6.00

      SCPE
      1      2
1  4.0  20.0
2  20.0  110.0

Regressor   XMIN   XMAX
      1      1.0   1.0
      2     -3.0   7.0
      3     -1.0   5.0
      4      0.0   7.0

IRANK = 4
DFE = 5.00000
NRMISS = 0

```

Example 3

The data for the third example are taken from Maindonald (1984, pages 104–106). The constant regressor and the independent variables X_1 , X_2 , and X_3 are linearly dependent

$$(X_3 = \frac{1}{2} + X_1 - \frac{1}{2}X_2)$$

```

USE RGIVN_INT
USE WRRRN_INT
USE UMACH_INT
INTEGER INTCEP, LDB, LDR, LDSCPE, LDX, NCOEF, NCOL, NDEP, &
        NIND, NOBS
PARAMETER (INTCEP=1, NCOL=5, NDEP=1, NIND=4, NOBS=9, &
        LDSCPE=NDEP, LDX=NOBS, NCOEF=INTCEP+NIND, LDB=NCOEF, &
        LDR=NCOEF)
!
INTEGER I, IDEP, IIND, INDDEP(1), INDIND(1), &
        IRANK, NOUT, NRMISS, NROW
REAL B(LDB,NDEP), D(NCOEF), DFE, R(LDR,NCOEF), &
        SCPE(LDSCPE,NDEP), TOL, X(LDX,NCOL), XMAX(NCOEF), &
        XMIN(NCOEF)
!
DATA (X(1,J),J=1,NCOL)/-1.0, 0.0, -0.5, 1.0, 0.0/
DATA (X(2,J),J=1,NCOL)/3.0, 0.0, 3.5, 1.0, 0.0/
DATA (X(3,J),J=1,NCOL)/2.0, -2.0, 3.5, -2.0, -2.0/
DATA (X(4,J),J=1,NCOL)/-2.0, -1.0, -1.0, 1.0, 1.0/
DATA (X(5,J),J=1,NCOL)/-1.0, 1.0, -1.0, -1.0, -1.0/
DATA (X(6,J),J=1,NCOL)/3.0, 3.0, 2.0, 1.0, 3.0/
DATA (X(7,J),J=1,NCOL)/2.0, 2.0, 1.5, 2.0, 4.0/
DATA (X(8,J),J=1,NCOL)/-2.0, -1.0, -1.0, -1.0, -2.0/
DATA (X(9,J),J=1,NCOL)/2.0, 1.0, 2.0, 1.0, 3.0/
!
NROW = NOBS
IIND = -NIND
IDEP = -NDEP
CALL RGIVN(X, IIND, INDIND, IDEP, INDDEP, B, R=R, D=D, &
        IRANK=IRANK, DFE=DFE, SCPE=SCPE, NRMISS=NRMISS, &
        XMIN=XMIN, XMAX=XMAX)
!
CALL WRRRN ('B', B)
CALL WRRRN ('R', R)
CALL UMACH (2, NOUT)
WRITE (NOUT,*)
WRITE (NOUT,*) 'Regressor Minimum Maximum'
DO 10 I=1, NCOEF
    WRITE (NOUT,'(1X,I5,2X,2F9.1)') I, XMIN(I), XMAX(I)
10 CONTINUE
WRITE (NOUT,*)
WRITE (NOUT,*) 'IRANK = ', IRANK
WRITE (NOUT,*) 'DFE = ', DFE, ' SCPE(1,1) = ', SCPE(1,1)
WRITE (NOUT,*) 'NRMISS = ', NRMISS
END

```

Output

```
      B
1    0.056
2    0.167
3    0.500
4    0.000
5    1.000

      R
      1      2      3      4      5
1    3.000  2.000  1.000  3.000  1.000
2    0.000  6.000  2.000  5.000  1.000
3    0.000  0.000  4.000 -2.000  2.000
4    0.000  0.000  0.000  0.000  0.000
5    0.000  0.000  0.000  0.000  3.000

Regressor  Minimum  Maximum
      1          1.0      1.0
      2         -2.0      3.0
      3         -2.0      3.0
      4         -1.0      3.5
      5         -2.0      2.0

IRANK = 4
DFE = 5.00000  SCPE(1,1) = 6.00000
NMISS = 0
```

RGLM

Fits a multivariate general linear model.

Required Arguments

X — |NROW| by NCOL matrix containing the data. (Input)

INDCL — Index vector of length NCLVAR containing the column numbers of *X* that are the classification variables. (Input)

NVEF — Vector of length NEF containing the number of variables associated with each effect in the model. (Input)

INDEF — Index vector of length $NVEF(1) + NVEF(2) + \dots + NVEF(NEF)$. (Input)
The first $NVEF(1)$ elements give the column numbers of *X* for each variable in the first effect. The next $NVEF(2)$ elements give the column numbers for each variable in the second effect. ... The last $NVEF(NEF)$ elements give the column numbers for each variable in the last effect.

IDEP — Dependent variable option. (Input)
The absolute value of *IDEP* is the number of dependent (response) variables. The sign of *IDEP* specifies the following options:

IDEP Meaning

- < 0 The last $-IDEP$ columns of X contain the dependent (response) variables. That is, columns $NCOL + IDEP + 1$, $NCOL + IDEP + 2$, ..., $NCOL$ contain the dependent variables.
- > 0 The data for the $IDEP$ dependent variables are in the columns of X whose column numbers are given by the elements of $INDDEP$.
- = 0 There are no dependent variables. (Generally, this option is not used. However, it is possible to get the R matrix from a QR decomposition of a matrix of regressors in this way.)

INDDEP — Index vector of length $IDEP$ containing the column numbers of X that are the dependent (response) variables. (Input, if $IDEP$ is positive)
If $IDEP$ is nonpositive, $INDDEP$ is not referenced and can be a vector of length one.

MAXCL — An upper bound on the sum of the number of distinct values taken on by each classification variable. (Input)

B — $NCOEF$ by $|IDEP|$ matrix containing on return from the final invocation of this routine a least-squares solution \hat{B} for the regression coefficients. (Output, if $IDO = 0$ or 1; input/output, if $IDO = 2$ or 3)
Here, $NCOEF = IRBEF(NEF + 1) - 1$ is the number of coefficients in the model. If $INTCEP = 1$, row 1 is for the intercept. Column j is for the j -th dependent variable.

IDO Action

- 1 or 2 A current least-squares solution is given by a solution x to the equation $R * x = B$
- 0 or 3 A least-squares solution for the regression coefficients is returned in B . Elements of the appropriate row(s) of B are set to 0.0 if linear dependence of the regressors is declared.

Optional Arguments

IDO — Processing option. (Input)
Default: $IDO = 0$.

IDO Action

- 0 This is the only invocation of `RGLM` for this data set, and all the data are input at once.
- 1 This is the first invocation, and additional calls to `RGLM` will be made. Initialization and updating for the data in X are performed.

- 2 This is an intermediate invocation of `RGLM`, and updating for the data in `X` is performed.
- 3 This is the final invocation of this routine. Updating for the data in `X` and wrap-up computation are performed.

NROW — The absolute value of `NROW` is the number of rows of data currently input in `X`. (Input)

`NROW` may be positive, zero, or negative. Negative `NROW` means that the $-NROW$ rows of data are to be deleted from some aspects of the analysis, and this should be done only if `IDO` is 2 or 3 and the wrap-up computations have not been performed. When a negative value is input for `NROW`, it is assumed that each of the $-NROW$ rows of `X` has been input (with positive `NROW`) in previous invocations of `RGLM`. Use of negative values of `NROW` should be made with care and with the understanding that `XMIN`, `XMAX`, and `CLVAL` cannot be updated properly in this case. It is also possible that a constant variable in the remaining data will not be recognized as such.

Default: `NROW = size (X,1)`.

NCOL — Number of columns in `X`. (Input)

Default: `NCOL = size (X,2)`.

LDX — Leading dimension of `X` exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDX = size (X,1)`.

INTCEP — Intercept option. (Input)

Default: `INTCEP = 1`.

INTCEP Action

0 An intercept is not in the model.

1 An intercept is in the model.

NCLVAR — Number of classification variables. (Input)

Default: `NCLVAR = size (INDCL,1)`.

NEF — Number of effects (sources of variation) in the model excluding error. (Input)

Default: `NEF = size (NVEF,1)`.

IFRQ — Frequency option. (Input)

`IFRQ = 0` means that all frequencies are 1.0. For positive `IFRQ`, column number `IFRQ` of `X` contains the frequencies. If `X(I, IFRQ) = 0.0`, none of the remaining elements of row `I` of `X` are referenced and updating of statistics is skipped for row `I`.

Default: `IFRQ = 0`.

IWT — Weighting option. (Input)

$IWT = 0$ means that all weights are 1.0. For positive IWT , column number IWT of X contains the weights.

Default: $IWT = 0$.

IDUMMY — Dummy variable option. (Input)

Default: $IDUMMY = 1$.

Some indicator variables are defined for the I -th class variable as follows: Let $J = NCLVAL(1) + NCLVAL(2) + \dots + NCLVAL(I - 1)$. $NCLVAL(I)$ indicator variables are defined such that for $K = 1, 2, \dots, NCLVAL(I)$ the K -th indicator variable for observation number $IOBS$ takes the value 1.0 if $X(IOBS, INDCL(I)) = CLVAL(J + K)$ and equals 0.0 otherwise. Dummy variables are generated from these indicator variables, and restrictions may be applied as given by the following:

IDUMMY Description

- 0 The $NCLVAL(I)$ indicator variables are the dummy variables. The usual balanced-data restrictions on the regression parameters are applied as part of the wrap-up computations regardless of whether the data are balanced.
- 1 The $NCLVAL(I)$ indicator variables are the dummy variables.
- 2 $NCLVAL(I) - 1$ indicator variables are used as the dummy variables. The indicator variable associated with the class value given in the first row of X on the first invocation is omitted.

ICEN — Data centering option. (Input)

If $INTCEP = 0$, $ICEN$ must equal 0.

Default: $ICEN = 1$.

ICEN Action

- 0 No centering. This option should be used when (1) the data are already centered, (2) there is no intercept in the model, or (3) the regressors for a large percentage of the data are zero, and sparsity of the problem needs to be preserved in order that the fast Givens transformations are performed quickly.
- 1 Variables are centered using the method of provisional means for improved accuracy of the computations. The final estimate for the intercept along with the R matrix are given for the uncentered data. This option is generally recommended.

TOL — Tolerance used in determining linear dependence. (Input)

For $RGLM$, $TOL = 100 * AMACH(4)$ is a common choice. See the documentation for IMSL routine *AMACH* (Reference Material).

Default: $TOL = 1.e-5$ for single precision and $2.d-14$ for double precision.

NCLVAL — Vector of length `NCLVAR` containing the number of values taken on by each classification variable. (Output, if `IDO = 0` or `1`; input/output, if `IDO = 2` or `3`)
`NCLVAL(I)` is the number of distinct values for the `I`-th classification variable.

CLVAL — Vector of length `NCLVAL(1) + NCLVAL(2) + ... + NCLVAL(NCLVAR)` containing the values of the classification variables. (Output, if `IDO = 0` or `1`; input/output, if `IDO = 2` or `3`)

Since in general the length of `CLVAL` will not be known in advance, `MAXCL` (an upper bound for this length) should be used for purposes of dimensioning `CLVAL`. The first `NCLVAL(1)` elements contain the values of the first classification variable. The next `NCLVAL(2)` elements contain the values of the second classification variable. ... The last `NCLVAL(NCLVAR)` elements contain the values of the last classification variable. If `IDUMMY = 0` or `1`, the values are in ascending order for each classification variable. If `IDUMMY = 2`, the last value for each classification variable is the value associated with the indicator variable omitted from the model. The remaining values for each classification variable are in ascending order.

IRBEF — Index vector of length `NEF + 1`. (Output, if `IDO = 0` or `1`; input/output, if `IDO = 2` or `3`)

For `I = 1, 2, ..., NEF`, rows `IRBEF(I), IRBEF(I) + 1, ..., IRBEF(I + 1) - 1` of `B` correspond to the `I`-th effect.

LDB — Leading dimension of `B` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDB = size(B,1)`.

R — `NCOEF` by `NCOEF` upper triangular matrix containing, on return from the final invocation of this routine, the `R` matrix from a `QR` decomposition of the matrix of regressors. (Output, if `IDO = 0` or `1`; input/output, if `IDO = 2` or `3`)

Upon completion of the wrap-up computations, a zero row indicates a nonfull rank model. If `IDUMMY = 0`, a negative diagonal element of `R` indicates that the associated row corresponds to a summation restriction.

LDR — Leading dimension of `R` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDR = size(R,1)`.

D — Vector of length `NCOEF`. (Output, if `IDO = 0` or `1`; input/output, if `IDO = 2` or `3`)

IDO Action

1 or 2 `D` contains the current scale factors associated with the fast Givens transformations. The current matrix of uncorrected sums of squares and crossproducts for the regressors can be found as $R^T \cdot \text{diag}(D) \cdot R$ where `diag(D)` is the diagonal matrix whose diagonal elements are the elements of `D`.

0 or 3 Each element of `D` is set to 1.0.

IRANK — Rank of R . (Output, if $IDO = 0$ or 3)
IRANK less than NCOEF indicates linear dependence of the regressors was declared.

DFE — Degrees of freedom for error on return from the final invocation of this routine.
(Output, if $IDO = 0$ or 1 ; input/output, if $IDO = 2$ or 3)
Prior to the final invocation, DFE is the sum of the frequencies.

SCPE — |IDEP| by |IDEP| matrix containing error (residual) sums of squares and crossproducts. (Output, if $IDO = 0$ or 1 ; input/output, if $IDO = 2$ or 3)
SCPE(M, N) is the current sum of crossproducts of residuals for the M-th and N -th dependent variables.

LDSCPE — Leading dimension of SCPE exactly as specified in the dimension statement in the calling program. (Input)
Default: LDSCPE = size (SCPE,1).

NRMIS — Number of rows of data encountered in calls to RGLM containing NaN (not a number) for the independent, dependent, weight, and/or frequency variables. (Output, if $IDO = 0$ or 1 , input/output, if $IDO = 2$ or 3)
If a row of data contains NaN for any of these variables, that row is excluded from the computations.

XMIN — Vector of length NCOEF containing the minimum values for each of the regressors.
(Output, if $IDO = 0$ or 1 ; input/output, if $IDO = 2$ or 3)

XMAX — Vector of length NCOEF containing the maximum values for each of the regressors.
(Output, if $IDO = 0$ or 1 ; input/output, if $IDO = 2$ or 3)

FORTRAN 90 Interface

Generic: CALL RGLM (X, INDCL, NVEF, INDEF, IDEP, INDDEP, MAXCL,
B [,...])

Specific: The specific interface names are S_RGLM and D_RGLM.

FORTRAN 77 Interface

Single: CALL RGLM (IDO, NROW, NCOL, X, LDX, INTCEP, NCLVAR, INDCL,
NEF, NVEF, INDEF, IDEP, INDDEP, IFRQ, IWT, IDUMMY, ICEN,
TOL, MAXCL, NCLVAL, CLVAL, IRBEF, B, LDB, R, LDR, D, IRANK,
DFE, SCPE, LDSCPE, NRMIS, XMIN, XMAX)

Double: The double precision name is DRGLM.

Example 1

A one-way analysis of covariance model is fitted to the turkey data discussed by Draper and Smith (1981, pages 243–249). The response variable is turkey weight y (in pounds). There are

three groups of turkeys corresponding to the three states where they were reared. The age of a turkey (in weeks) is the covariate. The explanatory variables are group, age, and interaction. The model is

$$y_{ij} = \mu + \alpha_i + \beta x_{ij} + \beta_i x_{ij} + \varepsilon_{ij} \quad i = 1, 2, 3; j = 1, 2, \dots, n_i$$

where $\alpha_3 = 0$ and $\beta_3 = 0$. Here, the IDUMMY = 2 option is used. The fitted model gives three separate lines, one for each state where the turkeys were reared.

```

USE IMSL_LIBRARIES
!
                                SPECIFICATIONS FOR PARAMETERS
INTEGER      IDEP, INTCEP, LDB, LDR, LDSCPE, LDX, MAXB, MAXCL, &
              NCLVAR, NCOL, NEF, NROW
PARAMETER    (IDEP=1, INTCEP=1, LDX=13, MAXB=6, MAXCL=3, NCLVAR=1, &
              NCOL=3, NEF=3, NROW=13, LDB=MAXB, LDR=MAXB, &
              LDSCPE=IDEP)
!
INTEGER      I, IDUMMY, INDCL(NCLVAR), INDDEP(IDEP), &
              INDEF(4), IRANK, IRBEF(NEF+1), J, &
              NCLVAL(NCLVAR), NCOEF, NOUT, NRMISS, NVEF(NEF)
REAL         B(LDB, IDEP), CLVAL(MAXCL), D(MAXB), DFE, &
              R(LDR, MAXB), SCPE(LDSCPE, IDEP), TOL, X(LDX, NCOL), &
              XMAX(MAXB), XMIN(MAXB)
CHARACTER    CLABEL(7)*6, RLABEL(1)*4
!
DATA (X(1, J), J=1, 3) /25, 13.8, 3/
DATA (X(2, J), J=1, 3) /28, 13.3, 1/
DATA (X(3, J), J=1, 3) /20, 8.9, 1/
DATA (X(4, J), J=1, 3) /32, 15.1, 1/
DATA (X(5, J), J=1, 3) /22, 10.4, 1/
DATA (X(6, J), J=1, 3) /29, 13.1, 2/
DATA (X(7, J), J=1, 3) /27, 12.4, 2/
DATA (X(8, J), J=1, 3) /28, 13.2, 2/
DATA (X(9, J), J=1, 3) /26, 11.8, 2/
DATA (X(10, J), J=1, 3) /21, 11.5, 3/
DATA (X(11, J), J=1, 3) /27, 14.2, 3/
DATA (X(12, J), J=1, 3) /29, 15.4, 3/
DATA (X(13, J), J=1, 3) /23, 13.1, 3/
DATA INDCL/3/, NVEF/1, 1, 2/, INDEF/3, 1, 1, 3/, INDDEP/2/
DATA CLABEL/' ', 'MU', 'ALPHA1', 'ALPHA2', 'BETA', 'BETA1', &
           'BETA2'/
DATA RLABEL/'NONE'/
!
IDUMMY = 2
CALL RGLM (X, INDCL, NVEF, INDEF, IDEP, INDDEP, MAXCL, B, &
           IDUMMY=IDUMMY, NCLVAR=NCLVAR, NCLVAL=NCLVAL, &
           CLVAL=CLVAL, IRBEF=IRBEF, R=R, D=D, IRANK=IRANK, &
           DFE=DFE, SCPE=SCPE, NRMISS=NRMISS, XMIN=XMIN, &
           XMAX=XMAX)
!
CALL UMACH (2, NOUT)
WRITE (NOUT, *) 'NRMISS = ', NRMISS
WRITE (NOUT, *) 'IRANK = ', IRANK, ' DFE = ', DFE, ' ' // &
           ' SCPE(1,1) = ', SCPE(1,1)
J = 0

```

```

DO 10 I=1, NCLVAR
  CALL WRRRN ('Class values', CLVAL((J+1):), 1, NCLVAL(I), 1)
  J = J + NCLVAL(I)
10 CONTINUE
NCOEF = IRBEF(NEF+1) - 1
CALL WRRRN ('XMIN', XMIN, 1, NCOEF, 1)
CALL WRRRN ('XMAX', XMAX, 1, NCOEF, 1)
CALL WRIRN ('IRBEF', IRBEF, 1, NEF+1, 1)
CALL WRRRN ('R-MATRIX', R)
CALL WRRRL ('B', B, RLABEL, CLABEL, 1, NCOEF, 1)
!
END

```

Output

```

NRMISS = 0
IRANK = 6 DFE = 7.00000 SCPE(1,1) = 0.706176

```

```

Class values
  1      2      3
1.000  2.000  3.000

```

```

XMIN
  1      2      3      4      5      6
1.00  0.00  0.00  20.00  0.00  0.00

```

```

XMAX
  1      2      3      4      5      6
1.00  1.00  1.00  32.00  32.00  29.00

```

```

IRBEF
  1  2  3  4
  2  4  5  7

```

```

R-MATRIX
  1      2      3      4      5      6
1  3.61  1.11  1.11  93.47  28.29  30.51
2      1.66 -0.74 -1.02  42.43 -20.34
3      1.49  3.73  0.00  40.99
4      11.66  7.80  0.43
5      5.49 -0.61
6      2.11

```

```

B
  MU      ALPHA1      ALPHA2      BETA      BETA1      BETA2
2.475    -3.454      -2.775      0.445      0.06104    0.025

```

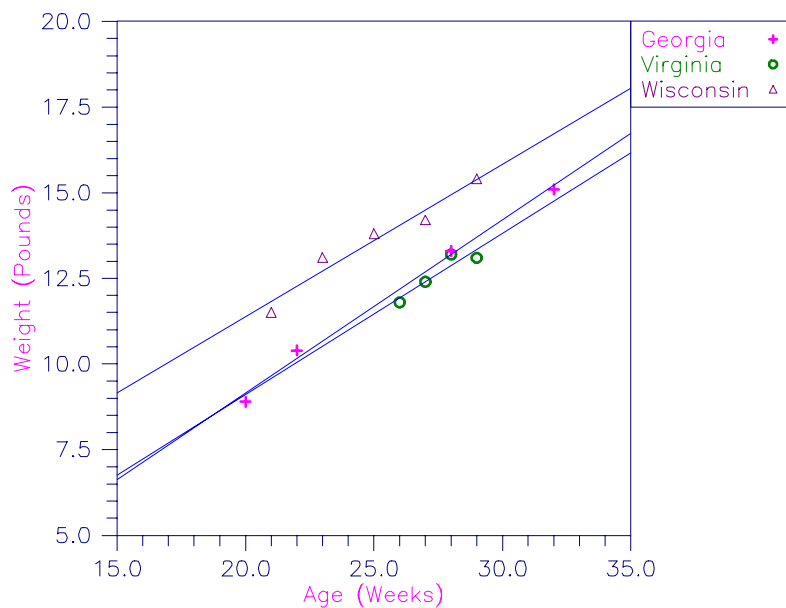


Figure 2-4 Plot of Turkey Weights and Fitted Lines by State

Comments

1. Workspace may be explicitly provided, if desired, by use of R2LM/DR2LM. The reference is:

```
CALL R2LM (IDO, NROW, NCOL, X, LDX, INTCEP, NCLVAR, INDCL, NEF,
NVEF, INDEF, IDEP, INDEP, IFRQ, IWT, IDUMMY, ICEN, TOL, MAXCL,
NCLVAL, VAL, IRBEF, B, LDB, R, LDR, D, IRANK, DFE, SCPE, LDSCPE,
NRMISS, XMIN, XMAX, IWK, WK)
```

The additional arguments are as follows:

IWK — Work vector of length $\max(\text{MAXB}, \text{NCLVAR})$.

WK — Work vector of length $\text{MAXB} + |\text{IDEP}| + 2$.

2. Informational errors

Type Code

4	1	Negative weight encountered.
4	2	Negative frequency encountered.
4	7	MAXCL is too small. Increase MAXCL and the dimension of CLVAL.
4	8	LDB or LDR is too small. One or more of the dimensions of B, R, D, XMIN, and XMAX must be increased.

3. Let the data matrix $x = (A, B, X_1, Y)$ where A and B are classification variables, X_1 is a continuous independent variable, and Y is a response variable. The model containing an

intercept and the effects A , B , AB , X_1 , AX_1 , BX_1 , and ABX_1 is specified as follows:
`INTCEP = 1, NCLVAR = 2, INDCL = (1, 2), NEF = 7, NVEF = (1, 1, 2, 1, 2, 2, 3), INDEF = (1, 2, 1, 2, 3, 1, 3, 2, 3, 1, 2, 3), IDEP = 1, and INDEP = (4).`

For this model suppose `NCLVAL(1) = 2, NCLVAL(2) = 3, and CLVAL = (1.0., 2.0, 1.0., 2.0, 3.0)`. Let A_1, A_2, B_1, B_2 , and B_3 , be the associated indicator variables. For each `IDUMMY` option the regressors following the intercept in their order of appearance in the model are given as follows:

IDUMMY Regressors

0 or 1 $A_1, A_2, B_1, B_2, B_3, A_1B_1, A_1B_2, A_1B_3, A_2B_1, A_2B_2, A_2B_3, X_1, A_1X_1, A_2X_1, B_1X_1, B_2X_1, B_3X_1, A_1B_1X_1, A_1B_2X_1, A_1B_3X_1, A_2B_1X_1, A_2B_2X_1, A_2B_3X_1$

2 $A_1, B_1, B_2, A_1B_1, A_1B_2, X_1, A_1X_1, B_1X_1, B_2X_1, A_1B_1X_1, A_1B_2X_1$

Within a group of regressors corresponding to an interaction effect, the indicator variables composing the regressors change most rapidly for the last classification variable, change next most rapidly for the next to last classification variable, etc.

4. If `NROW` is negative, no downdating of `XMIN, XMAX, NCLVAL, and CLVAL` can occur.

Description

Routine `RGLM` fits a multivariate linear regression model. (See the chapter introduction for a description of the multivariate linear regression model.) The routine `RGLM` is designed so that multiple invocations can be made. In this case, zero, one, or several rows of the data set can be input for each invocation of `RGLM` (with `IDO = 1, 2, 2, ..., 2, 3`). Alternatively, one invocation of `RGLM` (with `IDO = 0`) can be made with the entire data set contained in `X`. Routines `RSTAT` (page 152) and `RCASE` (page 204) can be invoked after the wrap-up computations are performed by `RGLM` to compute and print summary statistics and case statistics related to the fitted regression.

The data matrix can contain classification variables as well as continuous variables. The specification of a general linear model through the arguments `INTCEP, NCLVAR, INDCL, NEF, NVEF, INDEF` is discussed in the chapter introduction.

Regressors for effects composed solely of continuous variables are generated as powers and crossproducts. Consider a data matrix containing continuous variables as columns 3 and 4. The effect (3, 3) generates a regressor whose i -th value ($i = 1, 2, \dots, n$) is the square of the i -th value in column 3. The effect (3, 4) generates a regressor whose i -th value is the product of the i -th value in column 3 with the i -th value in column 4.

Regressors for an effect containing a single classification variable are generated using indicator variables. Let the classification variable A take on values a_1, a_2, \dots, a_n (stored in that order in `CLVAL`). From this classification variable, n indicator variables I_k are created. For $k = 1, 2, \dots, n$ we have

$$I_k = \begin{cases} 1 & \text{if } A = a_k \\ 0 & \text{otherwise} \end{cases}$$

For each classification variable, another set of variables is created from the indicator variables. We call these new variables *dummy variables*. Dummy variables are generated from the indicator variables in one of two manners: (1) the dummies are the n indicator variables, or (2) the dummies are the first $n - 1$ indicator variables. In particular, for $IDUMMY = 0$ or $IDUMMY = 1$, the dummy variables are $A_k = I_k$ ($k = 1, 2, \dots, n$). For $IDUMMY = 2$, the dummy variables are $A_k = I_k$ ($k = 1, 2, \dots, n - 1$).

Let m_j be the number of dummies generated for the j -th classification variable. Suppose there are two classification variables A and B with dummies A_1, A_2, \dots, A_{m_1} and B_1, B_2, \dots, B_{m_2} , respectively. The regressors generated for an effect composed of two classification variables A and B are

$$\begin{aligned} & A \otimes B \\ &= (A_1, A_2, \dots, A_{m_1}) \otimes (B_1, B_2, \dots, B_{m_2}) \\ &= (A_1 B_1, A_1 B_2, \dots, A_1 B_{m_2}, A_2 B_1, A_2 B_2, \dots, A_2 B_{m_2}, A_{m_1} B_1, A_{m_1} B_2, \dots, A_{m_1} B_{m_2}) \end{aligned}$$

More generally, the regressors generated for an effect composed of several classification variables and several continuous variables are given by the Kronecker products of variables, where the order of the variables is specified in `INDEF`. Consider a data matrix containing classification variables in columns 1 and 2 and continuous variables in columns 3 and 4. Label these four columns A, B, X_1 , and X_2 , respectively. The regressors generated by the effect (1, 2, 3, 4) are $A \otimes B \otimes X_1 X_2$.

Routine `RGLM` performs an orthogonal reduction of the matrix of regressors to upper triangular form. The reduction is based on fast Givens transformations. (See routines `SROTMG` and `SROTM`, Golub and Van Loan 1983, pages 156-162, Gentleman 1974.) This method has two main advantages: (1) the loss of accuracy resulting from forming the crossproduct matrix used in the normal equations is avoided, and (2) data can be conveniently added or deleted to take advantage of the previous computations performed.

With `ICEN = 1`, the current means of the regressors and dependent variables are used to center the data for improved accuracy. Let x_i be a column vector containing the i -th row of data for the regressors. Let \bar{x}_i represent the mean vector for the regressors given the data for observations 1, 2, ..., i . The mean vector is defined to be

$$\bar{x}_i = \frac{\sum_{j=1}^i w_j f_j x_j}{\sum_{j=1}^i w_j f_j}$$

where the w_j 's and f_j 's are the weights and frequencies, respectively. The i -th row of data has \bar{x}_i subtracted from it, and then, $w_j f_j$ is multiplied by the factor a_i/a_{i-1} where

$$a_i = \sum_{j=1}^i w_j f_j$$

Although a crossproduct matrix is not computed, the validity of this centering operation can be seen from the following formula for the sum of squares and crossproducts matrix:

$$\sum_{i=1}^n w_i f_i (x_i - \bar{x}_n)(x_i - \bar{x}_n)^T = \sum_{i=2}^n \frac{a_i}{a_{i-1}} w_i f_i (x_i - \bar{x}_i)(x_i - \bar{x}_i)^T$$

An orthogonal reduction on the centered matrix is computed. When wrap-up computations (IDO = 3 or IDO = 0) are performed, the first rows of R and B are updated so that they reflect the statistics for the original (uncentered) data. This means that the R matrix and the estimate of the intercept are for the uncentered data.

An orthogonal reduction on the centered matrix is computed. When wrap-up computations (IDO = 3 or IDO = 0) are performed, the first rows of R and B are updated so that they reflect the statistics for the original (uncentered) data. This means that the estimate of the intercept and the R matrix are for the uncentered data.

If the i -th regressor is a linear combination of the first $i - 1$ regressors, the i -th diagonal element of R will be close to zero (exactly zero if infinite precision arithmetic could be used) prior to the wrap-up computations. When performing the wrap-up computations, RGLM checks sequentially for linear dependent regressors. Linear dependence of the regressors is declared if any of the following three conditions is satisfied:

- A regressor equals zero, as determined from XMIN and XMAX.
- Two or more regressors are constant, as determined from XMIN and XMAX.
- The product of

$$\sqrt{1 - R_{i-1,2,\dots,i-1}^2}$$

is less than or equal to TOL. Here $R_{i-1,2,\dots,i-1}$ is the multiple correlation coefficient of the i -th regressor with the first $i - 1$ regressors. If no intercept is in the model (INTCEP = 0) the 'multiple correlation' coefficient is computed without adjusting for the mean.

When a dependence is declared, R is changed in the wrap-up computations so as to reflect the deletion of the i -th regressor from the model. On completion of the wrap-up computations, if the i -th regressor is declared to be dependent upon the previous $i - 1$ regressors, then the R and B matrices will have all elements in their i -th rows set to zero.

Additional Examples

Example 2

A two-way analysis-of-variance model is fitted to balanced data discussed by Snedecor and Cochran (1967, Table 12.5.1, page 347). The responses are the weight gains (in grams) of rats fed diets varying in two components—level of protein and source of protein. The model is

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \varepsilon_{ijk} \quad i = 1, 2; j = 1, 2, 3; k = 1, 2, \dots, 10$$

where

$$\sum_{i=1}^2 \alpha_i = 0; \sum_{j=1}^3 \beta_j = 0; \sum_{i=1}^2 \gamma_{ij} = 0 \text{ for } j = 1, 2, 3; \text{ and } \sum_{j=1}^3 \gamma_{ij} = 0 \text{ for } i = 1, 2$$

Here, the IDUMMY = 0 option is used.

```
USE IMSL_LIBRARIES
INTEGER IDEP, LDB, LDR, LDSCPE, LDX, LINDEF, MAXB, MAXCL, &
        NCLVAR, NCOL, NEF, NROW
```

```

PARAMETER (IDEP=1, LINDEF=4, MAXB=12, MAXCL=5, NCLVAR=2, &
           NCOL=3, NEF=3, NROW=60, LDB=MAXB, LDR=MAXB, &
           LDSCPE=IDEP, LDX=NROW)
!
INTEGER   I, IDUMMY, INDCL(NCLVAR), INDDP(IDEP), &
           INDEF(LINDEF), INTCEP, IRANK, IRBEF(NEF+1), J, &
           NCLVAL(NCLVAR), NCOEF, NOUT, NRMISS, NVEF(NEF)
REAL      B(LDB,IDEP), CLVAL(MAXCL), D(MAXB), DFE, &
           R(LDR,MAXB), SCPE(LDSCPE,IDEP), X(LDX,NCOL), &
           XMAX(MAXB), XMIN(MAXB)
CHARACTER CLABEL(MAXB+1)*7, RLABEL(1)*4
!
DATA X/73.0, 102.0, 118.0, 104.0, 81.0, 107.0, 100.0, 87.0, &
      117.0, 111.0, 98.0, 74.0, 56.0, 111.0, 95.0, 88.0, 82.0, &
      77.0, 86.0, 92.0, 94.0, 79.0, 96.0, 98.0, 102.0, 102.0, &
      108.0, 91.0, 120.0, 105.0, 90.0, 76.0, 90.0, 64.0, 86.0, &
      51.0, 72.0, 90.0, 95.0, 78.0, 107.0, 95.0, 97.0, 80.0, &
      98.0, 74.0, 74.0, 67.0, 89.0, 58.0, 49.0, 82.0, 73.0, 86.0, &
      81.0, 97.0, 106.0, 70.0, 61.0, 82.0, 30*1.0, 30*2.0, &
      10*1.0, 10*2.0, 10*3.0, 10*1.0, 10*2.0, 10*3.0/
DATA INDCL/2, 3/, NVEF/1, 1, 2/, INDEF/2, 3, 2, 3/, INDDP/1/
DATA CLABEL/' ', 'MU', 'ALPHA1', 'ALPHA2', 'BETA1', 'BETA2', &
           'BETA3', 'GAMMA11', 'GAMMA12', 'GAMMA13', 'GAMMA21', &
           'GAMMA22', 'GAMMA23'/
DATA RLABEL/'NONE'/
!
IDUMMY = 0
CALL RGLM (X, INDCL, NVEF, INDEF, IDEP, INDDP, MAXCL, B, &
          IDUMMY=IDUMMY, NCLVAR=NCLVAR, NCLVAL=NCLVAL, &
          CLVAL=CLVAL, IRBEF=IRBEF, R=R, D=D, IRANK=IRANK, &
          DFE=DFE, SCPE=SCPE, NRMISS=NRMISS, XMIN=XMIN, &
          XMAX=XMAX)
!
CALL UMACH (2, NOUT)
WRITE (NOUT,*) 'NRMISS = ', NRMISS
WRITE (NOUT,*) 'IRANK = ', IRANK, ' DFE = ', DFE, ' ' // &
           ' SCPE(1,1) = ', SCPE(1,1)
J = 0
DO 10 I=1, NCLVAR
   CALL WRRRN ('Class Values', CLVAL((J+1):), 1, NCLVAL(I), 1)
   J = J + NCLVAL(I)
10 CONTINUE
NCOEF = IRBEF(NEF+1) - 1
CALL WRRRN ('XMIN', XMIN, 1, NCOEF, 1)
CALL WRRRN ('XMAX', XMAX, 1, NCOEF, 1)
CALL WRIRN ('IRBEF', IRBEF, 1, NEF+1, 1)
CALL WRRRN ('R-MATRIX', R, NRA=NCOEF, NCA=NCOEF, ITRING=1)
CALL WRRRL ('B', B, RLABEL, CLABEL, 1, NCOEF, 1, FMT='(2W10.4)')
!
END

```

Output

```

NRMISS = 0
IRANK = 12 DFE = 54.0000 SCPE(1,1) = 11586.0

```

Class Values

1	2
1.000	2.000

Class Values

1	2	3
1.000	2.000	3.000

XMIN

1	2	3	4	5	6	7	8	9	10
1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
11	12								
0.000	0.000								

XMAX

1	2	3	4	5	6	7	8	9	10
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
11	12								
1.000	1.000								

IRBEF

1	2	3	4
2	4	7	13

R-MATRIX

	1	2	3	4	5	6	7	8	9
1	7.746	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2		-1.000	-1.000	0.000	0.000	0.000	0.000	0.000	0.000
3			7.746	0.000	0.000	0.000	0.000	0.000	0.000
4				-1.000	-1.000	-1.000	0.000	0.000	0.000
5					6.325	3.162	0.000	0.000	0.000
6						5.477	0.000	0.000	0.000
7							-1.000	0.000	0.000
8								-1.000	0.000
9									-1.000

	10	11	12
1	0.000	0.000	0.000
2	0.000	0.000	0.000
3	0.000	0.000	0.000
4	0.000	0.000	0.000
5	0.000	0.000	0.000
6	0.000	0.000	0.000
7	-1.000	0.000	0.000
8	0.000	-1.000	0.000
9	0.000	0.000	-1.000
10	-1.000	-1.000	-1.000
11		6.325	3.162
12			5.477

B

MU	ALPHA1	ALPHA2	BETA1	BETA2	BETA3
87.87	7.267	-7.267	1.733	-2.967	1.233

GAMMA11	GAMMA12	GAMMA13	GAMMA21	GAMMA22	GAMMA23
3.133	-6.267	3.133	-3.133	6.267	-3.133

RLEQU

Fits a multivariate linear regression model with linear equality restrictions $HB = G$ imposed on the regression parameters given results from routine `RGIVN` (page 117) after `IDO = 1` and `IDO = 2` and prior to `IDO = 3`.

Required Arguments

- H** — `NH` by `NCOEF` matrix with the i -th row specifying a linear combination of the regression parameters for the i -th row in the restriction $HB = G$. (Input)
- B** — `NCOEF` by `NDEP` matrix containing on return from the final invocation of this routine a least-squares solution for the regression coefficients in the restricted model. (Input/Output)
 Invocation of `RLEQU` with `INVOKE = 0` and `1` requires as input the `B` matrix from `RGIVN` (page 117) after `RGIVN`'s invocation with `IDO = 1` and `IDO = 2` and prior to `IDO = 3` with `NROW = 0`. After the wrap-up computations are computed by `RLEQU`, `B` contains a least-squares solution for the regression coefficients in the restricted model.
- R** — `NCOEF` by `NCOEF` upper triangular matrix containing, on return from the final invocation of this routine, the R matrix from the restricted regression fit. (Input/Output)
 Invocation of `RLEQU` with `INVOKE = 0` and `1` requires as input the R matrix from `RGIVN` after `RGIVN`'s invocation with `IDO = 1` and `IDO = 2` and prior to `IDO = 3` with `NROW = 0`. After the wrap-up computations are computed by `RLEQU`, R contains the R matrix from the restricted regression fit. Elements to the right of a diagonal element of R (that is zero) are also zero. A zero row in R indicates a nonfull rank model. Each row of R corresponding to a restriction has a corresponding diagonal element that is negative. Each remaining row of R has a corresponding diagonal element that is positive.
- D** — Vector of length `NCOEF` containing scale factors associated with the fast Givens transformations. (Input/Output)
 Invocation of `RLEQU` with `INVOKE = 0` and `1` requires as input the `D` from `RGIVN` after `RGIVN`'s invocation with `IDO = 1` and `IDO = 2` and prior to `IDO = 3` with `NROW = 0`. After the wrap-up computations are computed by `RLEQU`, `D` contains all its elements set to 1.0.
- DFE** — Degrees of freedom for error for the restricted model on return from the final invocation of this routine. (Input/Output)
 Prior to the final invocation of this routine, `DFE` contains the sum of the frequencies. Invocation of `RLEQU` with `INVOKE = 0` and `1` requires as input the `DFE` from `RGIVN`

after *RGIVN*'s invocation with *IDO* = 1 and *IDO* = 2 and prior to *IDO* = 3 with *NROW* = 0.

SCPE — *NDEP* by *NDEP* matrix containing error (residual) sums of squares and crossproducts for the restricted model. (Input/Output)

SCPE(*M*, *N*) is the current sum of crossproducts of residuals for the *M*-th and *N*-th dependent variables. Invocation of *RLEQU* with *INVOKE* = 0 and 1 requires as input the *SCPE* matrix from *RGIVN* after *RGIVN*'s invocation with *IDO* = 1 and *IDO* = 2 and prior to *IDO* = 3 with *NROW* = 0.

Optional Arguments

INVOKE — Invocation option. (Input)

Default: *INVOKE* = 0.

INVOKE **Action**

- | | |
|---|---|
| 0 | This is the only invocation of <i>RLEQU</i> . All the restrictions are input at once. |
| 1 | This is the first invocation, and additional calls to <i>RLEQU</i> will be made. Initialization and updating for the restrictions $HB = G$ are performed. |
| 2 | This is an intermediate invocation of <i>RLEQU</i> , and updating for the restrictions $HB = G$ is performed. |
| 3 | This is the final invocation of this routine. Updating for the restrictions $HB = G$ is performed, and wrap-up computations are performed. |

NH — Number of rows in the restriction $HB = G$. (Input)

Default: *NH* = size (*H*,1).

NCOEF — Number of coefficients in the regression equation for each dependent variable.

(Input)

Default: *NCOEF* = size (*H*,2).

LDH — Leading dimension of *H* exactly as specified in the dimension statement of the calling program. (Input)

Default: *LDH* = size (*H*,1).

IG — Option for *G* matrix. (Input)

Default: *IG* = 0.

IG **Restrictions**

- | | |
|---|----------|
| 0 | $HB = 0$ |
| 1 | $HB = G$ |

NDEP — Number of dependent (response) variables. (Input)
Default: NDEP = size (B,2).

G — NH by NDEP matrix containing the right-hand side of the restriction
 $HB = G$. (Input, if IG = 1)
If IG = 0, G is not referenced and can be a 1 by 1 array.
Default: G is a 1 by 1 array.

LDG — Leading dimension of G exactly as specified in the dimension statement in the calling program. (Input)
Default: LDG = size(G, 1).

TOL — Tolerance used in determining linear dependence. (Input)
For RLEQU, TOL = 100.0 * AMACH(4) is a common choice. See the documentation for IMSL routines AMACH (Reference Material).
Default: TOL = 1.e-5 for single precision and 2.d-14 for double precision.

LDB — Leading dimension B exactly as specified in the dimension statement in the calling program. (Input)
Default: LDB = size (B,1).

LDR — Leading dimension of R exactly as specified in the dimension statement in the calling program. (Input)
Default: LDR = size (R,1).

IRANKR — Rank of matrix R. (Output, if INVOKE = 0 or 3)

LDSCPE — Leading dimension of SCPE exactly as specified in the dimension statement in the calling program. (Input)
Default: LDSCPE = size (SCPE,1).

IRANKH — Rank of matrix H. (Output)

FORTRAN 90 Interface

Generic: CALL RLEQU (H, B, R, D, DFE, SCPE[,...])

Specific: The specific interface names are S_RLEQU and D_RLEQU.

FORTRAN 77 Interface

Single: CALL RLEQU (INVOKE, NH, NCOEF, H, LDH, IG, NDEP, G, LDG, TOL, B, LDB, R, LDR, D, IRANKR, DFE, SCPE, LDSCPE, IRANKH)

Double: The double precision name is DRLEQU.

Example 1

A grafted polynomial (spline function) is fit to data discussed by Fuller (1976, pages 396–398). The data set contains the response variable y measuring the annual wheat yield (in bushels per acre) for the years 1908 through 1971. In order to fit the trend, Fuller fits a function that is constant for the first 25 years, increases at a quadratic rate until 1961, and is linear for the last 10 years. This trend is represented by the function $f(t)$ where

$$f(t) = \begin{cases} \beta_1 & \text{if } 1 \leq t \leq 25 \\ \beta_2 + \beta_3 t + \beta_4 t^2 & \text{if } 25 \leq t \leq 54 \\ \beta_5 + \beta_6 t & \text{if } 54 \leq t \leq 64 \end{cases}$$

where $t = 1$ for 1908.

In order to fit a smooth function to the data, we require both continuity and differentiability. This imposes four restrictions on the coefficients given as follows:

1. $\beta_1 - \beta_2 - 25\beta_3 - 25^2\beta_4 = 0$
2. $\beta_2 + 54\beta_3 + 54^2\beta_4 - \beta_5 - 54\beta_6 = 0$
3. $\beta_3 + 50\beta_4 = 0$
4. $\beta_3 + 108\beta_4 - \beta_6 = 0$

The example program first calls routine `RGIVN` (page 117) with `IDO = 1`, which specifies that initialization and updating for the data are performed and wrap-up computations are not performed. This intermediate output from `RGIVN` along with the restrictions is the input to `RLEQU`.

```
USE IMSL_LIBRARIES
INTEGER IDEP, LDB, LDG, LDH, LDR, LDSCPE, LDX, NCOEF, NH, &
NOBS, NVAR
PARAMETER (IDEP=1, LDG=1, NCOEF=6, NH=4, NOBS=64, NVAR=7, &
LDB=NCOEF, LDH=NH, LDR=NCOEF, LDSCPE=IDEP, LDX=NOBS)
!
INTEGER I, IDO, IG, INDDEP(IDEP), INDIND(NCOEF), INTCEP, &
IRANK, IRANKH, IRANKR, ICEN, NOUT, NRMISS
REAL B(LDB, IDEP), D(NCOEF), DFE, G(LDG, IDEP), &
H(LDH, NCOEF), R(LDR, NCOEF), SCPE(LDSCPE, IDEP), &
X(LDX, NVAR), XMAX(NCOEF), XMIN(NCOEF)
CHARACTER*4 RLABEL(1), CLABEL(1)
!
DATA INDIND/1, 2, 3, 4, 5, 6/, INDDEP/7/
DATA X/384*0.0, 14.3, 15.5, 13.7, 12.4, 15.1, 14.4, 16.1, 16.7, &
11.9, 13.2, 14.8, 12.9, 13.5, 12.7, 13.8, 13.3, 16.0, 12.8, &
14.7, 14.7, 15.4, 13.0, 14.2, 16.3, 13.1, 11.2, 12.1, 12.2, &
12.8, 13.6, 13.3, 14.1, 15.3, 16.8, 19.5, 16.4, 17.7, 17.0, &
17.2, 18.2, 17.9, 14.5, 16.5, 16.0, 18.4, 17.3, 18.1, 19.8, &
20.2, 21.8, 27.5, 21.6, 26.1, 23.9, 25.0, 25.2, 25.8, 26.5, &
26.3, 25.9, 28.4, 30.6, 31.0, 33.9/
DATA (H(1, J), J=1, NCOEF)/1, -1, -25, -625, 0, 0/
```

```

DATA (H(2,J),J=1,NCOEF)/0, 1, 54, 2916, -1, -54/
DATA (H(3,J),J=1,NCOEF)/0, 0, 1, 50, 0, 0/
DATA (H(4,J),J=1,NCOEF)/0, 0, 1, 108, 0, -1/
!
DATA RLABEL/'NONE'//,CLABEL/'NONE'//
!
DO 10 I=1, NOBS
  IF (I .LE. 25) THEN
!                                     Constant function.
      X(I,1) = 1.0
  ELSE IF (I.GT.25 .AND. I.LE.54) THEN
!                                     Quadratic function.
      X(I,2) = 1.0
      X(I,3) = I
      X(I,4) = I**2
  ELSE IF (I .GT. 54) THEN
!                                     Linear function.
      X(I,5) = 1.0
      X(I,6) = I
  END IF
10 CONTINUE
IDO = 1
INTCEP = 0
ICEN = 0
CALL RGIVN (X, NCOEF, INDIND, IDEP, INDDEP, B, IDO=IDO, &
            INTCEP=INTCEP, ICEN=ICEN, R=R, D=D, DFE=DFE, SCPE=SCPE)
CALL RLEQU (H, B, R, D, DFE, SCPE, IRANKR=IRANKR, IRANKH=IRANKH)
CALL UMACH (2, NOUT)
WRITE (NOUT,*) 'IRANKR = ', IRANKR, ' IRANKH = ', IRANKH
WRITE (NOUT,*) 'DFE = ', DFE, ' SCPE(1,1) = ', SCPE(1,1)
CALL WRRRL ('%/B', B, RLABEL, CLABEL, 1, NCOEF, 1, FMT='(2W10.4)')
CALL WRRRL ('%/R', R, RLABEL, CLABEL, ITRING=1, FMT='(2W10.4)')
END

```

Output

```

IRANKR = 6 IRANKH = 4
DFE = 62.0000 SCPE(1,1) = 172.559

```

B					
13.99	21.58	-0.6068	0.01214	-13.81	0.7039

R					
-1	1	25	625	0.	0.0
	-1	-54	-2916	1.	54.0
		-1	-50	0.	0.0
			-58	0.	1.0
				8.	359.4
					59.4

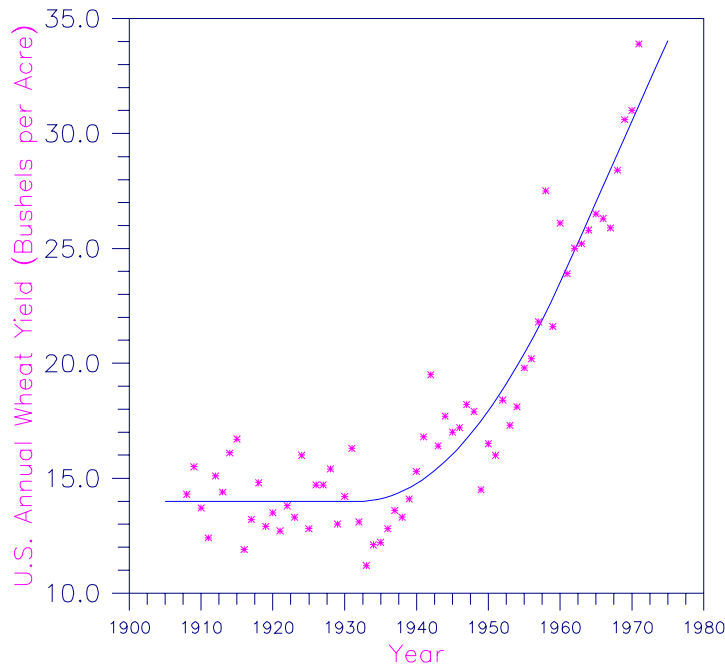


Figure 2-5 Annual U.S. Wheat Yield and a Grafted Polynomial Fit

Comments

1. Workspace may be explicitly provided, if desired, by use of R2REQ/DR2EQ. The reference is:

```
CALL R2REQ (INVOKE, NH, NCOEF, H, LDH, IG, NDEP, G, LDG, TOL,
B, LDB, R, LDR, D, IRANKR, DFE, SCPE, LDSCPE, IRANKH, WK)
```

The additional argument is:

WK — Work vector of length $NCOEF + NDEP$.

2. Informational error
Type Code
3 1 The restrictions are inconsistent.
3. The results of routine RGLM (page 128) can be used as input to RLEQU in place of the results of routine RGIVN (page 117).

Description

Routine RLEQU requires the output from routine RGIVN (page 117) after RGIVN has been invoked with $IDO = 1$ and $IDO = 2$ and prior to $IDO = 3$ with $NROW = 0$. Similarly, RLEQU can use results from IMSL routine RGLM (page 128).

The routine `RLEQU` is designed so that you can partition a large number of restrictions, as might arise in classification models, into several groups of restrictions (each requiring less space) and make multiple calls to `RLEQU` (with `INVOKE = 1, 2, 2, ..., 3`). Alternatively, one invocation of `RLEQU` (with `INVOKE = 0`) can be made with all the restrictions contained in `H` and `G`.

After the wrap-up computations are performed by `RLEQU`, routines `RSTAT` (page 152) and `RCASE` (page 204) can be used to compute and print summary statistics and case statistics related to the fitted regression.

Routine `RGIVN` (or `RGLM`) together with routine `RLEQU` compute estimates of the regression coefficients in a multivariate general linear model $Y = XB + E$ subject to $HB = G$. Here, Y is the $n \times q$ matrix of responses, X is the $n \times p$ matrix of regressors, B is the $p \times q$ matrix of regression coefficients, and E is the $n \times q$ matrix of errors whose q -dimensional rows are identically and independently distributed multivariate normal with mean vector 0 and variance-covariance matrix Σ . The restriction is specified by the $h \times p$ matrix H and the $h \times q$ matrix G .

Previously, algorithms for solving the restricted least-squares problem were based on solving the following equations (Rao, 1973, page 232):

$$\begin{aligned} X^T X \hat{B} + H^T \Lambda &= X^T Y \\ H \hat{B} &= G \end{aligned}$$

Routine `RLEQU` is based on an orthogonal reduction of X to upper triangular form. Fast Givens transformations with modifications described by Stirling (1981) for incorporating restrictions are used. This method has two main advantages: (1) the loss of accuracy resulting from forming $X^T X$ and $X^T Y$ is avoided, and (2) restrictions can be conveniently added so as to take advantage of the previous computations performed.

The method conceptually treats restrictions as observations with zero error variance. Fast Givens transformations as described by Golub and Van Loan (1983, pages 156–162) are used. The modification to the matrix R from the unrestricted fit to form a modified

$$\tilde{R}$$

for the restricted fit is as follows:

1. If the leading nonzero element of the first restriction is small (as determined by `TOL` times a computed scale factor), the element is set to zero.
2. Let i be the index of the leading nonzero element in the modified first restriction. Replace row i of R by the restriction. Flag the i -th row as a restriction. Use the restriction to reduce the first nonzero element of the row that was removed from R to zero. Incorporate the row that has been reduced by the restriction into the remaining rows of R as if it were new data.
3. Add additional restrictions into R by using Gaussian elimination, with the rows in R corresponding to restrictions, to reduce the restriction to a form so that it can replace a row of R corresponding to data and preserve the upper triangular structure of R . While performing the Gaussian elimination, set small nonzero elements (as determined by `TOL` times a computed scale factor) of the reduced restriction to zero, so that errors from in exact computer arithmetic are not incorporated as a new restriction. Flag the

row as a restriction. Use the restriction to reduce the first nonzero element of the row that was removed from R to zero. Incorporate the row that has been reduced by the restriction into the remaining rows of R as if it were new data.

4. After all the data and restrictions are incorporated, the i -th row of R (where i ranges over each row of R corresponding to a linearly independent constraint) is used to zero out elements of R in the i -th column of the previous rows of R that correspond to data. Although this step is not required to get a least-squares solution, Sallas (1988) recommends this step so that the rows and columns of

$$\tilde{R}$$

corresponding to data form the R matrix for the reduced model that arises from expressing some regression parameters, β_i , in terms of other regression parameters, $\beta_j (j > i)$.

Linear dependence of the regressors in the reduced model is then checked as part of the wrap-up computations, using the rows and columns of R corresponding to the reduced model. The check is complicated somewhat by the fact that a regressor could become zero in the reduced model, but because of the finite precision of computer arithmetic, the regressor is not exactly zero. Let d_i equal the i -th diagonal element of $X^T X$, and let

$$\tilde{d}_i$$

equal the corresponding diagonal from the crossproducts matrix for the reduced model. Linear dependence of regressors in the reduced model is declared if

$$\sqrt{1 - R_{i,2,\dots,i-1}^2}$$

is less than or equal to TOL or if

$$\sqrt{(1 - R_{i,2,\dots,i-1}^2) \tilde{d}_i / d_i}$$

is less than or equal to TOL . (The last check is designed to detect a zero regressor in the reduced model.) Here,

$$R_{i,2,\dots,i-1}^2$$

is the square of the “multiple correlation” coefficient of the i -th regressor in the reduced model with the first $i - 1$ regressors in the reduced model. The “multiple correlation” coefficient is computed using the regressors in the reduced model and adjusted for the mean only if the incorporated restrictions have that effect.

When a linear dependence is declared, R is changed so as to reflect the deletion of the i -th regressor from the model. On completion of the wrap-up computations, the rows of R can be partitioned into three classes according to the sign of the corresponding diagonal element:

1. A positive diagonal element means the row/column corresponds to data for regressors in the reduced model.

2. A negative diagonal element means the row corresponds to a linearly independent restriction imposed on the regression parameters by $HB = G$.
3. A zero diagonal element means a linear dependence in the reduced model was declared. The regression coefficients in the corresponding row of

$$\hat{B}$$

are set to zero. This represents an arbitrary restriction that is imposed to obtain a solution for the regression coefficients. The elements of the corresponding row of R are also set to zero.

Redundant restrictions on the regression parameters are frequently specified in general linear models. Routine `RLEQU` permits redundant restrictions and returns the rank of H . An informational error is issued if inconsistent restrictions are detected.

Additional Examples

Example 2

A fit to unbalanced data for a two-way classification model is computed. The model is

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \varepsilon_{ijk} \quad i = 1, 2; j = 1, 2; k = 1, 2, \dots, n_{ij}$$

where the α_i 's and β_j 's are the row and column effects, respectively, and γ_{ij} 's are the interaction effects. The responses y_{ijk} are given in the cells of the following 2×2 table:

17, 14, 11	13, 12
12, 14, 15, 14, 12	13, 14

The following restrictions can be imposed on the regression parameters in order to compute a cell-means fit to the responses:

1. $5\alpha_1 + 7\alpha_2 = 0$
2. $8\beta_1 + 4\beta_2 = 0$
3. $3\alpha_1 + 5\alpha_2 + 3\gamma_{11} + 5\gamma_{21} = 0$
4. $2\alpha_1 + 2\alpha_2 + 2\gamma_{12} + 2\gamma_{22} = 0$
5. $3\beta_1 + 2\beta_2 + 3\gamma_{11} + 2\gamma_{21} = 0$
6. $5\beta_1 + 2\beta_2 + 5\gamma_{12} + 2\gamma_{22} = 0$

The example program first calls IMSL routine `RGLM` (page 128) with `IDO = 1`, which specifies that initialization and updating for the data are performed and wrap-up computations are not performed. This intermediate output from `RGLM` along with the restrictions is the input to `RLEQU`.

A cell-means fit to the data could also be obtained without using RLEQU and using IDO = 0 in the call to RGLM in this example. Although the fitted y_{ijk} would be the same, the coefficient estimates and their interpretations would be different.

```

USE IMSL_LIBRARIES
INTEGER IDEP, INTCEP, LDB, LDG, LDH, LDR, LDSCPE, LDX, MAXCL, &
NCLVAR, NCOEF, NEF, NH, NOBS, NVAR
PARAMETER (IDEP=1, INTCEP=1, LDG=1, LDH=6, MAXCL=4, NCLVAR=2, &
NCOEF=9, NEF=3, NH=6, NOBS=12, NVAR=3, LDB=NCOEF, &
LDR=NCOEF, LDSCPE=IDEP, LDX=NOBS)
!
INTEGER IDO, INDCL(NCLVAR), INDDEP(1), INDEF(4), &
IRANK, IRANKH, IRANKR, IRBEF(NEF+1), ICEN, &
NCLVAL(NCLVAR), NOUT, NRMISS, NVEF(NEF)
REAL B(LDB, IDEP), CLVAL(MAXCL), D(NCOEF), DFE, &
G(LDG, IDEP), H(LDH, NCOEF), R(LDR, NCOEF), &
SCPE(LDSCPE, IDEP), X(LDX, NVAR), XMAX(NCOEF), &
XMIN(NCOEF)
CHARACTER CLABEL(10)*7, RLABEL(1)*4
!
DATA INDCL/1, 2/, NVEF/1, 1, 2/, INDEF/1, 2, 1, 2/, INDDEP/3/
DATA CLABEL/' ', 'MU', 'ALPHA1', 'ALPHA2', 'BETA1', 'BETA2', &
'GAMMA11', 'GAMMA12', 'GAMMA21', 'GAMMA22'/
DATA (X(1, J), J=1, NVAR) /1, 1, 17/
DATA (X(2, J), J=1, NVAR) /1, 1, 14/
DATA (X(3, J), J=1, NVAR) /1, 1, 11/
DATA (X(4, J), J=1, NVAR) /1, 2, 13/
DATA (X(5, J), J=1, NVAR) /1, 2, 12/
DATA (X(6, J), J=1, NVAR) /2, 1, 12/
DATA (X(7, J), J=1, NVAR) /2, 1, 14/
DATA (X(8, J), J=1, NVAR) /2, 1, 15/
DATA (X(9, J), J=1, NVAR) /2, 1, 14/
DATA (X(10, J), J=1, NVAR) /2, 1, 12/
DATA (X(11, J), J=1, NVAR) /2, 2, 13/
DATA (X(12, J), J=1, NVAR) /2, 2, 14/
DATA (H(1, J), J=1, NCOEF) /0, 5, 7, 0, 0, 0, 0, 0, 0/
DATA (H(2, J), J=1, NCOEF) /0, 0, 0, 8, 4, 0, 0, 0, 0/
DATA (H(3, J), J=1, NCOEF) /0, 3, 5, 0, 0, 3, 0, 5, 0/
DATA (H(4, J), J=1, NCOEF) /0, 2, 2, 0, 0, 0, 2, 0, 2/
DATA (H(5, J), J=1, NCOEF) /0, 0, 0, 3, 2, 3, 2, 0, 0/
DATA (H(6, J), J=1, NCOEF) /0, 0, 0, 5, 2, 0, 0, 5, 2/
!
IDO = 1
ICEN = 0
CALL RGLM (IDO=IDO, INDCL, NVEF, INDEF, IDEP, INDDEP, MAXCL, B, &
ICEN=ICEN, R=R, D=D, DFE=DFE, SCPE=SCPE)
CALL RLEQU (H, B, R, D, DFE, SCPE, IRANKR=IRANKR, &
IRANKH=IRANKH)
CALL UMACH (2, NOUT)
WRITE (NOUT, *) 'IRANKR = ', IRANKR, ' IRANKH = ', IRANKH
WRITE (NOUT, *) 'DFE = ', DFE, ' SCPE(1,1) = ', SCPE(1,1)
RLABEL(1) = 'NONE'
CALL WRRRL ('B', B, RLABEL, CLABEL, 1, NCOEF, 1, FMT='(F7.2)')
CALL WRRRN ('R', R, ITRING=1)
END

```

Output

IRANKR = 9 IRANKH = 5
DFE = 8.00000 SCPE(1,1) = 26.2000

B

MU	ALPHA1	ALPHA2	BETA1	BETA2	GAMMA11	GAMMA12	GAMMA21
13.42	-0.02	0.01	0.21	-0.42	0.39	-0.48	-0.24

GAMMA22
0.49

R

	1	2	3	4	5	6	7	8	9
1	3.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2		-5.00	-7.00	0.00	0.00	0.00	0.00	0.00	0.00
3			-0.80	0.00	0.00	-3.00	0.00	-5.00	0.00
4				-8.00	-4.00	0.00	0.00	0.00	0.00
5					-0.50	-3.00	-2.00	0.00	0.00
6						-3.00	-2.00	-5.00	-2.00
7							10.41	3.20	11.37
8								24.56	9.65
9									2.45

RSTAT

Computes statistics related to a regression fit given the coefficient estimates

$$\hat{\beta}$$

and the R matrix.

Required Arguments

IRBEF — Index vector of length $|\text{IEF}| + 1$. (Input, if IEF is positive.)

For $i = 1, 2, \dots, |\text{IEF}|$, element numbers $\text{IRBEF}(i), \text{IRBEF}(i) + 1, \dots, \text{IRBEF}(i + 1) - 1$, of B correspond to the i -th effect.

B — Vector of length NCOEF containing a least-squares solution

$$\hat{\beta}$$

for the regression coefficients. (Input)

Here, if $\text{IEF} > 0$, then $\text{NCOEF} = \text{IRBEF}(\text{IEF} + 1) - 1$; and if $\text{IEF} \leq 0$, then

$\text{NCOEF} = \text{INTCEP} - \text{IEF}$. If $\text{INTCEP} = 1$, then $B(1)$ must be the estimated intercept.

R — NCOEF by NCOEF upper triangular matrix containing the R matrix. (Input)

The R matrix can come from a regression fit based on a QR decomposition of the matrix of regressors or based on a Cholesky factorization $R^T R$ of the matrix of sums of squares and crossproducts of the regressors. Elements to the right of a diagonal element

of R that is zero must also be zero. A zero row indicates a nonfull rank model. For an R matrix that comes from a regression fit with linear equality restrictions on the parameters, each row of R corresponding to a restriction must have a corresponding diagonal element that is negative. The remaining rows of R must have positive diagonal elements. Only the upper triangle of R is referenced.

DFE — Degrees of freedom for error. (Input)

SSE — Sum of squares for error. (Input)

AOV — Vector of length 15 containing statistics relating to the analysis of variance. (Output)

I	AOV(I)
1	Degrees of freedom for regression
2	Degrees of freedom for error
3	Total degrees of freedom
4	Sum of squares for regression
5	Sum of squares for error
6	Total sum of squares
7	Regression mean square
8	Error mean square
9	F -statistic
10	p -value
11	R^2 (in percent)
12	Adjusted R^2 (in percent)
13	Estimated standard deviation of the model error
14	Mean of the response (dependent) variable
15	Coefficient of variation (in percent)

If `INTCEP = 1`, the regression and total are corrected for the mean. If `INTCEP = 0`, the regression and total are not corrected for the mean, and `AOV(14)` and `AOV(15)` are set to NaN (not a number).

SQSS — $|IEF|$ by 4 matrix containing in columns 1 through 4 the sequential degrees of freedom, sum of squares, F -statistic, and p -value. (Output)
Each row corresponds to an effect. If $IEF = 0$, $SQSS$ is not referenced and can be a vector of length one.

COEF — N_{COEF} by 5 matrix containing statistics relating to the regression coefficients. (Output)
Each row corresponds to a coefficient in the model. Row $INTCEP + I$ corresponds to the coefficient for the I -th independent variable. If $INTCEP = 1$, the first row corresponds to the intercept. The statistics in the columns are

Col. Description

- 1 Coefficient estimate.
- 2 Estimated standard error of the coefficient estimate.
- 3 t -statistic for the test that the coefficient is zero.
- 4 p -value for the two-sided t test.
- 5 Variance inflation factors. The square of the multiple correlation coefficient for the I -th regressor after all others can be obtained from $COEF(I, 5)$ by the formula $1.0 - 1.0/COEF(I, 5)$. If $INTCEP = 0$ or $INTCEP = 1$ and $I = 1$, the “multiple correlation coefficient” is not adjusted for the mean.

COVB — N_{COEF} by N_{COEF} matrix that is the estimated variance-covariance matrix of the estimated regression coefficients when R is nonsingular and is from an unrestricted regression fit. (Output)
See Comments for an explanation of $COVB$ when R is singular or R is from a restricted regression fit. If R is not needed, $COVB$ and R can share the same storage locations.

Optional Arguments

INTCEP — Intercept option. (Input)
Default: $INTCEP = 1$.

INTCEP Action

- 0 An intercept is not in the model.
- 1 An intercept is in the model.

IEF — Effect option. (Input)
Default: $IEF = 0$.
The absolute value of IEF is the number of effects (sources of variation) in the model excluding the error. The sign of IEF specifies the following options:

IEF Meaning

- < 0 Each effect corresponds to a single regressor (coefficient) in the model.
- > 0 Each effect corresponds to one or more regressors. The association between the effects and the regressors is given by elements of IRBEF.
- 0 There are no effects in the model. INTCEP must equal 1.

LDR — Leading dimension of R exactly as specified in the dimension statement in the calling program. (Input)
Default: LDR = size (R,1).

PRINT — Printing option. (Input)

Default: PRINT = 'N'.

PRINT is a character string indicating what is to be printed. The PRINT string is composed of one character print codes to control printing. These print codes are given as follows:

PRINT(I : I)	Printing that occurs
'A'	All
'N'	None
'1'	AOV
'2'	SQSS
'3'	COEF
'4'	COVB

The concatenated print codes 'A', 'N', '1', ..., '4' that comprise the PRINT string give the combination of statistics to be printed. Here are a few examples.

PRINT Printing Action

- 'A' All
- 'N' None
- '13' AOV and COEF
- '124' AOV, SQSS, and COVB

LDSQSS — Leading dimension of SQSS exactly as specified in the dimension statement in the calling program. (Input)

Default: LDSQSS = size (SQSS,1).

LDCOEF — Leading dimension of COEF exactly as specified in the dimension statement in the calling program. (Input)

Default: LDCOEF = size (COEF,1).

LDCOVB — Leading dimension of COVB exactly as specified in the dimension statement in the calling program. (Input)

Default: LDCOVB = size (COVB,1).

FORTRAN 90 Interface

Generic: CALL RSTAT (IRBEF, B, R, DFE, SSE, AOV, SQSS, COEF,
COVB [,...])

Specific: The specific interface names are S_RSTAT and D_RSTAT.

FORTRAN 77 Interface

Single: CALL RSTAT (INTCEP, IEF, IRBEF, B, R, LDR, DFE, SSE, PRINT,
AOV, SQSS, LDSQSS, COEF, LDCOEF, COVB, LDCOVB)

Double: The double precision name is DRSTAT.

Example 1

This example uses a data set discussed by Draper and Smith (1981, pages 629–630). This data set is put into the matrix *X* by routine GDATA (see Chapter 19, Utilities). There are 4 independent variables and 1 dependent variable. Routine RGIVN (page 117) is invoked to fit the regression model and RSTAT is invoked to compute summary statistics.

```
USE RSTAT_INT
USE GDATA_INT
USE RGIVN_INT
!
!                               SPECIFICATIONS FOR LOCAL VARIABLES
INTEGER    INTCEP, LDB, LDCOEF, LDCOVB, LDR, LDSCPE, LDSQSS, &
           LDX, NCOEF, NDEP, NDX, NIND
PARAMETER  (INTCEP=1, LDX=13, NDEP=1, NDX=5, NIND=4, &
           LDSCPE=NDEP, LDSQSS=NIND, NCOEF=INTCEP+NIND, &
           LDB=NCOEF, LDCOEF=NCOEF, LDCOVB=NCOEF, LDR=NCOEF)
!
INTEGER    IDEP, IDO, IEF, IFRQ, IIND, INDDEP(1), INDIND(1), &
           IRANK, IRBEF(1), IWT, NCOL, NRMISS, NROW
REAL       AOV(15), B(LDB,NDEP), COEF(LDCOEF,5), &
           COVB(LDCOVB,5), D(NCOEF), DFE, R(LDR,NCOEF), &
           SCPE(LDSCPE,NDEP), SQSS(LDSQSS,4), SSE, TOL, &
           X(LDX,NDX), XMAX(NCOEF), XMIN(NCOEF)
CHARACTER  PRINT*5
!
```

```

CALL GDATA (5, X, NROW, NCOL)
IIND = -NIND
IDEP = -NDEP

CALL RGIVN (X, IIND, INDIND, IDEP, INDDEP, B, R=R, DFE=DFE, &
           SCPE=SCPE)

PRINT = 'A'
IEF = -NIND
SSE = SCPE(1,1)
!
CALL RSTAT (IRBEF, B(:, 1), R, DFE, SSE, AOV, SQSS, COEF, COVB, &
           IEF=IEF, PRINT=PRINT)
!
END

```

Output

R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
98.238	97.356	2.446	95.42	2.563

```

* * * Analysis of Variance * * *

```

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Regression	4	2667.9	667.0	111.479	0.0000
Residual	8	47.9	6.0		
Corrected Total	12	2715.8			

```

* * * Sequential Statistics * * *

```

Indep. Variable	Degrees of Freedom	Sum of Squares	F-statistic	Prob. of Larger F
1	1	1450.1	242.368	0.0000
2	1	1207.8	201.870	0.0000
3	1	9.8	1.637	0.2366
4	1	0.2	0.041	0.8441

```

* * * Inference on Coefficients * * *

```

Coef.	Estimate	Standard Error	t-statistic	Prob. of Larger t	Variance Inflation
1	62.41	70.07	0.891	0.3991	10668.5
2	1.55	0.74	2.083	0.0708	38.5
3	0.51	0.72	0.705	0.5009	254.4
4	0.10	0.75	0.135	0.8959	46.9
5	-0.14	0.71	-0.203	0.8441	282.5

```

* * * Variance-Covariance Matrix for the Coefficient Estimates * * *

```

	1	2	3	4	5
1	4909.95	-50.51	-50.60	-51.66	-49.60
2		0.55	0.51	0.55	0.51
3			0.52	0.53	0.51
4				0.57	0.52
5					0.50

Comments

When R is nonsingular and comes from an unrestricted regression fit, COVB is the estimated variance-covariance matrix of the estimated regression coefficients, and

$\text{COVB} = (\text{SSE}/\text{DFE}) * (R^T R)^{-1}$. Otherwise, variances and covariances of estimable functions of the regression coefficients can be obtained using COVB , and $\text{COVB} = (\text{SSE}/\text{DFE}) * G D G^T$. Here, D is the diagonal matrix with diagonal elements equal to 0 if the corresponding rows of R are restrictions and with diagonal elements equal to one otherwise. Also, G is a particular generalized inverse of R . See the Algorithm section.

Description

Routine `RSTAT` computes summary statistics from a fitted general linear model. The model is $y = X\beta + \varepsilon$ where y is the $n \times 1$ vector of responses, X is the $n \times p$ matrix of regressors, β is the $p \times 1$ vector of regression coefficients, and ε is the $n \times 1$ vector of errors whose elements are each independently distributed with mean 0 and variance σ^2 . Routine `RGIVN` (page 117) or routine `RGLM` (page 128) can be used to compute the fit of the model. Next, `RSTAT` uses the results of this fit to compute summary statistics, including analysis of variance, sequential sum of squares, t tests, and estimated variance-covariance matrix of the estimated regression coefficients.

Some generalizations of the general linear model are allowed. If the i -th element of ε has variance σ^2/w_i and the weights w_i are used in the fit of the model, `RSTAT` produces summary statistics from the weighted least-squares fit. More generally, if the variance-covariance matrix of ε is $\sigma^2 V$, `RSTAT` can be used to produce summary statistics from the generalized least-squares fit. (Routine `RGIVN` can be used to perform a generalized least-squares fit, by regressing y^* on X^* where $y^* = (T^{-1})^T y$, $X^* = (T^{-1})^T X$ and T satisfies $T^T T = V$. Routines for computing y^* and X^* can be found in the IMSL MATH/LIBRARY.)

If the general linear model has the restriction $H\beta = g$ on the regression parameters, and this restriction is used in the fit of the model by routine `RLEQU` (page 142), `RSTAT` produces summary statistics from this restricted least-squares fit.

The sequential sum of squares for the i -th regression parameter is given by

$$(R\hat{\beta})_i^2$$

The regression sum of squares is given by the sum of the sequential sums of squares. If an intercept is in the model, the regression sum of squares is adjusted for the mean, i.e.,

$$(R\hat{\beta})_1^2$$

is not included in the sum.

The estimate of σ^2 is s^2 (stored in `AOV(8)`) that is computed as SSE/DFE .

If R is nonsingular, the estimated variance-covariance matrix of

$$\hat{\beta}$$

(stored in `COVB`) is computed by $s^2 R^{-1} (R^{-1})^T$.

If R is singular, corresponding to $\text{rank}(X) < p$, a generalized inverse is used. For a matrix G to be a g_i ($i = 1, 2, 3$, or 4) inverse of a matrix A , G must satisfy conditions j (for $j \leq i$) for the Moore-Penrose inverse but generally must fail conditions k (for $k > i$). The four conditions for G to be a Moore-Penrose inverse of A are as follows:

1. $AGA = A$
2. $GAG = G$
3. AG is symmetric
4. GA is symmetric

In the case where R is singular, the method for obtaining `COVB` follows the discussion of Maindonald (1984, pages 101–103). Let Z be the diagonal matrix with diagonal elements defined by

$$z_{ii} = \begin{cases} 1 & \text{if } r_{ii} \neq 0 \\ 0 & \text{if } r_{ii} = 0 \end{cases}$$

Let G be the solution to $RG = Z$ obtained by setting the i -th ($\{i : r_{ii} = 0\}$) row of G to zero. `COVB` is set to $s^2 GG^T$. (G is a g_3 inverse of R . For any g_3 inverse of R , represented by

$$R^{g_3}$$

the result

$$R^{g_3} R^{g_3 T}$$

is a symmetric g_2 inverse of $R^T R = X^T X$. See Sallas and Lioni [1988].)

Note that `COVB` can only be used to get variances and covariances of estimable functions of the regression coefficients, i.e., nonestimable functions (linear combinations of the regression coefficients not in the space spanned by the nonzero rows of R) must not be used. See, for example, Maindonald (1984, pages 166–168) for a discussion of estimable functions.

The estimated standard errors of the estimated regression coefficients (stored in column 2 of `COEF`) are computed as square roots of the corresponding diagonal entries in `COVB`.

For the case where an intercept is in the model, put

$$\bar{R}$$

equal to the matrix R with the first row and column deleted. Generally, the variance inflation factor (VIF) for the i -th regression coefficient is computed as the product of the i -th diagonal element of $R^T R$ and the i -th diagonal element of its computed inverse. If an intercept is in the model, the VIF for those coefficients not corresponding to the intercept uses the diagonal elements of

$$\bar{R}^T \bar{R}$$

(see Maindonald 1984, page 40).

The preceding discussion can be modified to include the restricted least-squares problem. The modification is based on the work of Stirling (1981). Let the matrix $D = \text{diag}(d_1, d_2, \dots, d_p)$ be a diagonal matrix with elements $d_i = 0$ if the i -th row of R corresponds to restriction. In the unrestricted case, D is simply the $p \times p$ identity matrix. The formula for COVB is s^2GDG^T . The formula for the sequential sum of squares for the i -th ($\{i : r_{ii} > 0\}$) regression parameter is given by

$$(DR\hat{\beta})_i^2$$

Sequential sums of squares for $\{i : r_{ii} \leq 0\}$ are set to zero.

For the restricted least-squares problem, the sequential and regression sums of squares correspond to those from a fitted reduced model obtained by first substituting the restriction $H\beta = g$ into the model. In general, the reduced model is not unique. Care must be taken to interpret the sequential sums of squares in the context of the particular reduced model indicated by the R matrix. If $g = 0$, any of the reduced models that could be computed from the restrictions will produce the same regression sum of squares. However, if $g \neq 0$, different reduced models resulting from the same restricted model can have different regressands, and hence, different total and regression sums of squares.

Additional Examples

Example 2

A one-way analysis of covariance model is fitted to the turkey data discussed by Draper and Smith (1981, pages 243–249). The response variable is turkey weight y (in pounds). Three groups of turkeys corresponding to the three states where they were reared are used. The age of a turkey (in weeks) is the covariate. The explanatory variables are age, group, and interaction. The model is

$$y_{ij} = \mu + \beta x_{ij} + \alpha_i + \beta_i x_{ij} + \varepsilon_{ij} \quad i = 1, 2, 3; j = 1, 2, \dots, n_i$$

where $\alpha_3 = 0$ and $\beta_3 = 0$. Routine `RGLM` (page 128) is used to fit the model with the option `IDUMMY = 2`. Then, `RSTAT` is used to compute summary statistics. The fitted model gives three separate lines with slopes 0.506, 0.470, and 0.445. The F test for interaction (the last effect) suggests omitting the interaction from the model and using a model with identical slopes for each group.

```

USE RSTAT_INT
USE RGLM_INT
!
                                SPECIFICATIONS FOR PARAMETERS
INTEGER      IDEP, IEF, INTCEP, LDB, LDcoef, LDCOVb, LDR, LDSCPE, &
              LDSQSS, LDX, MAXB, MAXCL, NCLVAR, NCOL, NROW
PARAMETER    (IDEP=1, IEF=3, INTCEP=1, LDX=13, MAXB=6, MAXCL=3, &
              NCLVAR=1, NCOL=3, NROW=13, LDB=MAXB, LDcoef=MAXB, &
              LDCOVb=MAXB, LDR=MAXB, LDSCPE=IDEP, LDSQSS=IEF)
!
INTEGER      IDO, IDUMMY, IFRQ, INDCL(NCLVAR), INDDEP(IDEP), &
              INDEF(4), IRANK, IRBEF(IEF+1), IWT, NCLVAL(NCLVAR), &
              NRMISS, NVEF(IEF)
REAL         AOV(15), B(LDB, IDEP), CLVAL(MAXCL), &

```

```

COEF(LDCOEF,5), COVB(LDCOVB,MAXB), D(MAXB), DFE, &
R(LDR,MAXB), SCPE(LDSCPE,IDEF), SQSS(LDSQSS,4), SSE, &
TOL, X(LDX,NCOL), XMAX(MAXB), XMIN(MAXB)
CHARACTER PRINT*1
!
DATA (X(1,J),J=1,3)/25, 13.8, 3/
DATA (X(2,J),J=1,3)/28, 13.3, 1/
DATA (X(3,J),J=1,3)/20, 8.9, 1/
DATA (X(4,J),J=1,3)/32, 15.1, 1/
DATA (X(5,J),J=1,3)/22, 10.4, 1/
DATA (X(6,J),J=1,3)/29, 13.1, 2/
DATA (X(7,J),J=1,3)/27, 12.4, 2/
DATA (X(8,J),J=1,3)/28, 13.2, 2/
DATA (X(9,J),J=1,3)/26, 11.8, 2/
DATA (X(10,J),J=1,3)/21, 11.5, 3/
DATA (X(11,J),J=1,3)/27, 14.2, 3/
DATA (X(12,J),J=1,3)/29, 15.4, 3/
DATA (X(13,J),J=1,3)/23, 13.1, 3/
DATA INDCL/3/, NVEF/1, 1, 2/, INDEF/1, 3, 1, 3/, INDDEP/2/
!
IDUMMY = 2
CALL RGLM (X, INDCL, NVEF, INDEF, IDEP, INDDEP, MAXCL, &
B, IDUMMY=IDUMMY, IRBEF=IRBEF, R=R, DFE=DFE, SCPE=SCPE)
!
SSE = SCPE(1,1)
PRINT = 'A'
CALL RSTAT (IRBEF, B(:,1), R, DFE, SSE, AOV, SQSS, COEF, COVB, &
IEF=IEF, PRINT=PRINT)
!
END

```

Output

R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
98.208	96.929	0.3176	12.78	2.484

```

* * * Analysis of Variance * * *

```

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Regression	5	38.71	7.742	76.744	0.0000
Residual	7	0.71	0.101		
Corrected Total	12	39.42			

```

* * * Sequential Statistics * * *

```

Effect	Degrees of Freedom	Sum of Squares	F-statistic	Prob. of Larger F
1	1	26.20	259.728	0.0000
2	2	12.40	61.477	0.0000
3	2	0.11	0.520	0.6156

```

* * * Inference on Coefficients * * *

```

Coef.	Estimate	Standard Error	t-statistic	Prob. of Larger t	Variance Inflation
1	2.475	1.264	1.959	0.0910	205.7
2	0.445	0.050	8.861	0.0000	3.8

3	-3.454	1.531	-2.257	0.0586	64.3
4	-2.775	4.109	-0.675	0.5211	463.4
5	0.061	0.060	1.013	0.3447	68.1
6	0.025	0.151	0.166	0.8729	472.3

* * * Variance-Covariance Matrix for the Coefficient Estimates * * *

	1	2	3	4	5
1	1.5965	-0.0631	-1.5965	-1.5965	0.0631
2		0.0025	0.0631	0.0631	-0.0025
3			2.3425	1.5965	-0.0913
4				16.8801	-0.0631
5					0.0036

	6
1	0.0631
2	-0.0025
3	-0.0631
4	-0.6179
5	0.0025
6	0.0227

Example 3

A two-way analysis-of-variance model is fitted to balanced data discussed by Snedecor and Cochran (1967, Table 12.5.1, page 347). The responses are the weight gains (in grams) of rats fed diets varying in two components—level of protein and source of protein. The model is

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \varepsilon_{ijk} \quad i = 1, 2; j = 1, 2, 3; k = 1, 2, \dots, 10$$

where

$$\sum_{i=1}^2 \alpha_i = 0; \sum_{j=1}^3 \beta_j = 0; \sum_{i=1}^2 \gamma_{ij} = 0 \text{ for } j = 1, 2, 3; \text{ and } \sum_{j=1}^3 \gamma_{ij} = 0 \text{ for } i = 1, 2$$

Routine RGLM (page 128) is used to fit the model with the IDUMMY = 0 option. Then, RSTAT is used to compute summary statistics.

```

USE RSTAT _INT
USE RGLM _INT
INTEGER   IDEP, IEF, LDB, LDCOEF, LDCOV, LDR, LDSCPE, LDSQSS, &
          LDX, LINDEF, MAXB, MAXCL, NCLVAR, NCOL, NEF, NROW
PARAMETER (IDEP=1, LINDEF=4, MAXB=12, MAXCL=5, NCLVAR=2, &
          NCOL=3, NEF=3, NROW=60, IEF=NEF, LDB=MAXB, &
          LDCOEF=MAXB, LDCOV=MAXB, LDR=MAXB, LDSCPE=IDEP, &
          LDSQSS=NEF, LDX=NROW)
!
INTEGER   IDO, IDUMMY, IFRQ, INDCL(NCLVAR), INDEP(IDEP), &
          INDEF(LINDEF), INTCEP, IRANK, IRBEF(NEF+1), IWT, &
          NCLVAL(NCLVAR), NRMISS, NVEF(NEF)
REAL      AOV(15), B(LDB, IDEP), CLVAL(MAXCL), &
          COEF(LDCOEF, 5), COVB(LDCOV, MAXB), D(MAXB), DFE, &
          R(LDR, MAXB), SCPE(LDSCPE, IDEP), SQSS(LDSQSS, 4), SSE, &
          TOL, X(LDX, NCOL), XMAX(MAXB), XMIN(MAXB)
CHARACTER PRINT*1
!
```



```

DATA X/73.0, 102.0, 118.0, 104.0, 81.0, 107.0, 100.0, 87.0, &
      117.0, 111.0, 98.0, 74.0, 56.0, 111.0, 95.0, 88.0, 82.0, &
      77.0, 86.0, 92.0, 94.0, 79.0, 96.0, 98.0, 102.0, 102.0, &
      108.0, 91.0, 120.0, 105.0, 90.0, 76.0, 90.0, 64.0, 86.0, &
      51.0, 72.0, 90.0, 95.0, 78.0, 107.0, 95.0, 97.0, 80.0, &
      98.0, 74.0, 74.0, 67.0, 89.0, 58.0, 49.0, 82.0, 73.0, 86.0, &
      81.0, 97.0, 106.0, 70.0, 61.0, 82.0, 30*1.0, 30*2.0, &
      10*1.0, 10*2.0, 10*3.0, 10*1.0, 10*2.0, 10*3.0/
DATA INDCL/2, 3/, NVEF/1, 1, 2/, INDEF/2, 3, 2, 3/, INDDEP/1/
!
IDUMMY = 0
CALL RGLM (X, INDCL, NVEF, INDEF, IDEP, INDDEP, MAXCL, B, &
          IDUMMY=IDUMMY, IRBEF=IRBEF, R=R, DFE=DFE, SCPE=SCPE)
!
SSE = SCPE(1,1)
PRINT = 'A'
CALL RSTAT (IRBEF, B(:,1), R, DFE, SSE, AOV, SQSS, COEF, &
          COVB, IEF=IEF, PRINT=PRINT)
!
END

```

Output

R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Coefficient of Mean Var. (percent)
28.477	21.854	14.65	87.87 16.67

```

* * * Analysis of Variance * * *

```

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Regression	5	4612.9	922.6	4.300	0.0023
Residual	54	11586.0	214.6		
Reduced Model Total	59	16198.9			

```

* * * Sequential Statistics * * *

```

Effect	Degrees of Freedom	Sum of Squares	F-statistic	Prob. of Larger F
1	1	3168.3	14.767	0.0003
2	2	266.5	0.621	0.5411
3	2	1178.1	2.746	0.0732

```

* * * Inference on Coefficients * * *

```

Coef.	Estimate	Standard Error	t-statistic	Prob. of Larger t	Variance Inflation
1	87.87	1.891	46.47	0.0000	1.000
2	7.27	1.891	3.84	0.0003	NaN
3	-7.27	1.891	-3.84	0.0003	1.000
4	1.73	2.674	0.65	0.5196	NaN
5	-2.97	2.674	-1.11	0.2722	1.333
6	1.23	2.674	0.46	0.6465	1.333
7	3.13	2.674	1.17	0.2465	NaN
8	-6.27	2.674	-2.34	0.0228	NaN
9	3.13	2.674	1.17	0.2465	NaN
10	-3.13	2.674	-1.17	0.2465	NaN
11	6.27	2.674	2.34	0.0228	1.333
12	-3.13	2.674	-1.17	0.2465	1.333

```

* * * Variance-Covariance Matrix for the Coefficient Estimates * * *
      1      2      3      4      5
1      3.57593      0.00000      0.00000      0.00000      0.00000
2      0.00000      3.57593     -3.57593      0.00000      0.00000
3      0.00000      0.00000      3.57593      0.00000      0.00000
4      0.00000      0.00000      0.00000      7.15185     -3.57592
5      0.00000      0.00000      0.00000      0.00000      7.15185

      6      7      8      9      10
1      0.00000      0.00000      0.00000      0.00000      0.00000
2      0.00000      0.00000      0.00000      0.00000      0.00000
3      0.00000      0.00000      0.00000      0.00000      0.00000
4     -3.57593      0.00000      0.00000      0.00000      0.00000
5     -3.57593      0.00000      0.00000      0.00000      0.00000
6      7.15185      0.00000      0.00000      0.00000      0.00000
7      0.00000      7.15185     -3.57592     -3.57593     -7.15185
8      0.00000      0.00000      7.15185     -3.57593      3.57592
9      0.00000      0.00000      0.00000      7.15185      3.57593
10     0.00000      0.00000      0.00000      0.00000      7.15185

      11     12
1      0.00000      0.00000
2      0.00000      0.00000
3      0.00000      0.00000
4      0.00000      0.00000
5      0.00000      0.00000
6      0.00000      0.00000
7      3.57592      3.57593
8     -7.15185      3.57593
9      3.57593     -7.15185
10     -3.57592     -3.57593
11      7.15185     -3.57593
12     0.00000      7.15185

```

RCOVB

Computes the estimated variance-covariance matrix of the estimated regression coefficients given the R matrix.

Required Arguments

R — NCOEF by NCOEF upper triangular matrix containing the R matrix. (Input)

The R matrix can come from a regression fit based on a QR decomposition of the matrix of regressors or based on a Cholesky factorization $R^T R$ of the matrix of sums of squares and crossproducts of the regressors. Elements to the right of a diagonal element of R that is zero must also be zero. A zero row indicates a nonfull rank model. For an R matrix that comes from a regression fit with linear equality restrictions on the parameters, each row of R corresponding to a restriction must have a corresponding diagonal element that is negative. The remaining rows of R must have positive diagonal elements. Only the upper triangle of R is referenced.

S2 — s^2 , the estimated variance of the error in the regression model. (Input)
 s^2 is the error mean square from the regression fit.

COVB — NCOEF by NCOEF matrix that is the estimated variance-covariance matrix of the estimated regression coefficients when R is nonsingular and is from an unrestricted regression fit. (Output)
See Comments for an explanation of COVB when R is singular or R is from a restricted regression fit. If R is not needed, COVB and R can share the same storage locations.

Optional Arguments

NCOEF — Number of regression coefficients in the model. (Input)
Default: NCOEF = size (R,1).

LDR — Leading dimension of R exactly as specified in the dimension statement in the calling program. (Input)
Default: LDR = size (R,1).

LDCOVB — Leading dimension of COVB exactly as specified in the dimension statement in the calling program. (Input)
Default: LDCOVB = size (COVB,1).

FORTRAN 90 Interface

Generic: CALL RCOVB (R, S2, COVB [,...])

Specific: The specific interface names are S_RCOVB and D_RCOVB.

FORTRAN 77 Interface

Single: CALL RCOVB (NCOEF, R, LDR, S2, COVB, LDCOVB)

Double: The double precision name is DRCOVB.

Example 1

This example uses a data set discussed by Draper and Smith (1981, pages 629-630). This data set is put into the matrix X by routine GDATA (see Chapter 19, Utilities). There are 4 independent variables and 1 dependent variable. Routine RGIVN (page 117) is invoked to fit the regression model, and RCOVB is invoked to compute summary statistics.

```
!                               SPECIFICATIONS FOR LOCAL VARIABLES
USE RCOVB_INT
USE GDATA_INT
USE RGIVN_INT
USE WRRRL_INT
INTEGER INTCEP, LDB, LDCEOF, LDCOVB, LDR, LDSCPE, LDX, NCOEF, &
        NDEP, NDX, NIND
PARAMETER (INTCEP=1, LDX=13, NDEP=1, NDX=5, NIND=4, &
```

```

LDSCPE=NDEP, NCOEF=INTCEP+NIND, LDB=NCOEF, &
LDCOEF=NCOEF, LDCOVB=NCOEF, LDR=NCOEF)
!
INTEGER IDEP, IDO, IFRQ, IIND, INDDEP(1), INDIND(1), IRANK, &
ICEN, IWT, NCOL, NRMIS, NROW
REAL B(LDB,NDEP), COVB(LDCOVB,5), DFE, R(LDR,NCOEF), &
S2, SCPE(LDSCPE,NDEP), X(LDX,NDX)
CHARACTER CLABEL(6)*10, RLABEL(5)*10
!
DATA RLABEL/'Intercept', 'X1', 'X2', 'X3', 'X4'/
DATA CLABEL/' ', 'Intercept', 'X1', 'X2', 'X3', 'X4'/
!
CALL GDATA (5, X, NROW, NCOL)
IIND = -NIND
IDEP = -NDEP
CALL RGIVN (X, IIND, INDIND, IDEP, INDDEP, B, R=R, DFE=DFE, &
SCPE=SCPE)
S2 = SCPE(1,1)/DFE
!
CALL RCOVB (R, S2, COVB)
CALL WRRRL ('COVB', COVB, RLABEL, CLABEL, FMT='(2W10.4)')
!
END

```

Output

	Intercept	COVB			
		X1	X2	X3	X4
Intercept	4910.0	-50.51	-50.60	-51.66	-49.60
X1	-50.5	0.55	0.51	0.55	0.51
X2	-50.6	0.51	0.52	0.53	0.51
X3	-51.7	0.55	0.53	0.57	0.52
X4	-49.6	0.51	0.51	0.52	0.50

Comments

When R is nonsingular and comes from an unrestricted regression fit, COVB is the estimated variance-covariance matrix of the estimated regression coefficients, and $\text{COVB} = s^2(R^T R)^{-1}$. Otherwise, variances and covariances of estimable functions of the regression coefficients can be obtained using COVB , and $\text{COVB} = s^2 G D G^T$. Here, D is the diagonal matrix with diagonal elements equal to 0 if the corresponding rows of R are restrictions and with diagonal elements equal to one otherwise. Also, G is a particular generalized inverse of R . See the Algorithm section.

Description

Routine `RCOVB` computes an estimated variance-covariance matrix of estimated regression parameters from the R matrix in several models. In the simplest situation, the model is a general linear model given by $y = X\beta + \varepsilon$ where y is the $n \times 1$ vector of responses, X is the $n \times p$ matrix of regressors, β is the $p \times 1$ vector of regression coefficients, and ε is the $n \times 1$ vector of errors whose elements are each independently distributed with mean 0 and variance σ^2 . Routine `RGIVN` ([page 117](#)) can be used to get the fit of the model and the R matrix.

If the i -th element of ε has variance σ^2/w_i and the weights w_i are used in the fit of the model, `RCOVB` produces the estimated variance-covariance matrix from the R matrix in the weighted least squares fit. More generally, if the variance-covariance matrix of ε is σ^2V , `RCOVB` can be used to produce the estimated variance-covariance matrix from the generalized least-squares fit. (Routine `RGIVN` can be used to perform a generalized least-squares fit, by regressing y^* on X^* where $y^* = (T^{-1})^T y$, $X^* = (T^{-1})^T X$ and T satisfies $T^T T = V$.)

If the general linear model has the restriction $H\beta = g$ on the regression parameters and this restriction is used in the fit of the model by routine `RLEQU` (page 142), `RCOVB` produces the estimated variance-covariance from the R matrix in the restricted least squares fit.

Routine `RCOVB` computes an estimated variance-covariance matrix for the estimated regression coefficients,

$$\hat{B}$$

in a fitted multivariate general linear model. The model is $Y = XB + E$ where Y is the $n \times q$ matrix of responses, X is the $n \times p$ matrix of regressors, B is the $p \times q$ matrix of regression coefficients, and E is the $n \times q$ matrix of errors whose rows are each independently distributed as a q -dimensional multivariate normal each with mean vector 0 and variance-covariance matrix Σ . Let

$$\hat{B} = (\hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_q)$$

The estimated covariance matrix

$$\text{Cov}(\hat{\beta}_i, \hat{\beta}_j) = s_{ij} (X^T X)^{-1}$$

Here, s_{ij} (input in `S2`) is the estimate of the ij -th element of Σ .

If a nonlinear regression model is fit using routine `RNLIN` (page 304), `RCOVB` produces the asymptotic estimated variance-covariance matrix from the R matrix in that fit.

If R is singular, corresponding to $\text{rank}(R) < p$, a generalized inverse is used to compute `COVB`. For a matrix G to be a g_i ($i = 1, 2, 3$, or 4) inverse of a matrix A , G must satisfy conditions j (for $j \leq i$) for the Moore-Penrose inverse but, generally, must fail conditions k (for $k > i$). The four conditions for G to be a Moore-Penrose inverse of A are as follows:

1. $AGA = A$
2. $GAG = G$
3. AG is symmetric
4. GA is symmetric

In the case that R is singular, the method for obtaining `COVB` follows the discussion of Maindonald (1984, pages 101–103). Let Z be the diagonal matrix with diagonal elements defined by

$$z_{ii} = \begin{cases} 1 & \text{if } r_{ii} \neq 0 \\ 0 & \text{if } r_{ii} = 0 \end{cases}$$

Let G be the solution to $RG = Z$ obtained by setting the i -th ($\{i : r_{ii} = 0\}$) row of G to zero. `COVB` is set to s^2GG^T . (G is a g_3 inverse of R . For any g_3 inverse of R , represented by

$$R^{g_3}$$

the result

$$R^{g_3} R^{g_3 T}$$

is a symmetric g_2 inverse of $R^T R = X^T X$. See Sallas and Lioni [1988].)

Note that `COVB` can only be used to get variances and covariances of estimable functions of the regression coefficients, i.e., nonestimable functions (linear combinations of the regression coefficients not in the space spanned by the nonzero rows of R) must not be used. See, for example, Maindonald (1984, pages 166–168) for a discussion of estimable functions.

The preceding discussion can be modified to include the restricted least-squares problem. The modification is based on the work of Stirling (1981). Let the matrix $D = \text{diag}(d_1, d_2, \dots, d_p)$ be a diagonal matrix with elements $d_{ii} = 0$ if the i -th row of R corresponds to a restriction and 1 otherwise. In the unrestricted case, D is simply the $p \times p$ identity matrix. The formula for `COVB` is s^2GDG^T .

Additional Example

Example 2

In this example, routine `RNLIN` (page 304) is first invoked to fit the following nonlinear regression model discussed by Neter, Wasserman, and Kutner (1983, pages 475–478):

$$y_i = \theta_1 e^{\theta_2 x_i} + \varepsilon_i \quad i = 1, 2, \dots, 15$$

Then, `RCOVB` is used to compute the estimated asymptotic variance-covariance matrix of the estimated nonlinear regression parameters. Finally, the diagonal elements of the output matrix from `RCOVB` are used together with routine `TIN` (see Chapter 17, Probability Distribution Functions and Inverses) to compute 95% confidence intervals on the regression parameters.

```

USE IMSL_LIBRARIES
INTEGER    LDR, NOBS, NPARM
PARAMETER (NOBS=15, NPARM=2, LDR=NPARM)
!
INTEGER    I, IDERIV, IRANK, ISETNG, NOUT
REAL       A, DFE, R(LDR,NPARM), SQRT, SSE, THETA(NPARM)
INTRINSIC  SQRT
EXTERNAL   EXAMPL
!
DATA THETA/60.0, -0.03/
!
CALL UMACH (2, NOUT)

```

```

!
  IDERIV = 1
  CALL RNLIN (EXAMPL, THETA, IDERIV=IDERIV, R=R, DFE=DFE, SSE=SSE)
!
  CALL RCOVB (R, SSE/DFE, R)
!
!                                     Print
  ISETNG=2
  CALL WROPT (-6, ISETNG, 0)
  CALL WRRRN ('Estimated Asymptotic Variance-Covariance Matrix', &
    R)
!
!                                     Compute and print 95 percent
!                                     confidence intervals.
  WRITE (NOUT,*)
  WRITE (NOUT,*) '          95% Confidence Intervals          '
  WRITE (NOUT,*) ' Estimate Lower Limit Upper Limit'
  DO 10 I=1, NPARM
    A = TIN(0.975,DFE)*SQRT(R(I,I))
    WRITE (NOUT,'(1X, F10.3, 2F13.3)') THETA(I), THETA(I) - A, &
      THETA(I) + A
10 CONTINUE
  END
!
  SUBROUTINE EXAMPL (NPARM, THETA, IOPT, IOBS, FRQ, WT, E, DE, &
    IEND)
  INTEGER  NPARM, IOPT, IOBS, IEND
  REAL    THETA(NPARM), FRQ, WT, E, DE(NPARM)
!
  INTEGER  NOBS
  PARAMETER (NOBS=15)
!
  REAL    EXP, XDATA(NOBS), YDATA(NOBS)
  INTRINSIC  EXP
!
  DATA YDATA/54.0, 50.0, 45.0, 37.0, 35.0, 25.0, 20.0, 16.0, 18.0, &
    13.0, 8.0, 11.0, 8.0, 4.0, 6.0/
  DATA XDATA/2.0, 5.0, 7.0, 10.0, 14.0, 19.0, 26.0, 31.0, 34.0, &
    38.0, 45.0, 52.0, 53.0, 60.0, 65.0/
!
  IF (IOBS .LE. NOBS) THEN
    WT = 1.0E0
    FRQ = 1.0E0
    IEND = 0
    IF (IOPT .EQ. 0) THEN
      E = YDATA(IOBS) - THETA(1)*EXP(THETA(2)*XDATA(IOBS))
    ELSE
      DE(1) = -EXP(THETA(2)*XDATA(IOBS))
      DE(2) = -THETA(1)*XDATA(IOBS)*EXP(THETA(2)*XDATA(IOBS))
    END IF
  ELSE
    IEND = 1
  END IF
  RETURN
  END

```

Output

Estimated Asymptotic Variance-Covariance Matrix

	1	2
1	2.16701E+00	-1.78121E-03
2	-1.78121E-03	2.92786E-06

95% Confidence Intervals

Estimate	Lower Limit	Upper Limit
58.603	55.423	61.784
-0.040	-0.043	-0.036

CESTI

Constructs an equivalent completely testable multivariate general linear hypothesis $HBU = G$ from a partially testable hypothesis $H_pBU = G_p$.

Required Arguments

HP — NHP by NCOEF matrix H_p with each row corresponding to a row in the hypothesis and containing the constants that specify a linear combination of the regression coefficients. (Input)

NDEP — Number of dependent (response) variables. (Input)

NU — U matrix option. (Input)

For positive NU, NU is the number of linear combinations of the dependent variables to be considered. If NU = 0, the hypothesis is $H_pB = G_p$, and U is automatically taken to be the identity. NU must be less than or equal to NDEP.

GP — Matrix G_p containing the null hypothesis values. (Input)

If NU = 0, then GP is NHP by NDEP; otherwise, GP is NHP by NU.

R — NCOEF by NCOEF upper triangular matrix containing the R matrix. (Input)

The R matrix can come from a regression fit based on a QR decomposition of the matrix of regressors or based on a Cholesky factorization $R^T R$ of the matrix of sums of squares and crossproducts of the regressors. Elements to the right of a diagonal element of R that is zero must also be zero. A zero row indicates a nonfull rank model. For an R matrix that comes from a regression fit with linear equality restrictions on the parameters, each row of R corresponding to a restriction must have a corresponding diagonal element that is negative. The remaining rows of R must have positive diagonal elements. Only the upper triangle of R is referenced.

IRANKP — Rank of H_p . (Output)

NH — Number of rows in the completely testable hypothesis (also, the degrees of freedom for the hypothesis). (Output)

The degrees of freedom for the hypothesis (NH) classify the hypothesis $H_p BU = G_p$ as nontestable ($NH = 0$), partially testable ($0 < NH < IRANKP$), or completely testable ($0 < NH = IRANKP$).

H — NH by $NCOEF$ matrix H with each row corresponding to a row in the completely testable hypothesis and containing the constants that specify an estimable linear combination of the regression coefficients. (Output)

If HP is not needed, H and HP can occupy the same storage locations.

G — Matrix G containing the null hypothesis values for the completely testable hypothesis. (Output)

If $NU = 0$, then G is NH by $NDEP$, otherwise, G is NH by NU . If GP is not needed, G and GP can occupy the same storage locations.

Optional Arguments

NHP — Number of rows in the hypothesis. (Input)

Default: $NHP = \text{size}(HP,1)$.

$NCOEF$ — Number of regression coefficients in the model. (Input)

Default: $NCOEF = \text{size}(HP,2)$.

$LDHP$ — Leading dimension of HP exactly as specified in the dimension statement of the calling program. (Input)

Default: $LDHP = \text{size}(HP,1)$.

$LDGP$ — Leading dimension of GP exactly as specified in the dimension statement in the calling program. (Input)

Default: $LDGP = \text{size}(GP,1)$.

LDR — Leading dimension of R exactly as specified in the dimension statement in the calling program. (Input)

Default: $LDR = \text{size}(R,1)$.

LDH — Leading dimension of H exactly as specified in the dimension statement of the calling program. (Input)

Default: $LDH = \text{size}(H,1)$.

LDG — Leading dimension of G exactly as specified in the dimension statement in the calling program. (Input)

Default: $LDG = \text{size}(G,1)$.

FORTRAN 90 Interface

Generic: `CALL CESTI (HP, NDEP, NU, GP, R, IRANKP, NH, H, G [, ...])`

Specific: The specific interface names are `S_CESTI` and `D_CESTI`.

FORTRAN 77 Interface

Single: CALL CESTI (NHP, NCOEF, HP, LDHP, NDEP, NU, GP, LDGP, R,
 LDR, IRANKP, NH, H, LDH, G, LDG)

Double: The double precision name is DCESTI.

Example

A one-way analysis-of-variance model discussed by Peixoto (1986) is fitted to some data. The model is

$$y_{ij} = \mu + \alpha_i + \varepsilon_{ij} \quad (i, j) = (1, 1), (2, 1), (2, 2)$$

The model is fitted using routine RGLM (page 128). Next, the partially testable hypothesis

$$H_0: \begin{matrix} \alpha_1 = 5 \\ \alpha_2 = 3 \end{matrix}$$

is converted to a completely testable hypothesis using CESTI. Sum of squares associated with the hypothesis are computed using routine RHPSS (page 176). Finally, the F statistic is computed along with the associated p -value using routine FDF (see Chapter 17, Probability Distribution Functions and Inverses).

```
USE IMSL_LIBRARIES
INTEGER    LDB, LDG, LDGP, LDH, LDHP, LDR, LDSCPE, LDSCPH, &
          LDX, LINDEF, MAXB, NCOL, NDEP, NEF, NHP, NROW
PARAMETER (LINDEF=1, MAXB=3, MAXCL=2, NCLVAR=1, NCOL=2, &
          NDEP=1, NEF=1, NHP=2, NROW=3, LDB=MAXB, LDG=NHP, &
          LDGP=NHP, LDH=NHP, LDHP=NHP, LDR=MAXB, LDSCPE=NDEP, &
          LDSCPH=NDEP, LDX=NROW)
!
INTEGER    INDCL(NCLVAR), INDDEP(NDEP), INDEF(LINDEF), INTCEP, &
          IRANK, IRANKP, IRBEF(NEF+1), NCOEF, NH, NOUT, &
          NVEF(NEF)
REAL       B(LDB, NDEP), DFE, DFH, F, G(LDG, NDEP), GP(LDGP, NDEP), &
          H(LDH, MAXB), HP(LDHP, MAXB), PVALUE, R(LDR, MAXB), &
          SCPE(LDSCPE, NDEP), SCPH(LDSCPH, NDEP), X(LDX, NCOL)
!
DATA X/1.0, 2.0, 2.0, 17.3, 24.1, 26.3/
DATA INDCL/1/, NVEF/1/, INDEF/1/, INDDEP/2/
DATA (HP(1, J), J=1, MAXB)/0.0, 1.0, 0.0/
DATA (HP(2, J), J=1, MAXB)/0.0, 0.0, 1.0/
DATA GP/5.0, 3.0/
!
CALL RGLM (X, INDCL, NVEF, INDEF, NDEP, INDDEP, MAXCL, B, &
          IRBEF=IRBEF, R=R, DFE=DFE, SCPE=SCPE)
NCOEF = IRBEF(NEF+1) - 1
!
NU = 0
CALL CESTI (HP, NDEP, NU, GP, R, IRANKP, NH, H, G, NCOEF=NCOEF)
!
CALL UMACH (2, NOUT)
IF (NH .EQ. 0) THEN
```

```

        WRITE (NOUT,*) 'Nontestable hypothesis'
    ELSE IF (NH .LT. IRANKP) THEN
        WRITE (NOUT,*) 'Partially testable hypothesis'
    ELSE
        WRITE (NOUT,*) 'Completely testable hypothesis'
    END IF
    CALL WRRRN ('H', H, NH, NCOEF, LDH)
    CALL WRRRN ('G', G, NH, NDEP, LDG)
    CALL RHPSS (H, B, G, R, SCPH, DFH=DFH)
!
    F      = (SCPH(1,1)/DFH)/(SCPE(1,1)/DFE)
    PVALUE = 1.0 - FDF(F,DFH,DFE)
    WRITE (NOUT,*)
    WRITE (NOUT,*) 'Degrees of      Sum of      Prob. of'
    WRITE (NOUT,*) 'Freedom      Squares      F-statistic      Larger F'
    WRITE (NOUT,99999) DFH, SCPH(1,1), F, PVALUE
99999 FORMAT (F8.1, 3X, 1F10.3, F11.3, 2X, F10.4)
    END

```

Output

Partially testable hypothesis

H		
1	2	3
0.0000	0.7071	-0.7071

G
1.414

Degrees of Freedom	Sum of Squares	F-statistic	Prob. of Larger F
1.0	65.340	27.000	0.1210

Comments

1. Workspace may be explicitly provided, if desired, by use of C2STI/DC2STI. The reference is:

```
CALL C2STI (NCOEF, NHP, HP, LDHP, NDEP, NU, GP, LDGP, R, LDR,
IRANKP, NH, H, LDH, G, LDG, IWK, WK)
```

The additional arguments are as follows:

IWK — Work vector of length $\max\{NHP, NCOEF\}$.

WK — Work vector of length $NCOEF * m + NCOEF^2 + NHP^2 + n * r + n^2 + m + \max\{2 * m, n + r + \max(n, r) - 1\}$.

2. Informational errors
Type Code

4	1	There is inadequate space to store the completely testable hypothesis. Increase LDH or LDG so that it is greater than or equal to NH.
---	---	---

Description

Once a general linear model $y = X\beta + \varepsilon$ is fitted, particular hypothesis tests are frequently of interest. If the matrix of regressors X is not full rank (as evidenced by the fact that some diagonal elements of the R matrix output from the fit are equal to zero), methods that use the results of the fitted model to compute the hypothesis sum of squares (see routine `RHPSS`, page 176) require one to specify in the hypothesis only linear combinations of the regression parameters that are estimable. A linear combination of regression parameters $c^T \beta$ is *estimable* means that there exists some vector a such that $c^T = a^T X$, i.e., c^T is in the space spanned by the rows of X . For a further discussion of estimable functions, see Maindonald (1984, pages 166–168) and Searle (1971, pages 180 – 188). Routine `CESTI` is only useful in the case of nonfull rank regression models, i.e., when the problem of estimability arises.

Peixoto (1986) noted that the customary definition of testable hypothesis in the context of a general linear hypothesis test $H\beta = g$ is overly restrictive. He extended the notion of a testable hypothesis (a hypothesis composed of estimable functions of the regression parameters) to include partially testable and completely testable hypotheses. A hypothesis $H\beta = g$ is *partially testable* means that the intersection of the row space of H (denoted by $R(H)$) and the row space of X ($R(X)$) is not essentially empty and is a proper subset of $R(H)$, i.e., $\{0\} \subset R(H) \cap R(X) \subset R(H)$. A hypothesis $H\beta = g$ is *completely testable* means that $\{0\} \subset R(H) \subseteq R(X)$. Peixoto also demonstrated a method for converting a partially testable hypothesis to one that is completely testable so that the usual method for obtaining the sum of squares for the hypothesis from the results of the fitted model can be used. The method replaces H_p in the partially testable hypothesis $H_p \beta = g_p$ by a matrix H whose rows are a basis for the intersection of the row space of H_p and the row space of X . A corresponding conversion of the null hypothesis values from g_p to g is also made. A sum of squares for the completely testable hypothesis can then be computed (see routine `RHPSS`). The sum of squares that is computed for the hypothesis $H\beta = g$ equals the difference in the error sums of squares from two fitted models: the restricted model with the partially testable hypothesis $H_p \beta = g_p$ adjoined to the model as linear equality restrictions (see routine `RLEQU` on page 142) and the unrestricted model.

Routines `RGLM` (page 128), `RGIVN` (page 117), `RLEQU` (page 142), and `RCOV` (page 113) can be used to compute the fit of the general linear model prior to invoking `CESTI`. The R matrix is required for input to `CESTI`. After converting a partially testable hypothesis to a completely testable hypothesis, `RHPSS` (page 176) can be invoked to compute the sum of squares for the hypothesis.

For the general case of the multivariate general linear model $Y = XB + E$ (see the chapter introduction, page 75) with possible linear equality restrictions on the regression parameters, `CESTI` converts the partially testable hypothesis $H_p BU = G_p$ to a completely testable hypothesis $HBU = G$. For the case of the linear model with linear equality restrictions, the definitions of estimable functions, nontestable hypotheses, partially testable hypotheses, and completely testable hypothesis are similar to those previously given for the unrestricted model with the exception that $R(X)$ is replaced by $R(R)$ where R is the upper triangular matrix output from `RLEQU`. The nonzero rows of R form a basis for the row space of the matrix $(X^T, A^T)^T$. The rows of H form an orthonormal basis for the intersection of two subspaces: the subspace spanned by

the rows of H_p and the subspace spanned by the rows of R . The algorithm used by `CESTI` for computing the intersection of these two subspaces is based on an algorithm for computing angles between linear subspaces due to Bjorck and Golub (1973). (See also Golub and Van Loan 1983, pages 429–430). The method is closely related to a canonical correlation analysis discussed by Kennedy and Gentle (1980, 56–565). The algorithm is as follows:

1. Compute a QR factorization of

$$H_p^T$$

with column permutations so that

$$H_p^T = Q_1 R_1 P_1^T$$

Here, P_1 is the associated permutation matrix that is also an orthogonal matrix. Determine the rank of H_p as the number of nonzero diagonal elements of R_1 , say n_1 . Partition $Q_1 = (Q_{11}, Q_{12})$ so that Q_{11} is the first n_1 columns of Q_1 . Set $\text{IRANKP} = n_1$.

2. Compute a QR factorization of the transpose of the R matrix input to `CESTI` with column permutations so that

$$R^T = Q_2 R_2 P_2^T$$

Determine the rank of R from the number of nonzero diagonal elements of R , say n_2 . Partition $Q_2 = (Q_{21}, Q_{22})$ so that Q_{21} is the first n_2 columns of Q_2 .

3. Form

$$A = Q_{11}^T Q_{21}$$

4. Compute the singular values of A

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(n_1, n_2)}$$

and the left singular vectors W of the singular value decomposition of A so that

$$W^T A V = \text{diag}(\sigma_1, \dots, \sigma_{\min(n_1, n_2)})$$

If $\sigma_1 < 1$, then the dimension of the intersection of the two subspaces is $s = 0$. Otherwise, take the dimension of the intersection to be s if $\sigma_s = 1 > \sigma_{s+1}$. Set $\text{NH} = s$.

5. Let W_1 be the first s columns of W . Set $H = (Q_1 W_1)^T$.

6. Take R_{11} to be a NHP by NHP matrix related to R_1 as follows. If $\text{NHP} \leq \text{NCOEF}$, R_{11} equals the first NHP rows of R_1 . Otherwise, R_{11} contains R_1 in its first NCOEF rows and zeros in the remaining rows. Compute a solution Z to the linear system

$$R_{11}^T Z = P_1^T G_p$$

using routine `GIRTS` (IMSL MATH/LIBRARY). If this linear system is declared inconsistent, an error message with error code equal to 2 is issued.

7. Partition

$$Z^T = (Z_1^T, Z_2^T)$$

so that Z_1 is the first n_1 rows of Z . Set

$$G = W_1^T Z_1$$

The degrees of freedom (NH) classify the hypothesis $H_p BU = G_p$ as nontestable ($NH = 0$), partially testable ($0 < NH < IRANKP$), or completely testable ($0 < NH = IRANKP$).

For further details concerning the algorithm, see Sallas and Lioni (1988).

RHPSS

Computes the matrix of sums of squares and crossproducts for the multivariate general linear hypothesis $H BU = G$ given the coefficient estimates

$$\hat{B}$$

and the R matrix.

Required Arguments

H — NH by $NCOEF$ matrix H with each row corresponding to a row in the hypothesis and containing the constants that specify an estimable linear combination of the regression coefficients. (Input)

B — $NCOEF$ by $NDEP$ matrix

$$\hat{B}$$

containing a least-squares solution for the regression coefficients. (Input)

G — Matrix containing the null hypothesis values. (Input)

If $NU = 0$, then G is NH by $NDEP$; otherwise, G is NH by NU .

R — $NCOEF$ by $NCOEF$ upper triangular matrix containing the R matrix. (Input)

The R matrix can come from a regression fit based on a QR decomposition of the matrix of regressors or based on a Cholesky factorization $R^T R$ of the matrix of sums of squares and crossproducts of the regressors. Elements to the right of a diagonal element of R that is zero must also be zero. A zero row indicates a nonfull rank model. For an R matrix that comes from a regression fit with linear equality restrictions on the parameters, each row of R corresponding to a restriction must have a corresponding diagonal element that is negative. The remaining rows of R must have positive diagonal elements. Only the upper triangle of R is referenced.

SCPH — Matrix containing sums of squares and crossproducts attributable to the hypothesis. (Output)
If $NU = 0$, SCPH is a NDEP by NDEP matrix, otherwise, SCPH is a NU by NU matrix.

Optional Arguments

NH — Number of rows in the hypothesis. (Input)
Default: $NH = \text{size}(H,1)$.

NCOEF — Number of regression coefficients in the model. (Input)
Default: $NCOEF = \text{size}(H,2)$.

LDH — Leading dimension of H exactly as specified in the dimension statement of the calling program. (Input)
Default: $LDH = \text{size}(H,1)$.

NDEP — Number of dependent (response) variables. (Input)
Default: $NDEP = \text{size}(B,2)$.

LDB — Leading dimension of B exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDB = \text{size}(B,1)$.

NU — U matrix option. (Input)
For positive NU , NU is the number of linear combinations of the dependent variables to be considered. If $NU = 0$, the hypothesis is $HB = G$, i.e., U is automatically taken to be the identity. NU must be less than or equal to $NDEP$.
Default: $NU = 0$.

U — $NDEP$ by NU matrix U in test $HB U = G$. (Input, if NU is positive)
If $NU = 0$, U is not referenced and can be a 1 x 1 array.
Default: U is a 1x 1 array.

LDU — Leading dimension of U exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDU = \text{size}(U, 1)$.

LDG — Leading dimension of G exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDG = \text{size}(G,1)$.

LDR — Leading dimension of R exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDR = \text{size}(R,1)$.

DFH — Degrees of freedom for SCPH. (Output)
 DFH equals the rank of H .

LDSCPH — Leading dimension of SCPH exactly as specified in the dimension statement in the calling program. (Input)
 Default: LDSCPH = size (SCPH,1).

FORTRAN 90 Interface

Generic: CALL RHPSS (H, B, G, R, SCPH [,...])

Specific: The specific interface names are S_RHPSS and D_RHPSS.

FORTRAN 77 Interface

Single: CALL RHPSS (NH, NCOEF, H, LDH, NDEP, B, LDB, NU, U, LDU, G, LDG, R, LDR, DFH, SCPH, LDSCPH)

Double: The double precision name is DRHPSS.

Example 1

A two-way analysis-of-variance model is fitted to balanced data discussed by Snedecor and Cochran (1967, Table 12.5.1, page 347). The responses are the weight gains (in grams) of rats fed diets varying in two components-level of protein and source of protein. The model is

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \varepsilon_{ijk} \quad i = 1, 2; j = 1, 2, 3; k = 1, 2, \dots, 10$$

where

$$\sum_{i=1}^2 \alpha_i = 0; \sum_{j=1}^3 \beta_j = 0; \sum_{i=1}^2 \gamma_{ij} = 0 \text{ for } j = 1, 2, 3; \text{ and } \sum_{j=1}^3 \gamma_{ij} = 0 \text{ for } i = 1, 2$$

The model is fitted using routine RGLM (page 128). Next, the sum of squares for interaction

$$H_0: \begin{aligned} \gamma_{11} - \gamma_{12} - \gamma_{21} + \gamma_{22} &= 0 \\ \gamma_{11} - \gamma_{13} - \gamma_{21} + \gamma_{23} &= 0 \end{aligned}$$

is computed using RHPSS. Finally, the F statistic is computed along with the associated p -value using routine FDF (see Chapter 17, Probability Distribution Functions and Inverses).

```
USE RHPSS_INT
USE RGLM_INT
USE UMACH_INT
USE FDF_INT

INTEGER LDB, LDG, LDH, LDR, LDSCPE, LDSCPH, LDU, LDX, LINDEF, &
  MAXB, NCOL, NDEP, NEF, NH, NROW

PARAMETER (NDEP=1, LINDEF=4, MAXB=12, MAXCL=5, NCLVAR=2, NCOL=3, &
  NEF=3, NH=2, NROW=60, LDB=MAXB, LDG=NH, LDH=NH, &
  LDR=MAXB, LDSCPE=NDEP, LDSCPH=NDEP, LDX=NROW)
```

!


```

INTEGER  INDCL(NCLVAR),  INDEP(NDEP),  INDEF(LINDEF),  INTCEP,&
IRANK,  IRBEF(NEF+1),  NCOEF,  NOUT,  NVEF(NEF)
REAL    B(LDB,NDEP),  DFE,  DFH,  F,  G(LDG,NDEP),  H(LDH,MAXB),  &
PVALUE,  R(LDR,MAXB),  SCPE(LDSCPE,NDEP),  &
SCPH(LDSCPH,NDEP),  X(LDX,NCOL),  XMAX(MAXB),  &
XMIN(MAXB)
!
DATA X/73.0, 102.0, 118.0, 104.0, 81.0, 107.0, 100.0, 87.0, &
117.0, 111.0, 98.0, 74.0, 56.0, 111.0, 95.0, 88.0, 82.0, &
77.0, 86.0, 92.0, 94.0, 79.0, 96.0, 98.0, 102.0, 102.0, &
108.0, 91.0, 120.0, 105.0, 90.0, 76.0, 90.0, 64.0, 86.0, &
51.0, 72.0, 90.0, 95.0, 78.0, 107.0, 95.0, 97.0, 80.0, &
98.0, 74.0, 74.0, 67.0, 89.0, 58.0, 49.0, 82.0, 73.0, 86.0, &
81.0, 97.0, 106.0, 70.0, 61.0, 82.0, 30*1.0, 30*2.0, &
10*1.0, 10*2.0, 10*3.0, 10*1.0, 10*2.0, 10*3.0/
DATA INDCL/2, 3/, NVEF/1, 1, 2/, INDEF/2, 3, 2, 3/, INDEP/1/
DATA (H(1,J),J=1,MAXB)/6*0.0, 1.0, -1.0, 0.0, -1.0, 1.0, 0.0/
DATA (H(2,J),J=1,MAXB)/6*0.0, 1.0, 0.0, -1.0, -1.0, 0.0, 1.0/
DATA G/2*0.0/
!
CALL RGLM (X, INDCL, NVEF, INDEF, NDEP, INDEP, MAXCL, B, &
IRBEF=IRBEF, R=R, DFE=DFE, SCPE=SCPE)
!
NCOEF = IRBEF(NEF+1) - 1
CALL RHPSS (H, B, G, R, SCPH, DFH=DFH)
!
F      = (SCPH(1,1)/DFH)/(SCPE(1,1)/DFE)
PVALUE = 1.0 - FDF(F,DFH,DFE)
CALL UMACH (2, NOUT)
WRITE (NOUT,*) 'Degrees of      Sum of          Prob. of'
WRITE (NOUT,*) '  Freedom  Squares    F-statistic  Larger F'
WRITE (NOUT,99999) DFH, SCPH(1,1), F, PVALUE
99999 FORMAT (F8.1, 3X, 1F10.3, F11.3, 2X, F10.4)
END

```

Output

Degrees of Freedom	Sum of Squares	F-statistic	Prob. of Larger F
2.0	1178.135	2.746	0.0732

Comments

1. Workspace may be explicitly provided, if desired, by use of R2PSS/DR2PSS. The reference is:

```
CALL R2PSS (NCOEF, NH, H, LDH, NDEP, B, LDB, NU, U, LDU, G, LDG,
R, LDR, DFH, SCPH, LDSCPH, IWK, WK)
```

The additional arguments are as follows:

IWK — Work vector of length NH.

WK — Work vector of length $NH * (NDEP + NCOEF + \max(NCOEF, NH) + 3) + NU * NDEP - 1$.

2. Informational errors

Type Code

- | | | |
|---|---|--|
| 3 | 1 | The hypothesis is not completely testable. Each row of H must be in the space spanned by the rows of R . |
| 3 | 2 | The hypothesis is inconsistent. The linear system $HB U = G$ combined with any restrictions from a regression fit with linear equality restrictions must have a solution for B . |

3. $SCPH = (H \hat{B}U - G)^T (C^T DC)^- (H \hat{B}U - G)$

where $(C^T DC)^-$ is a generalized inverse of $C^T DC$, C is a solution to $R^T C = H^T$, and D is a diagonal matrix with

$$d_{ii} = \begin{cases} 1 & \text{if } r_{ii} > 0 \\ 0 & \text{if } r_{ii} \leq 0 \end{cases}$$

Description

Routine `RHPSS` computes the matrix of sums of squares and crossproducts for the general linear hypothesis $HBU = G$ for the multivariate general linear model $Y = XB + E$ with possible linear equality restrictions $AB = Z$. (See the chapter introduction for a description of the multivariate general linear model.) Routines `RGLM` (page 128), `RGIVN` (page 117), `RLEQU` (page 142), and `RCOV` (page 113) can be used to compute the fit of the general linear model prior to invoking `RHPSS`. The R matrix and \hat{B} from any of those routines are required for input to `RHPSS`.

The rows of H must be linear combinations of the rows of R , i.e., $HB = G$ must be completely testable. If the hypothesis is not completely testable, Routine `CESTI` (page 170) can be used to construct an equivalent completely testable hypothesis.

Computations are based on an algorithm discussed by Kennedy and Gentle (1980, page 317) that is extended by Sallas and Lioni (1988) for multivariate nonfull rank models with possible linear equality restrictions. The algorithm is as follows:

1. Form

$$W = H \hat{B}U - G$$

2. Find C as the solution of $R^T C = H^T$ using routine `GIRTS` (IMSL MATH/LIBRARY). If the equations are declared inconsistent within a computed tolerance, an error message with code 1 is issued that the hypothesis is not completely testable.
3. For all rows of R corresponding to restrictions, i.e., containing negative diagonal elements from a restricted least-squares fit using `RLEQU`, zero out the corresponding rows of C , i.e., form DC .

4. Decompose DC using Householder transformations and column pivoting to yield a square, upper triangular matrix T with diagonal elements of nonincreasing magnitude and permutation matrix P such that

$$DCP = Q \begin{bmatrix} T \\ 0 \end{bmatrix}$$

where Q is an orthogonal matrix.

5. Determine the rank of T , say r . If $t_{11} = 0$, then $r = 0$. Otherwise, the rank of T is r if

$$|t_{rr}| > |t_{11}| \varepsilon \geq |t_{r+1,r+1}|$$

where $\varepsilon = 10.0 * \text{AMACH}(4)$. Then, zero out all rows of T below row r . Set the degrees of freedom for the hypothesis, output in DFH, to r .

6. Find V as a solution to $T^T V = P^T W$ using routine `GIRTS`. If the equations are inconsistent, an error message with code 2 is issued that the hypothesis is inconsistent within a computed tolerance, i.e., the linear system

$$HBU = G$$

$$AB = Z$$

does not have a solution for B .

7. Form $V^T V$, which is the required matrix of sum of squares and crossproducts output in `SCPH`.

In general, the two errors with code 1 and 2 are serious user errors that require the user to correct the hypothesis before any meaningful sums of squares from this routine can be computed. However, in some cases, the user may know the hypothesis is consistent and completely testable, but the checks in `RHPSS` are too tight. For this reason, `RHPSS` continues with the computations.

Routine `RHPSS` gives a matrix of sums of squares and crossproducts that could also be obtained from separate fittings of the two models

$$\begin{aligned} Y^* &= XB^* + E^* \\ AB^* &= Z^* \\ HB^* &= G \end{aligned} \tag{1}$$

and

$$\begin{aligned} Y^* &= XB^* + E^* \\ AB^* &= Z^* \end{aligned} \tag{2}$$

where $Y^* = YU$, $B^* = BU$, $E^* = EU$, and $Z^* = ZU$. The error sum of squares and crossproduct matrix for (1) minus that for (2) is the matrix of sum of squares and crossproducts output in `SCPH`. Note that this approach avoids entirely the question of testability.

Additional Example

Example 2

The data for the second example are taken from Maindonald (1984, pages 203–204). The data are saved in the matrix X . A multivariate regression model containing two dependent variables and three independent variables is fit using routine `RGIVN` (page 117). The sum of squares and crossproducts matrix is computed for the third independent variable in the model.

```
USE IMSL_LIBRARIES
INTEGER INTCEP, LDB, LDG, LDH, LDR, LDSCPE, LDSCPH, LDX, &
        NCOEF, NCOL, NDEP, NH, NIND, NROW
PARAMETER (INTCEP=1, LDU=1, NCOL=5, NDEP=2, NH=1, NIND=3, &
        NROW=9, LDG=NH, LDH=NH, LDSCPE=NDEP, LDSCPH=NDEP, &
        LDX=NROW, NCOEF=INTCEP+NIND, LDB=NCOEF, LDR=NCOEF)
!
INTEGER IDEP, IIND, INDDEP(1), INDIND(1), &
        NOUT, NRMISS
REAL B(LDB,NDEP), D(NCOEF), DFE, DFH, G(LDG,NDEP), &
        H(LDH,NCOEF), R(LDR,NCOEF), SCPE(LDSCPE,NDEP), &
        SCPH(LDSCPH,NDEP), X(LDX,NCOL)
!
DATA (X(1,J),J=1,NCOL)/7.0, 5.0, 6.0, 7.0, 1.0/
DATA (X(2,J),J=1,NCOL)/2.0, -1.0, 6.0, -5.0, 4.0/
DATA (X(3,J),J=1,NCOL)/7.0, 3.0, 5.0, 6.0, 10.0/
DATA (X(4,J),J=1,NCOL)/-3.0, 1.0, 4.0, 5.0, 5.0/
DATA (X(5,J),J=1,NCOL)/2.0, -1.0, 0.0, 5.0, -2.0/
DATA (X(6,J),J=1,NCOL)/2.0, 1.0, 7.0, -2.0, 4.0/
DATA (X(7,J),J=1,NCOL)/-3.0, -1.0, 3.0, 0.0, -6.0/
DATA (X(8,J),J=1,NCOL)/2.0, 1.0, 1.0, 8.0, 2.0/
DATA (X(9,J),J=1,NCOL)/2.0, 1.0, 4.0, 3.0, 0.0/
DATA H/3*0.0, 1.0/, G/0.0, 0.0/
!
IIND = -NIND
IDEP = -NDEP
CALL RGIVN (X, IIND, INDIND, IDEP, INDDEP, B, R=R)
CALL RHPSS (H, B, G, R, SCPH, DFH=DFH)
CALL UMACH (2, NOUT)
WRITE (NOUT,*) 'DFH = ', DFH
CALL WRRRN ('SCPH', SCPH)
END
```

Output

```
DFH =      1.00000

      SCPH
      1      2
1    100.0   -40.0
2    -40.0    16.0
```

RHPTE

Performs tests for a multivariate general linear hypothesis $HBU = G$ given the hypothesis sums of squares and crossproducts matrix S_H and the error sums of squares and crossproducts matrix S_E .

Required Arguments

DFE — Degrees of freedom for error matrix $SCPE$. (Input)

SCPE — $NDEP$ by $NDEP$ matrix S_E containing sums of squares and crossproducts for error. (Input)

DFH — Degrees of freedom for hypothesis matrix S_H . (Input)

SCPH — Matrix S_H containing sums of squares and crossproducts attributable to the hypothesis. (Input)
If $NU = 0$, S_H is a $NDEP$ by $NDEP$ matrix; otherwise, S_H is a NU by NU matrix.

TEST — Vector of length 8 containing test statistics and p -values for the hypothesis $HBU = G$. (Output)

Elem. Description

- 1, 5 Wilks' lambda and p -value
- 2, 6 Roy's maximum root criterion and p -value
- 3, 7 Hotelling's trace and p -value
- 4, 8 Pillai's trace and p -value

Optional Arguments

NDEP — Number of dependent variables. (Input)
Default: $NDEP = \text{size}(SCPE, 2)$.

LDSCPE — Leading dimension of $SCPE$ exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDSCPE = \text{size}(SCPE, 1)$.

NU — U matrix option. (Input)
For positive NU , NU is the number of linear combinations of the dependent variables to be considered. If $NU = 0$, the hypothesis is $HB = G$, i.e., U is automatically taken to be the identity.
Default: $NU = 0$.

U — NDEP by NU matrix used to test $HBU = G$. (Input, if NU is positive)
 The rank of the matrix U must equal the number of columns. If NU = 0, U is not referenced and can be a 1 x 1 array.
 Default: U is a 1 x 1 array.

LDU — Leading dimension of U exactly as specified in the dimension statement in the calling program. (Input)
 Default: $LDU = \text{size}(U, 1)$.

$LDSCPH$ — Leading dimension of $SCPH$ exactly as specified in the dimension statement in the calling program. (Input)
 Default: $LDSCPH = \text{size}(SCPH, 1)$.

FORTRAN 90 Interface

Generic: CALL RHPTE (DFE, SCPE, DFH, SCPH, TEST [,...])

Specific: The specific interface names are S_RHPTE and D_RHPTE.

FORTRAN 77 Interface

Single: CALL RHPTE (DFE, NDEP, SCPE, LDSCPE, NU, U, LDU, DFH, SCPH, LDSCPH, TEST)

Double: The double precision name is DRHPTE.

Example

The data for the example are taken from Maindonald (1984, pages 203–204). The data are stored in the matrix X . A multivariate regression model containing two dependent variables and three independent variables is fit using routine `RGIVN` (page 117). The sum of squares and crossproducts matrix is computed for the third independent variable in the model using `RHPSS` (page 176). Routine `RHPTE` is used to test whether the third independent variable should be included in the regression.

```

USE IMSL_LIBRARIES
INTEGER LDB, LDG, LDH, LDR, LDSCPE, LDSCPH, LDU, LDX, &
        NCOEF, NCOL, NDEP, NH, NIND, NROW
PARAMETER (INTCEP=1, LDU=1, NCOL=5, NDEP=2, NH=1, NIND=3, &
        NROW=9, LDG=NH, LDH=NH, LDSCPE=NDEP, LDSCPH=NDEP, &
        LDX=NROW, NCOEF=INTCEP+NIND, LDB=NCOEF, LDR=NCOEF)
!
INTEGER IDEP, IND, INDDEP(1), INDIND(1)
REAL B(LDB,NDEP), DFE, DFH, G(LDG,NDEP), &
      H(LDH,NCOEF), R(LDR,NCOEF), SCPE(LDSCPE,NDEP), &
      SCPH(LDSCPH,NDEP), TEST(8), X(LDX,NCOL)
CHARACTER CLABEL(3)*14, RLABEL(4)*9
!
DATA (X(1,J),J=1,NCOL)/7.0, 5.0, 6.0, 7.0, 1.0/
DATA (X(2,J),J=1,NCOL)/2.0, -1.0, 6.0, -5.0, 4.0/

```

```

DATA (X(3,J),J=1,NCOL)/7.0, 3.0, 5.0, 6.0, 10.0/
DATA (X(4,J),J=1,NCOL)/-3.0, 1.0, 4.0, 5.0, 5.0/
DATA (X(5,J),J=1,NCOL)/2.0, -1.0, 0.0, 5.0, -2.0/
DATA (X(6,J),J=1,NCOL)/2.0, 1.0, 7.0, -2.0, 4.0/
DATA (X(7,J),J=1,NCOL)/-3.0, -1.0, 3.0, 0.0, -6.0/
DATA (X(8,J),J=1,NCOL)/2.0, 1.0, 1.0, 8.0, 2.0/
DATA (X(9,J),J=1,NCOL)/2.0, 1.0, 4.0, 3.0, 0.0/
DATA H/3*0.0, 1.0/, G/0.0, 0.0/
DATA RLABEL/'Wilks', 'Roy', 'Hotelling', 'Pillai'/
DATA CLABEL/' ', 'Test statistic', 'p-value'/
!
IIND = -NIND
IDEP = -NDEP
CALL RGIWN (X, IIND, INDIND, IDEP, INDDEP, B, R=R, DFE=DFE, SCPE=SCPE)
CALL RHPSS (H, B, G, R, SCPH, DFH=DFH)
CALL RHPTE (DFE, SCPE, DFH, SCPH, TEST)
CALL WRRRL (' ', TEST, RLABEL, CLABEL, 4, 2, 4, FMT= '(F14.3, F9.6)')
END

```

Output

	Test statistic	p-value
Wilks	0.003	0.000010
Roy	316.601	0.000010
Hotelling	316.601	0.000010
Pillai	0.997	0.000010

Comments

1. Workspace may be explicitly provided, if desired, by use of R2PTE/DR2PTE. The reference is:

```
CALL R2PTE (DFE, NDEP, SCPE, LDSCPE, NU, U, LDU, DFH, SCPH,
LDSCPH, TEST, WK)
```

The additional argument is:

WK — Work vector of length $2 * p^2 + 2 * p + NDEP + 2 * NU^2$.

2. Informational errors

Type Code

- | | | |
|---|---|--|
| 3 | 1 | $U^T S_E U$ is singular. Only the Pillai trace statistic can be computed. Other statistics are set to NaN. |
| 4 | 2 | $U^T S_E U + S_H$ is singular. No tests can be computed. |
| 4 | 3 | Iterations for eigenvalues for the generalized eigenvalue problem $S_H x = \lambda(S_H + U^T S_E U)x$ failed to converge. Statistics cannot be computed. |

Description

Routine `RHPTE` computes test statistics and p -values for the general linear hypothesis $HBU = G$ for the multivariate general linear model. See the section “Multivariate General Linear Model” in the chapter introduction (page 75).

Routines `RGLM` (page 128), `RGIVN` (page 117), `RLEQU` (page 142), and `RCOV` (page 113) can be used to compute the fit of the general linear model prior to invoking `RHPTE`. The error sum of squares and crossproducts matrix (`SCPE`) is required for input to `RHPTE`. In addition, the hypothesis sum of squares and crossproducts matrix (`SCPH`), which can be computed using routine `RHPSS` (page 176), is required for input to `RHPTE`.

The hypothesis sum of squares and crossproducts matrix input in `SCPH` is

$$S_H = (H \hat{B}U - G)^T (C^T DC)^- (H \hat{B}U - G)$$

where C is a solution to $R^T C = H$ and where D is a diagonal matrix with diagonal elements

$$d_{ii} = \begin{cases} 1 & \text{if } r_{ii} > 0 \\ 0 & \text{otherwise} \end{cases}$$

See the section “Linear Dependence and the R Matrix” in the chapter introduction (page 77).

The error sum of squares and crossproducts matrix for the model $Y = XB + E$ is

$$(Y - X\hat{B})^T (Y - X\hat{B})$$

which is input in `SCPE`. The error sum of squares and crossproducts matrix for the hypothesis $HBU = G$ computed by `RHPTE` is

$$S_E = U^T (Y - X\hat{B})^T (Y - X\hat{B}) U$$

Let p equal the order of the matrices S_E and S_H , i.e.,

$$p = \begin{cases} \text{NU} & \text{if NU} > 0 \\ \text{NDEP} & \text{otherwise} \end{cases}$$

Let q (stored in `DFH`) be the degrees of freedom for the hypothesis. Let v (stored in `DFE`) be the degrees of freedom for error. Routine `RHTPE` computes three test statistics based on eigenvalues λ_i ($i = 1, 2, \dots, p$) of the generalized eigenvalue problem $S_H x = \lambda S_E x$. These test statistics are as follows:

Wilks' lambda

$$\begin{aligned} \Lambda &= \frac{\det(S_E)}{\det(S_H + S_E)} \\ &= \prod_{i=1}^p \frac{1}{1 + \lambda_i} \end{aligned}$$

Λ is output in TEST(1). The p -value output in TEST(5) is based on an approximation discussed by Rao (1973, page 556). The statistic

$$F = \frac{ms - pq/2 + 1}{pq} \frac{1 - \Lambda^{1/s}}{\Lambda^{1/s}}$$

has an approximate F distribution with pq and $ms - pq/2 + 1$ numerator and denominator degrees of freedom, respectively, where

$$s = \begin{cases} 1 & \text{if } p = 1 \text{ or } q = 1 \\ \sqrt{\frac{p^2 q^2 - 4}{p^2 + q^2 - 5}} & \text{otherwise} \end{cases}$$

and

$$m = v - (p - q + 1)/2$$

The F test is exact if $\min(p, q) \leq 2$ (Kshirsagar 1972, Theorem 4, pages 299–300).

Roy's maximum root

$$c = \max_i \lambda_i$$

c is output in TEST(2). The p -value output in TEST(6) is based on the approximation

$$F = \frac{v + q - s}{s} c$$

where $s = \max(p, q)$ has an approximate F distribution with s and $v + q - s$ numerator and denominator degrees of freedom, respectively. The F test is exact if $s = 1$, and then the p -value output in TEST(7) is exact. In general, the value output in TEST(7) is a lower bound on the actual p -value.

Hotelling's trace

$$U = \text{tr}(HE^{-1}) = \sum_{i=1}^p \lambda_i$$

U is output in TEST(3). The p -value output in TEST(7) is based on the approximation of McKeon (1974) that supersedes the approximation of Hughes and Saw (1972). McKeon's approximation is also discussed by Seber (1984, page 39). For

$$b = 4 + \frac{pq + 2}{\frac{(v + q - p - 1)(v - 1)}{(v - p - 3)(v - p)} - 1}$$

the p -value output in TEST(7) is based on the result that

$$F = \frac{b(v-p-1)U}{(b-2)pq}$$

has an approximate F distribution with pq and b degrees of freedom. The test is exact if $\min(p, q) = 1$. For $v \leq p + 1$, the approximation is not valid, and `TEST(7)` is set to NaN (not a number).

These three test statistics are valid when S_E is positive definite. A necessary condition for S_E to be positive definite is $v \geq p$. If S_E is not positive definite, a warning error message with error code 1 is issued, and the entries in `TEST` corresponding to the computed test statistics and p -values are set to NaN (not a number).

Because the requirement $v \geq p$ can be a serious drawback, `RHTPE` computes a fourth test statistic based on eigenvalues $\theta_i (i = 1, 2, \dots, p)$ of the generalized eigenvalue problem $S_H w = \theta(S_H + S_E)w$. This test statistic requires a less restrictive assumption— $S_H + S_E$ is positive definite. A necessary condition for $S_H + S_E$ to be positive definite is $v + q \geq p$. If S_E is positive definite, `RHTPE` avoids the computation of this generalized eigenvalue problem from scratch. In this case, the eigenvalues θ_i are obtained from λ_i by

$$\theta_i = \frac{\lambda_i}{1 + \lambda_i}$$

The fourth test statistic is as follows:

Pillai's trace

$$\begin{aligned} V &= \text{tr} \left[S_H (S_H + S_E)^{-1} \right] \\ &= \sum_{i=1}^p \theta_i \end{aligned}$$

V is output in `TEST(4)`. The p -value output in `TEST(8)` is based on an approximation discussed by Pillai (1985). The statistic

$$F = \frac{2n + s + 1}{2m + s + 1} \frac{V}{s - V}$$

has an approximate F distribution with $s(2m + s + 1)$ and $s(2n + s + 1)$ numerator and denominator degrees of freedom, respectively, where

$$\begin{aligned} s &= \min(p, q) \\ m &= \frac{1}{2}(|p - q| - 1) \\ n &= \frac{1}{2}(v - p - 1) \end{aligned}$$

The F test is exact if $\min(p, q) = 1$

RLOFE

Computes a lack of fit test based on exact replicates for a fitted regression model.

Required Arguments

X — NOBS by NCOL matrix containing the data. (Input)

IREP — Variable option. (Input)

IREP Meaning

- < 0 The first $-IREP$ columns of *X* contain the variables used to determine exact replicates.
- > 0 The *IREP* variables used to determine exact replicates are specified by the column numbers in *INDREP*.
- 0 The exact replicates are specified in *IGROUP*.

INDREP — Index vector of length *IREP* containing the column numbers of *X* that are the variables used to determine replication. (Input, if *IREP* is positive)
If *IREP* is less than or equal to 0, *INDREP* is not referenced and can be a vector of length one.

IRSP — Column number *IRSP* of *X* contains data for the response (dependent) variable. (Input)

DFE — Degrees of freedom for error from the fitted regression. (Input)

SSE — Sum of squares for error from the fitted regression. (Input)

IGROUP — Vector of length NOBS specifying group numbers. (Output, if *IREP* is nonzero; input, if *IREP* = 0)
On output, $IGROUP(I) = J$ means row *I* of *X* is in the *J*-th group of replicates ($J = 0, 1, 2, \dots, NGROUP$). Here, $J = 0$ indicates the group of observations not used in the analysis because NaN (not a number) was input for one of more of the values of the response, replication, frequency, or weight variables. On input, $IGROUP(I) = IGROUP(K)$, $K \neq I$, indicates that row *I* and row *K* of *X* are in the same group. *IGROUP(I)* must equal 0 if row *I* of *X* has NaN as one or more of the values of the response, replication, frequency, or weight variables.

NGROUP — Number number of groups in the lack of fit test. (Output)

TESTLF — Vector of length 10 containing statistics relating to the test for lack of fit of the model. (Output)

Elem. Description

1	Degrees of freedom for lack of fit
2	Degrees of freedom for pure error
3	Degrees of freedom for error (TESTLF(1)+ TESTLF(2))
4	Sum of squares for lack of fit
5	Sum of squares for pure error
6	Sum of squares for error
7	Mean square for lack of fit
8	Mean square for pure error
9	<i>F</i> statistic
10	<i>p</i> -value

If there are no replicates in the data set, a test for lack of fit cannot be performed. In this case, elements 8, 9, and 10 of TESTLF are set to NaN (not a number).

Optional Arguments

NOBS — Number of observations. (Input)
Default: NOBS = size (x,1).

NCOL — Number of columns in x. (Input)
Default: NCOL = size (x,2).

LDX — Leading dimension of x exactly as specified in the dimension statement in the calling program. (Input)
Default: LDX = size (x,1).

IFRQ — Frequency option. (Input)
IFRQ = 0 means that all frequencies are 1.0. For positive IFRQ, column number IFRQ of x contains the frequencies.
Default: IFRQ = 0.

IWT — Weighting option. (Input)
IWT = 0 means that all weights are 1.0. For positive IWT, column number IWT of x contains the weights.
Default: IWT = 0.

FORTRAN 90 Interface

Generic: CALL RLOFE (X, IREP, INDREP, IRSP, DFE, SSE, IGROUP, NGROUP, TESTLF [,...])

Specific: The specific interface names are S_RLOFE and D_RLOFE.

FORTRAN 77 Interface

Single: CALL RLOFE (NOBS, NCOL, X, LDX, IREP, INDREP, IRSP, IFRQ, IWT, DFE, SSE, IGROUP, NGROUP, TESTLF)

Double: The double precision name is DRLOFE.

Example 1

This example uses data from Draper and Smith (1981, page 374), which is input in X. A multiple linear regression of column 6 of X on an intercept and columns 1, 3, and 4 has already been computed. The fit gave a residual sum of squares $SSE = 163.93$ with $DFE = 16$ degrees of freedom. A test for lack of fit is computed using routine RLOFE.

```
USE RLOFE_INT
USE UMACH_INT
USE WRIRN_INT

INTEGER    LDX, NCOL, NOBS, NREP
PARAMETER (NCOL=6, NOBS=20, NREP=3, LDX=NOBS)
!
INTEGER    IGROUP(NOBS), INDREP(NREP), IREP, IRSP, &
           NGROUP, NOUT
REAL      DFE, SSE, TESTLF(10), X(LDX,NCOL)
!

DATA (X(1,J), J=1, 6) /1.0, 1.0, 1.0, 0.0, 1.0, 246.0/
DATA (X(2,J), J=1, 6) /1.0, 0.0, 1.0, 0.0, 1.0, 252.0/
DATA (X(3,J), J=1, 6) /1.0, 1.0, 1.0, 0.0, 1.0, 253.0/
DATA (X(4,J), J=1, 6) /0.0, 1.0, 1.0, 1.0, 0.0, 164.0/
DATA (X(5,J), J=1, 6) /1.0, 1.0, 0.0, 0.0, 1.0, 203.0/
DATA (X(6,J), J=1, 6) /0.0, 1.0, 1.0, 1.0, 0.0, 173.0/
DATA (X(7,J), J=1, 6) /1.0, 1.0, 0.0, 0.0, 1.0, 210.0/
DATA (X(8,J), J=1, 6) /1.0, 0.0, 1.0, 0.0, 1.0, 247.0/
DATA (X(9,J), J=1, 6) /0.0, 1.0, 0.0, 1.0, 0.0, 120.0/
DATA (X(10,J), J=1, 6) /0.0, 1.0, 1.0, 1.0, 0.0, 171.0/
DATA (X(11,J), J=1, 6) /0.0, 1.0, 1.0, 1.0, 0.0, 167.0/
DATA (X(12,J), J=1, 6) /0.0, 0.0, 1.0, 1.0, 0.0, 172.0/
DATA (X(13,J), J=1, 6) /1.0, 1.0, 1.0, 0.0, 1.0, 247.0/
DATA (X(14,J), J=1, 6) /1.0, 1.0, 1.0, 0.0, 1.0, 252.0/
DATA (X(15,J), J=1, 6) /1.0, 0.0, 1.0, 0.0, 1.0, 248.0/
DATA (X(16,J), J=1, 6) /0.0, 1.0, 1.0, 1.0, 0.0, 169.0/
DATA (X(17,J), J=1, 6) /0.0, 1.0, 0.0, 0.0, 0.0, 104.0/
DATA (X(18,J), J=1, 6) /0.0, 1.0, 1.0, 1.0, 0.0, 166.0/
DATA (X(19,J), J=1, 6) /0.0, 1.0, 1.0, 1.0, 0.0, 168.0/
DATA (X(20,J), J=1, 6) /0.0, 1.0, 1.0, 0.0, 0.0, 148.0/
DATA INDREP/1, 3, 4/
!
```

```

IREP = NREP
IRSP = 6
DFE = 16.0
SSE = 163.93
CALL RLOFE (X, IREP, INDREP, IRSP, DFE, SSE, IGROUP, NGROUP, TESTLF)
CALL UMACH (2, NOUT)
WRITE (NOUT,*) ' NGROUP = ', NGROUP
CALL WRIRN ('IGROUP', IGROUP, 1, NOBS, 1)
WRITE (NOUT,*) ' '
WRITE (NOUT,99999) '          Test for Lack of '// &
'Fit'
WRITE (NOUT,99999) '          Sum of      Mean '// &
'          Prob. of'
WRITE (NOUT,99999) ' Source of Error  DF  Squares  Square '// &
'          F Larger F'
WRITE (NOUT,99999) ' Lack of Fit      ', TESTLF(1), TESTLF(4), &
TESTLF(7), TESTLF(9), TESTLF(10)
WRITE (NOUT,99999) ' Expanded model ', TESTLF(2), TESTLF(5), &
TESTLF(8)
WRITE (NOUT,99999) ' Original model ', TESTLF(3), TESTLF(6)
99999 FORMAT (A, F5.1, F9.1, F8.2, F7.3, F10.3)
END

```

Output

NGROUP = 6

```

                                IGROUP
1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20
6  6  6  4  5  4  5  6  2  4  4  4  6  6  6  4  1  4  4  3

```

```

          Test for Lack of Fit
          Sum of      Mean          Prob. of
Source of Error  DF  Squares  Square      F Larger F
Lack of Fit      2.0    20.5    10.25    1.001  0.393
Expanded model   14.0   143.4    10.24
Original model   16.0   163.9

```

Comments

1. Workspace may be explicitly provided, if desired, by use of R2OFE/DR2OFE. The reference is:

```

CALL R2OFE (NOBS, NCOL, X, LDX, IREP, INDREP, IRSP, IFRQ, IWT,
DFE, SSE, IGROUP, NGROUP, TESTLF, IWK, WK)

```

The additional arguments are as follows:

IWK — Work vector. If IREP = 0, the length of IWK is 3 * NOBS; otherwise, the length of IWK is |IREP| + $m + 2.8854 * \ln(m) + 3 * NOBS + 5$.

WK — Work vector. If IREP = 0, WK is not referenced and can be a vector of length 1; otherwise, WK is of length 2 * m .

2. Informational errors

Type	Code	Description
3	1	DFE is less than the degrees of freedom for pure error. The degrees of freedom for lack of fit is set to zero.
3	2	SSE is less than the sum of squares for pure error. The sum of squares for lack of fit is set to zero.
4	3	An invalid weight or frequency is encountered. Weights and frequencies must be nonnegative.
4	4	An element in X contains NaN (not a number), but the corresponding element in IGROUP is not zero. When IREP = 0, missing values in a row of X are indicated by setting the corresponding row of IGROUP to zero.

Description

Routine RLOFE computes a lack of fit test based on exact replicates for a fitted regression model. The data need not be sorted prior to invoking RLOFE. The column indices of X for determining exact replicates can be input in INDREP. If the groups of exact replicates are known prior to invoking RLOFE, the option IREP = 0 allows RLOFE to bypass the computation of the groups. This option is particularly useful for computing a second lack of fit for a different dependent variable that uses the same columns of X for determining exact replicates as the first test.

If IREP is nonzero, routine SROWR (see Chapter 19, Utilities) is used to compute a permutation vector that specifies the sorted X along with the n_i 's, the number of rows of X in

each group. If IREP is zero, the permutation vector and the n_i 's are computed from IGROUP.

Let n_i be the number of rows of X in the i -th group of replicates ($i = 1, 2, \dots, k$). Let y_{ij} be the response for the j -th row within the i -th group. Let w_{ij} and f_{ij} be the associated weight and frequency, respectively. The pure error (within group) sum of squares is

$$SSPE = \sum_{i=1}^k \sum_{j=1}^{n_i} w_{ij} f_{ij} (y_{ij} - \bar{y}_{i\cdot})^2$$

The associated degrees of freedom are

$$DFPE = \left(\sum_{i=1}^k \sum_{j=1}^{n_i} f_{ij} \right) - k$$

The lack of fit sum of squares is $SSE - SSPE$ and the lack of fit degrees of freedom are $DFE - DFPE$.

The F statistic for the test of the null hypothesis of no lack of fit is

$$F = \frac{(SSE - SSPE)/(DFE - DFPE)}{SSPE/DFPE}$$

Under the hypothesis of no lack of fit, the computed F has an F distribution with numerator and denominator degrees of freedom $DFE - DFPE$ and $DFPE$, respectively. The p -value for the test

is computed as the probability that a random variable with this distribution is greater than or equal to the computed F statistic.

Additional Example

Example 2

This example uses the same data as in Example 1. Here, the option `IREP = 0` is used because `IGROUP` is known before invoking routine `RLOFE`. Routine `SROWR` (see Chapter 19, Utilities) is used to compute the group numbers contained in `IGROUP`.

```

USE RLOFE_INT
USE SROWR_INT
USE UMACH_INT
USE WRIRN_INT

INTEGER   LDX, NCOL, NKEY, NOBS
PARAMETER (NCOL=6, NKEY=3, NOBS=20, LDX=NOBS)
!
INTEGER   I, IGROUP(NOBS), INDKEY(NKEY), &
          INDREP(1), IPERM(NOBS), IREP, IRET, IRSP, &
          K, NGROUP, NI(NOBS), NOUT, NRMISS
REAL      DFE, SSE, TESTLF(10), X(LDX,NCOL)
!

DATA (X(1,J),J=1,6)/1.0, 1.0, 1.0, 0.0, 1.0, 246.0/
DATA (X(2,J),J=1,6)/1.0, 0.0, 1.0, 0.0, 1.0, 252.0/
DATA (X(3,J),J=1,6)/1.0, 1.0, 1.0, 0.0, 1.0, 253.0/
DATA (X(4,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 164.0/
DATA (X(5,J),J=1,6)/1.0, 1.0, 0.0, 0.0, 1.0, 203.0/
DATA (X(6,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 173.0/
DATA (X(7,J),J=1,6)/1.0, 1.0, 0.0, 0.0, 1.0, 210.0/
DATA (X(8,J),J=1,6)/1.0, 0.0, 1.0, 0.0, 1.0, 247.0/
DATA (X(9,J),J=1,6)/0.0, 1.0, 0.0, 1.0, 0.0, 120.0/
DATA (X(10,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 171.0/
DATA (X(11,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 167.0/
DATA (X(12,J),J=1,6)/0.0, 0.0, 1.0, 1.0, 0.0, 172.0/
DATA (X(13,J),J=1,6)/1.0, 1.0, 1.0, 0.0, 1.0, 247.0/
DATA (X(14,J),J=1,6)/1.0, 1.0, 1.0, 0.0, 1.0, 252.0/
DATA (X(15,J),J=1,6)/1.0, 0.0, 1.0, 0.0, 1.0, 248.0/
DATA (X(16,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 169.0/
DATA (X(17,J),J=1,6)/0.0, 1.0, 0.0, 0.0, 0.0, 104.0/
DATA (X(18,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 166.0/
DATA (X(19,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 168.0/
DATA (X(20,J),J=1,6)/0.0, 1.0, 1.0, 0.0, 0.0, 148.0/
DATA INDKEY/1, 3, 4/
!

IRET = 1
CALL SROWR (X, INDKEY, IPERM, NGROUP, NI, IRET=IRET)
K = 1
DO 20 I=1, NGROUP
  DO 10 J=1, NI(I)
    IGROUP(IPERM(K)) = I
    K = K + 1
10  CONTINUE
20  CONTINUE

```



```

IREP = 0
IRSP = 6
DFE = 16.0
SSE = 163.93
CALL RLOFE (X, IREP, INDREP, IRSP, DFE, SSE, IGROUP, NGROUP, TESTLF)
CALL UMACH (2, NOUT)
WRITE (NOUT,*) ' NGROUP = ', NGROUP
CALL WRIRN ('IGROUP', IGROUP, 1, NOBS, 1)
WRITE (NOUT,*) ' '
WRITE (NOUT,99999) '          Test for Lack of '// &
'Fit'
WRITE (NOUT,99999) '          Sum of      Mean '// &
'          Prob. of'
WRITE (NOUT,99999) ' Source of Error  DF  Squares  Square '// &
'          F Larger F'
WRITE (NOUT,99999) ' Lack of Fit      ', TESTLF(1), TESTLF(4), &
TESTLF(7), TESTLF(9), TESTLF(10)
WRITE (NOUT,99999) ' Expanded model ', TESTLF(2), TESTLF(5), &
TESTLF(8)
WRITE (NOUT,99999) ' Original model ', TESTLF(3), TESTLF(6)
99999 FORMAT (A, F5.1, F9.1, F8.2, F7.3, F10.3)
END

```

Output

NGROUP = 6

```

                                IGROUP
1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20
6  6  6  4  5  4  5  6  2  4  4  4  6  6  6  4  1  4  4  3

```

```

                                Test for Lack of Fit
                                Sum of      Mean          Prob. of
Source of Error  DF  Squares  Square      F  Larger F
Lack of Fit      2.0   20.5   10.25  1.001   0.393
Expanded model  14.0  143.4   10.24
Original model  16.0  163.9

```

RLOFN

Computes a lack of fit test based on near replicates for a fitted regression model.

Required Arguments

X— NOBS by NCOL matrix containing the data. (Input)

IIND — Independent variable option. (Input)

IIND Meaning

< 0 The first -IIND columns of X contain the independent (explanatory) variables.

> 0 The `IIND` independent variables are specified by the column numbers in `INDIND`.

= 0 There are no independent variables.

There are $\text{NCOEF} = \text{INTCEP} + |\text{IIND}|$ regressors—the intercept (if `INTCEP` = 1) and the independent variables.

INDIND — Index vector of length `IIND` containing the column numbers of `X` that are the independent variables. (Input, if `IIND` is positive)
If `IIND` is nonnegative, `INDIND` is not referenced and can be a vector of length one.

IRSP — Column number `IRSP` of `X` contains data for the response (dependent) variable. (Input)

B — Vector of length `NCOEF` containing a least-squares solution

$$\hat{\beta}$$

for the regression coefficients. (Input)

R — `NCOEF` by `NCOEF` upper triangular matrix containing the R matrix. (Input)
The R matrix can come from a regression fit based on a QR decomposition of the matrix of regressors or based on a Cholesky factorization $R^T R$ of the matrix of sums of squares and crossproducts of the regressors. Elements to the right of a diagonal element of R that is zero must also be zero. A zero row indicates a nonfull rank model. For an R matrix that comes from a regression fit with linear equality restrictions on the parameters, each row of R corresponding to a restriction must have a corresponding diagonal element that is negative. The remaining rows of R must have positive diagonal elements. Only the upper triangle of R is referenced.

DFE — Degrees of freedom for error from the fitted regression. (Input)

SSE — Sum of squares for error from the fitted regression. (Input)

NGROUP — Number of groups. (Input)
A cluster analysis based on `NGROUP` groups is performed. A good choice for `NGROUP` is the number of groups of near replicates in the data set.

IGROUP — Vector of length `NOBS` specifying group numbers. (Input, if `ICLUST` = 0; output, if `ICLUST` ≥ 1)
 $\text{IGROUP}(I) = J$ means row I of `X` is in the J -th group of near replicates ($J = 0, 1, 2, \dots, \text{NGROUP}$). Here, $J = 0$ indicates the group of observations not used in the analysis because NaN (not a number) was input for one or more of the values of the response, independent, frequency, or weight variables.

TESTLF — Vector of length 10 containing statistics relating to the test for lack of fit of the model. (Output)

Elem. Description

- 1 Degrees of freedom for lack of fit.
- 2 Degrees of freedom for error from the expanded model (one-way analysis of covariance model using clusters of near replicates as the groups).
- 3 Degrees of freedom for error ($DFE = TESTLF(1) + TESTLF(2)$).
- 4 Sum of squares for lack of fit.
- 5 Sum of squares for error from the expanded model.
- 6 Sum of squares for error ($SSE = TESTLF(4) + TESTLF(5)$).
- 7 Mean square for lack of fit.
- 8 Mean square for error from the expanded model.
- 9 F statistic.
- 10 p -value.

Optional Arguments

NOBS — Number of observations. (Input)
Default: $NOBS = size(x,1)$.

NCOL — Number of columns in x . (Input)
Default: $NCOL = size(x,2)$.

LDX — Leading dimension of x exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDX = size(x,1)$.

INTCEP — Intercept option. (Input)
Default: $INTCEP = 1$.

INTCEP Action

- 0 An intercept is not in the model.
- 1 An intercept is in the model.

IFRQ — Frequency option. (Input)

IFRQ = 0 means that all frequencies are 1.0. For positive IFRQ, column number IFRQ of X contains the frequencies.

Default: IFRQ = 0.

IWT — Weighting option. (Input)

IWT = 0 means that all weights are 1.0. For positive IWT, column number IWT of X contains the weights.

Default: IWT = 0.

LDR — Leading dimension of R exactly as specified in the dimension statement in the calling program. (Input)

Default: LDR = size (R,1).

ICLUST — Clustering option. (Input)

Default: ICLUST = 1.

ICLUST Meaning

0 Cluster groups are input in IGROUP.

1 Cluster groups are obtained using Euclidean distance.

2 Cluster groups are obtained using Mahalanobis distance.

MAXIT — Maximum number of iterations for the cluster analysis to determine near replicates. (Input, if ICLUST is positive, otherwise, MAXIT is not referenced)

MAXIT = 30 is usually sufficient for convergence.

Default: MAXIT = 30.

TOL — Tolerance used in determining linear dependence for the one-way analysis of covariance model using clusters as the groups. (Input)

TOL = $\text{EPS}^{2/3}$ is a good choice. For RLOFN, EPS = AMACH(4). See documentation for AMACH (Reference Material).

Default: TOL = 2.4e-5 for single precision and 3.6d - 11 for double precision.

FORTRAN 90 Interface

Generic: CALL RLOFN (X, IIND, INDIND, IRSP, B, R, DFE, SSE, NGROUP, IGROUP, TESTLF [,...])

Specific: The specific interface names are S_RLOFN and D_RLOFN.

FORTRAN 77 Interface

Single: CALL RLOFN (NOBS, NCOL, X, LDX, INTCEP, IIND, INDIND, IRSP, IFRQ, IWT, B, R, LDR, DFE, SSE, ICLUST, MAXIT, TOL, NGROUP, IGROUP, TESTLF)

Double: The double precision name is DRLOFN.

Example 1

This example uses data from Draper and Smith (1981, page 374), which is input in X. A multiple linear regression of column 6 of X on an intercept and columns 1, 3, and 4 is computed using routine RGIWN (page 117). Tests for lack of fit are computed for choices of NGROUP equal to 4 and 6 using routine RLOFN. Note that for NGROUP equal to 6 the results are exactly the same as for routine RLOFE (page 189). (If there are exact replicates in the data and the number of clusters used by RLOFN equals the number of distinct cases of the independent variables, then RLOFN and RLOFE produce the same output.)

```
USE IMSL_LIBRARIES
INTEGER LDB, LDR, LDSCPE, LDX, NCOEF, NCOL, NDEP, &
        NIND, NOBS
PARAMETER (INTCEP=1, NCOL=6, NDEP=1, NIND=3, NOBS=20, &
        LDSCPE=NDEP, LDX=NOBS, NCOEF=INTCEP+NIND, LDB=NCOEF, &
        LDR=NCOEF)
!
INTEGER ICLUST, IDEP, IGROUP(NOBS), IIND, INDDEP(NDEP), &
        INDIND(NIND), IRSP, NGROUP, NOUT, NRMISS, NROW
REAL B(LDB,NDEP), DFE, R(LDR,NCOEF), SCPE(LDSCPE,NDEP), &
        SSE, TESTLF(10), X(LDX,NCOL)
!
DATA (X(1,J),J=1,6)/1.0, 1.0, 1.0, 0.0, 1.0, 246.0/
DATA (X(2,J),J=1,6)/1.0, 0.0, 1.0, 0.0, 1.0, 252.0/
DATA (X(3,J),J=1,6)/1.0, 1.0, 1.0, 0.0, 1.0, 253.0/
DATA (X(4,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 164.0/
DATA (X(5,J),J=1,6)/1.0, 1.0, 0.0, 0.0, 1.0, 203.0/
DATA (X(6,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 173.0/
DATA (X(7,J),J=1,6)/1.0, 1.0, 0.0, 0.0, 1.0, 210.0/
DATA (X(8,J),J=1,6)/1.0, 0.0, 1.0, 0.0, 1.0, 247.0/
DATA (X(9,J),J=1,6)/0.0, 1.0, 0.0, 1.0, 0.0, 120.0/
DATA (X(10,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 171.0/
DATA (X(11,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 167.0/
DATA (X(12,J),J=1,6)/0.0, 0.0, 1.0, 1.0, 0.0, 172.0/
DATA (X(13,J),J=1,6)/1.0, 1.0, 1.0, 0.0, 1.0, 247.0/
DATA (X(14,J),J=1,6)/1.0, 1.0, 1.0, 0.0, 1.0, 252.0/
DATA (X(15,J),J=1,6)/1.0, 0.0, 1.0, 0.0, 1.0, 248.0/
DATA (X(16,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 169.0/
DATA (X(17,J),J=1,6)/0.0, 1.0, 0.0, 0.0, 0.0, 104.0/
DATA (X(18,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 166.0/
DATA (X(19,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 168.0/
DATA (X(20,J),J=1,6)/0.0, 1.0, 1.0, 0.0, 0.0, 148.0/
DATA INDIND/1, 3, 4/, INDDEP/6/
!
NROW = NOBS
IIND = NIND
IDEP = NDEP
CALL RGIWN(X, IIND, INDIND, IDEP, INDDEP, B, R=R, DFE=DFE, SCPE=SCPE)
SSE = SCPE(1,1)
IRSP = 6
ICLUST = 2
DO 10 NGROUP=4, 6, 2
```

```

CALL RLOFN (X, IIND, INDIND, IRSP, B(1:, 1), R, DFE, SSE, NGROUP, &
           IGROUP, TESTLF, ICLUST=ICLUST)
CALL UMACH (2, NOUT)
WRITE (NOUT,*) ' '
WRITE (NOUT,*) 'NGROUP = ', NGROUP
CALL WRIRN ('IGROUP', IGROUP, 1, NOBS, 1)
WRITE (NOUT,*) ' '
WRITE (NOUT,99999) '                               Test for Lack of '// &
                  'Fit'
WRITE (NOUT,99999) '                               Sum of    Mean '// &
                  '                               Prob. of'
WRITE (NOUT,99999) ' Source of Error   DF Squares Square '// &
                  '   F Larger F'
WRITE (NOUT,99999) ' Lack of Fit      ', TESTLF(1), TESTLF(4), &
                  TESTLF(7), TESTLF(9), TESTLF(10)
WRITE (NOUT,99999) ' Expanded model ', TESTLF(2), TESTLF(5), &
                  TESTLF(8)
WRITE (NOUT,99999) ' Original model ', TESTLF(3), TESTLF(6)
10 CONTINUE
99999 FORMAT (A, F5.1, F9.1, F8.2, F7.3, F10.3)
END

```

Output

NGROUP = 4

																				IGROUP			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20				
4	4	4	4	2	4	2	4	2	4	4	4	4	4	4	4	1	4	4	3				

						Test for Lack of Fit				
		DF	Sum of Squares	Mean Square	F	Prob. of Larger F				
Lack of Fit		1.0	0.4	0.38	0.035	0.855				
Expanded model		15.0	163.6	10.90						
Original model		16.0	163.9							

NGROUP = 6

																				IGROUP			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20				
6	6	6	4	5	4	5	6	2	4	4	4	6	6	6	4	1	4	4	3				

						Test for Lack of Fit				
		DF	Sum of Squares	Mean Square	F	Prob. of Larger F				
Lack of Fit		2.0	20.5	10.25	1.001	0.393				
Expanded model		14.0	143.4	10.24						
Original model		16.0	163.9							

Comments

1. Workspace may be explicitly provided, if desired, by use of R2OFN/DR2OFN. The reference is:

```
CALL R2OFN (NOBS, NCOL, X, LDX, INTCEP, IIND, INDIND, IRSP, FRQ,
IWT, B, R, LDR, DFE, SSE, ICLUST, MAXIT, TOL, NGROUP, IGROUP,
TESTLF, IWK, WK)
```

The additional arguments are as follows:

IWK — Work array of length $3 * \text{NOBS} + |\text{IIND}| + \text{NGROUP} + 3 + \max\{m + 2.8854 * \ln(m) + 2, 3 * \text{NGROUP}, \text{NCOEF}\}$, if **ICLUST** is positive. If **ICLUST** = 0, **IWK** can be an array of length 1.

WK — Work array of length **LWK**.

2. Informational errors

Type	Code	Description
3	1	Convergence did not occur in the cluster analysis for the lack of fit test within MAXIT iterations. Better results may be obtained by increasing MAXIT .
4	2	An invalid weight or frequency is encountered. Weights and frequencies must be nonnegative.
3	3	The matrix of sum of squares and crossproducts computed for the within cluster model for testing lack of fit is not nonnegative definite within the tolerance defined by TOL .
4	4	At least one element in the columns containing the independent variables, IRSP , IFRQ , or IWT of X contains NaN (not a number), but the corresponding element in IGROUP is not zero. When ICLUST = 0, missing values in a row of X are indicated by setting the corresponding row of IGROUP to zero.

Description

Routine **RLOFN** computes a lack of fit test based on near replicates for a fitted regression model. The data need not be sorted prior to invoking **RLOFN**. The column indices of **X** for determining near replicates must correspond to the independent variables in the original fitted model. If the groups of near replicates are known prior to invoking **RLOFN**, the option **ICLUST** = 0 allows **RLOFN** to bypass the computation of the groups.

The data can contain missing values indicated by NaN. (NaN is **AMACH**(6). Routine **AMACH** is described in the section “Machine-Dependent Constants” in the Reference Material.) For **ICLUST** equal to 1 or 2, any row of **X** containing NaN as a value for the response, weight, frequency, or independent variables is omitted from the analysis. For **ICLUST** equal to 0, if the *i*-th row of **X** contains NaN for one of the variables in the analysis, the *i*-th element of **IGROUP** must be 0 on input.

Routine **KMEAN** (see Chapter 11, Cluster Analysis) is used to compute *k* clusters or groups of near replicates. Prior to invoking **KMEAN**, a detached sort of the independent variables in the regression model is performed using routine **SROWR** (See Chapter 19, Utilities). If there are fewer than **NGROUP** distinct observations, a warning message is issued and *k* is set equal to the number of distinct observations. Otherwise, *k* equals **NGROUP**. For purposes of the cluster analysis, **ICLUST** = 1 specifies Euclidean distance and **ICLUST** = 2 specifies Mahalanobis

distance. For Mahalanobis distance, the data are transformed before invoking `KMEAN` so that the Euclidean metric applied by `KMEAN` for the transformed data is equivalent to the sample Mahalanobis distance for the original (untransformed) data.

Let X be the $n \times p$ matrix of regressors, and let R be the upper triangular matrix computed from the fitted regression model. The matrix R can be computed by routines `RGLM` (page 128), `RGINV` (page 117), or `RLEQU` (page 142) for fitting the regression model. A linear equality restriction on the regression parameters corresponds to a row of R with a negative diagonal element. Let D be a $p \times p$ diagonal matrix with diagonal elements

$$d_{ii} = \begin{cases} 1 & \text{if } r_{ii} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Let

$$x_i^T$$

be the i -th row of X , and let $t_i = Ds_i$ where s_i satisfies

$$R^T s_i = x_i$$

Then, the Mahalanobis distance from x_i to x_j equals the Euclidean distance from t_i to t_j because

$$\begin{aligned} \sigma^{-2} \text{Var} \left[(x_i - x_j)^T \hat{\beta} \right] &= \sigma^{-2} \text{Var} \left[(s_i - s_j)^T R \hat{\beta} \right] \\ &= \sigma^{-2} (s_i - s_j)^T \text{Var} (R \hat{\beta}) (s_i - s_j) \\ &= (s_i - s_j)^T D (s_i - s_j) \\ &= (t_i - t_j)^T (t_i - t_j) \end{aligned}$$

Once the clusters are identified by `KMEAN` an expanded regression model—a one-way analysis of covariance model—is fitted to the original (untransformed) data. Denote the original model by $y = X\beta + \varepsilon$ and the expanded model by $y = X\beta + Z\gamma + \varepsilon$. The added regressors that are contained in the $n \times k$ matrix Z in the expanded model are indicator variables specifying cluster membership. The lack of fit test that is computed is an exact test of the hypothesis that $\gamma = 0$ in the expanded model. This test was proposed as a lack of fit test by Christensen (1989).

Let $\text{SSE}(X, Z)$ be the error sum of squares from the fit of the expanded model and let $\text{SSE}(X)$ be the error sum of squares from the fit of the original model. The lack of fit sum of squares is $\text{SSE}(X) - \text{SSE}(X, Z)$ and the lack of fit degrees of freedom are $\text{DFE}(X) - \text{DFE}(X, Z)$. The F statistic for the test of the null hypothesis of no lack of fit is

$$F = \frac{(\text{SSE}(X) - \text{SSE}(X, Z)) / (\text{DFE}(X) - \text{DFE}(X, Z))}{\text{SSE}(X, Z) / \text{DFE}(X, Z)}$$

Under the hypothesis of no lack of fit, the computed F has an F distribution with numerator and denominator degrees of freedom $\text{DFE}(X) - \text{DFE}(X, Z)$ and $\text{DFE}(X, Z)$, respectively. The p -value for the test is computed as the probability that a random variable with this distribution is greater than or equal to the computed F statistic.

The error degrees of freedom and error sum of squares from the fit of the expanded model are computed as the error degrees of freedom and sum of squares from the reduced model where Z and y have been adjusted for X . Routine `RCOV` (page 113) is used to fit the reduced model. Let e be the vector of residuals from the original fitted model, let W be the diagonal matrix whose i -th diagonal element is the product of the weight and frequency for the i -th observation. The sum of squares and crossproducts matrix for the adjusted Z and y in the reduced model, which is input into `RCOV`, is

$$\begin{bmatrix} Z^T W Z - A^T A & Z^T W e \\ & e^T W e \end{bmatrix}$$

where A is a solution of $R^T A = D X^T W Z$.

Additional Example

Example 2

This example uses the same data and model from Example 1. Here, the option `ICLUST = 0` is input so that the group numbers for performing the lack of fit test are input.

```

USE IMSL_LIBRARIES
INTEGER    LDB, LDR, LDSCPE, LDX, NCOEF, NCOL, NDEP, &
           NIND, NOBS
PARAMETER  (INTCEP=1, NCOL=6, NDEP=1, NIND=3, NOBS=20, &
           LDSCPE=NDEP, LDX=NOBS, NCOEF=INTCEP+NIND, LDB=NCOEF, &
           LDR=NCOEF)
!
INTEGER    ICLUST, IDEP, IGROUP(NOBS), IIND, &
           INDDEP(NDEP), INDIND(NIND), IRSP, &
           NGROUP, NOUT
REAL      B(LDB,NDEP), DFE, R(LDR,NCOEF), SCPE(LDSCPE,NDEP), &
           SSE, TESTLF(10), TOL, X(LDX,NCOL), &
           XMAX(NCOEF), XMIN(NCOEF)
!
DATA (X(1,J),J=1,6)/1.0, 1.0, 1.0, 0.0, 1.0, 246.0/
DATA (X(2,J),J=1,6)/1.0, 0.0, 1.0, 0.0, 1.0, 252.0/
DATA (X(3,J),J=1,6)/1.0, 1.0, 1.0, 0.0, 1.0, 253.0/
DATA (X(4,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 164.0/
DATA (X(5,J),J=1,6)/1.0, 1.0, 0.0, 0.0, 1.0, 203.0/
DATA (X(6,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 173.0/
DATA (X(7,J),J=1,6)/1.0, 1.0, 0.0, 0.0, 1.0, 210.0/
DATA (X(8,J),J=1,6)/1.0, 0.0, 1.0, 0.0, 1.0, 247.0/
DATA (X(9,J),J=1,6)/0.0, 1.0, 0.0, 1.0, 0.0, 120.0/
DATA (X(10,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 171.0/
DATA (X(11,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 167.0/
DATA (X(12,J),J=1,6)/0.0, 0.0, 1.0, 1.0, 0.0, 172.0/
DATA (X(13,J),J=1,6)/1.0, 1.0, 1.0, 0.0, 1.0, 247.0/
DATA (X(14,J),J=1,6)/1.0, 1.0, 1.0, 0.0, 1.0, 252.0/
DATA (X(15,J),J=1,6)/1.0, 0.0, 1.0, 0.0, 1.0, 248.0/
DATA (X(16,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 169.0/
DATA (X(17,J),J=1,6)/0.0, 1.0, 0.0, 0.0, 0.0, 104.0/
DATA (X(18,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 166.0/
DATA (X(19,J),J=1,6)/0.0, 1.0, 1.0, 1.0, 0.0, 168.0/

```

```

DATA (X(20,J),J=1,6)/0.0, 1.0, 1.0, 0.0, 0.0, 148.0/
DATA INDIND/1, 3, 4/, INDDEP/6/
DATA IGROUP/4*4, 2, 4, 2, 4, 2, 7*4, 1, 2*4, 3/
!
IIND = NIND
IIEP = NDEP
CALL RGIVN (X, IIND, INDIND, IIEP, INDDEP, B, R=R, DFE=DFE, SCPE=SCPE)
SSE = SCPE(1,1)
IRSP = 6
ICLUST = 0
NGROUP = 4
CALL RLOFN (X, IIND, INDIND, IRSP, B(1:, 1), R, DFE, SSE, NGROUP, &
            IGROUP, TESTLF, ICLUST=ICLUST)
CALL UMACH (2, NOUT)
WRITE (NOUT,*) ' '
WRITE (NOUT,*) 'NGROUP = ', NGROUP
CALL WRIRN ('IGROUP', IGROUP, 1, NOBS, 1)
WRITE (NOUT,*) ' '
WRITE (NOUT,99999) ' Test for Lack of ' // &
                  'Fit'
WRITE (NOUT,99999) ' Sum of Mean ' // &
                  ' Prob. of'
WRITE (NOUT,99999) ' Source of Error DF Squares Square ' // &
                  ' F Larger F'
WRITE (NOUT,99999) ' Lack of Fit ', TESTLF(1), TESTLF(4), &
                  TESTLF(7), TESTLF(9), TESTLF(10)
WRITE (NOUT,99999) ' Expanded model ', TESTLF(2), TESTLF(5), &
                  TESTLF(8)
WRITE (NOUT,99999) ' Original model ', TESTLF(3), TESTLF(6)
99999 FORMAT (A, F5.1, F9.1, F8.2, F7.3, F10.3)
END

```

Output

NGROUP = 4

										IGROUP									
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	4	4	4	2	4	2	4	2	4	4	4	4	4	4	4	1	4	4	3

Test for Lack of Fit						
	DF	Sum of Squares	Mean Square	F	Prob. of Larger F	
Lack of Fit	1.0	0.4	0.38	0.035	0.855	
Expanded model	15.0	163.6	10.90			
Original model	16.0	163.9				

RCASE

Computes case statistics and diagnostics given data points, coefficient estimates

$$\hat{\beta}$$

and the R matrix for a fitted general linear model.

Required Arguments

X — NRX by NCOL matrix containing the data. (Input)

IRSP — Column number IRSP of *X* contains the data for the response (dependent) variable. (Input)

B — Vector of length NCOEF containing a least-squares solution

$$\hat{\beta}$$

for the regression coefficients. (Input)

R — NCOEF by NCOEF upper triangular matrix containing the *R* matrix. (Input)

The *R* matrix can come from a regression fit based on a *QR* decomposition of the matrix of regressors or based on a Cholesky factorization $R^T R$ of the matrix of sums of squares and crossproducts of the regressors. Elements to the right of a diagonal element of *R* that is zero must also be zero. A zero row indicates a nonfull rank model. For an *R* matrix that comes from a regression fit with linear equality restrictions on the parameters, each row of *R* corresponding to a restriction must have a corresponding diagonal element that is negative. The remaining rows of *R* must have positive diagonal elements. Only the upper triangle of *R* is referenced.

DFE — Degrees of freedom for error. (Input)

SSE — Sum of squares for error. (Input)

CASE — NRX by 12 matrix containing the case statistics. (Output)
Columns 1 through 12 contain the following:

Col.	Description
------	-------------

1	Observed response
2	Predicted response
3	Residual
4	Leverage
5	Standardized residual
6	Jackknife residual
7	Cook's distance
8	DFFITs

9, 10 Confidence interval on the mean

11, 12 Prediction interval

Optional Arguments

IDO — Processing option. (Input)

Default: `IDO = 0`.

IDO Action

0 This is the only invocation of `RCASE` for this data set, and all the data are input at once.

1 This is the first invocation, and additional calls to `RCASE` will be made. Case statistics are computed for the data in `X`.

2 This is an intermediate or final invocation of `RCASE`. Case statistics are computed for the data in `X`.

NRX — Number of rows in `X`. (Input)

Default: `NRX = size(X,1)`.

NCOL — Number of columns in `X`. (Input)

Default: `NCOL = size(X,2)`.

LDX — Leading dimension of `X` exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDX = size(X,1)`.

INTCEP — Intercept option. (Input)

Default: `INTCEP = 1`.

INTCEP Action

0 An intercept is not in the model.

1 An intercept is in the model.

IEF — Effect option. (Input)

Default: `IEF = -1`.

The absolute value of `IEF` is the number of effects (sources of variation) due to the model. The sign of `IEF` specifies the following options.

IEF Meaning

- < 0 Each effect corresponds to a single regressor (coefficient) in the model. In this case, arguments *NCLVAR*, *INDCL*, *NCLVAL*, *CLVAL*, *NVEF*, *INDEF*, and *IDUMMY* are not referenced.
- > 0 Each effect corresponds to one or more regressors. A general linear model is specified through the arguments *NCLVAR*, *INDCL*, *NCLVAL*, *CLVAL*, *NVEF*, *INDEF*, and *IDUMMY*.
- 0 There are no effects in the model. *INTCEP* must equal 1.

NCLVAR — Number of classification variables. (Input, if *IEF* is positive)

INDCL — Index vector of length *NCLVAR* containing the column numbers of *X* that are the classification variables. (Input, if *IEF* is positive)

NCLVAL — Vector of length *NCLVAR* containing the number of values taken on by each classification variable. (Input, if *IEF* is positive)
NCLVAL(I) is the number of distinct values for the *I*-th classification variable.

CLVAL — Vector of length *NCLVAL(1) + NCLVAL(2) + ... + NCLVAL(NCLVAR)* containing the values of the classification variables. (Input, if *IEF* is positive)
The first *NCLVAL(1)* variables contain the values of the first classification variable. The next *NCLVAL(2)* variables contain the values of the second classification variable. ... The last *NCLVAL(NCLVAR)* variables contain the values of the last classification variable.

NVEF — Vector of length *IEF* containing the number of variables associated with each effect in the model. (Input, if *IEF* is positive)

INDEF — Index vector of length *NVEF(1) + NVEF(2) + ... + NVEF(IEF)*. (Input, if *IEF* is positive)
The first *NVEF(1)* elements give the column numbers of *X* for each variable in the first effect. The next *NVEF(2)* elements give the column numbers for each variable in the second effect. ... The last *NVEF(NEF)* elements give the column numbers for each variable in the last effect.

IDUMMY — Dummy variable option. (Input, if *IEF* is positive)
Some indicator variables are defined for the *I*-th class variable as follows: Let $J = NCLVAL(1) + NCLVAL(2) + \dots + NCLVAL(I - 1)$. *NCLVAL(I)* indicator variables are defined such that for $K = 1, 2, \dots, NCLVAL(I)$ the *K*-th indicator variable for row *M* of *X* takes the value 1.0 if $X(M, INDCL(I)) = CLVAL(J + K)$ and equals 0.0 otherwise. Dummy variables are generated from these indicator variables in one of the three following ways:

IDUMMY Method

- 0, 1 The $NCLVAL(I)$ indicator variables are the dummy variables (In *RCASE*, the computations for $IDUMMY = 0$ and $IDUMMY = 1$ are the same. The two values 0 and 1 are provided so that *RCASE* can be called after routine *RGLM* ([page 128](#)) with no change in *IDUMMY*.)
- 2 The first $NCLVAL(I) - 1$ indicator variables are the dummy variables. The last indicator variable is omitted.
- 3 The K -th indicator variable minus the $NCLVAL(I)$ -th indicator variable is the K -th dummy variable ($K = 1, 2, \dots, NCLVAL(I) - 1$).

IWT — Weighting option. (Input)

$IWT = 0$ means that all weights are 1.0. For positive *IWT*, column number *IWT* of *X* contains the weights, and the computed prediction interval uses $SSE/(DFE * X(I, IWT))$ for the estimated variance of a future response.

Default: $IWT = 0$.

IPRED — Prediction interval option. (Input)

$IPRED = 0$ means that prediction intervals are desired for a single future response. For positive *IPRED*, column number *IPRED* of *X* contains the number of future responses for which a prediction interval is desired on the average of the future responses.

Default: $IPRED = 0$.

CONPCM — Confidence level for two-sided interval estimates on the mean, in percent.

(Input)

CONPCM percent confidence intervals are computed, hence, *CONPCM* must be greater than or equal to 0.0 and less than 100.0. *CONPCM* often will be 90.0, 95.0, or 99.0. For one-sided intervals with confidence level *ONECL*, where *ONECL* is greater than or equal to 50.0 and less than 100.0, set $CONPCM = 100.0 - 2.0 * (100.0 - ONECL)$.

Default: $CONPCM = 95.0$.

CONPCP — Confidence level for two-sided prediction intervals, in percent. (Input)

CONPCP percent prediction intervals are computed, hence, *CONPCP* must be greater than or equal to 0.0 and less than 100.0. *CONPCP* often will be 90.0, 95.0, or 99.0. For one-sided intervals with confidence level *ONECL*, where *ONECL* is greater than or equal to 50.0 and less than 100.0, set $CONPCP = 100.0 - 2.0 * (100.0 - ONECL)$.

Default: $CONPCP = 95.0$.

PRINT — Printing option. (Input)

Default: $PRINT = 'N'$.

PRINT is a character string indicating what is to be printed. The *PRINT* string is composed of one-character print codes to control printing. These print codes are given as follows:

PRINT (I : I)	Printing that Occurs
'A'	All
'N'	None
'1'	Observed response
'2'	Predicted response
'3'	Residual
'4'	Leverage
'5'	Standardized residual
'6'	Jackknife residual
'7'	Cook's distance
'8'	DFBETS
'M'	Confidence interval on the mean
'P'	Prediction interval
'X'	Influential cases (unusual "x-value")
'Y'	Outlier cases (unusual "y-value")

The concatenated print codes 'A', 'N', '1', ..., 'P' that comprise the PRINT string give the combination of statistics to be printed. Concatenation of these codes with print codes 'X' or 'Y' restricts printing to cases determined to be influential or outliers. Here are a few examples.

PRINT Printing Action

- 'A' All.
- 'N' None.
- '46' Leverage and jackknife residual for all cases.
- 'AXY' All statistics are printed for cases that are highly influential or are outliers.
- '46XY' Leverage and jackknife residual are printed for cases that are highly influential or are outliers.

IOBS — Number of the observation corresponding to the first row of x . (Input)
This observation number is used only for printing the row labels for the individual case statistics.

Default: `IOBS = size (x,1)`.

NCOEF — Number of regression coefficients in the model. (Input)

Default: `NCOEF = size (B,1)`.

LDR — Leading dimension of R exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDR = size (R,1)`.

LDCASE — Leading dimension of $CASE$ exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDCASE = size (CASE,1)`.

NRMISS — Number of rows of $CASE$ containing NaN (not a number). (Output)

If any row of data contains NaN as a value of a variable other than the response variable, columns 3 through 12 of the corresponding row in $CASE$ are set to NaN. If the response is missing, columns 1, 3, and 5 through 8 are set to NaN.

FORTRAN 90 Interface

Generic: `CALL RCASE (X, IRSP, B, R, DFE, SSE, CASE [, ...])`

Specific: The specific interface names are `S_RCASE` and `D_RCASE`.

FORTRAN 77 Interface

Single: `CALL RCASE (IDO, NRX, NCOL, X, LDX, INTCEP, IEF, NCLVAR, INDCL, NCLVAL, CLVAL, NVEF, INDEF, IDUMMY, IRSP, IWT, IPRED, CONPCM, CONPCP, PRINT, IOBS, NCOEF, B, R, LDR, DFE, SSE, CASE, LDCASE, NRMISS)`

Double: The double precision name is `DRCASE`.

Example 1

A multiple linear regression model is fitted and case statistics computed for data discussed by Cook and Weisberg (1982, page 103). The fitted model is

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2$$

Some of the statistics in row 6 of the output matrix $CASE$ are undefined (0.0/0.0) and are set to NaN (not a number). Some statistics in row 4 of $CASE$ are set to Inf (positive machine infinity). The values of NaN and positive machine infinity can be retrieved by routine `AMACH` (or `DMACH` when using double precision), which is documented in the section “Machine-Dependent Constants” in Reference Material.


```

USE RCASE_INT
USE RGIVN_INT
INTEGER INTCEP, LDB, LDCASE, LDR, LDSCPE, LDX, NCOEF, NCOL, &
NDEP, NIND, NROW
PARAMETER (INTCEP=1, NCOL=3, NDEP=1, NIND=2, NROW=7, &
LDCASE=NROW, LDSCPE=NDEP, LDX=NROW, &
NCOEF=INTCEP+NIND, LDB=NCOEF, LDR=NCOEF)
!
INTEGER IDEP, IEF, IIND, INDDEP(1), INDIND(1), IOBS, IRSP
REAL B(LDB,NDEP), CASE(LDCASE,12), DFE, R(LDR,NCOEF), &
SCPE(LDSCPE,NDEP), SSE, X(LDX,NCOL)
CHARACTER PRINT*1
!
DATA (X(1,J),J=1,NIND+NDEP) /1.0, 1.0, 3.0/
DATA (X(2,J),J=1,NIND+NDEP) /1.0, 2.0, 4.0/
DATA (X(3,J),J=1,NIND+NDEP) /1.0, 3.0, 5.0/
DATA (X(4,J),J=1,NIND+NDEP) /1.0, 4.0, 7.0/
DATA (X(5,J),J=1,NIND+NDEP) /1.0, 5.0, 7.0/
DATA (X(6,J),J=1,NIND+NDEP) /0.0, 6.0, 8.0/
DATA (X(7,J),J=1,NIND+NDEP) /1.0, 7.0, 9.0/
!
IIND = -NIND
IDEP = -NDEP
CALL RGIVN (X, IIND, INDIND, IDEP, INDDEP, B, R=R, DFE=DFE, SCPE=SCPE)
IEF = -NIND
IRSP = NCOL
PRINT = 'A'
IOBS = 1
SSE = SCPE(1,1)
CALL RCASE (X, IRSP, B(1:,1), R, DFE, SSE, CASE, IEF=IEF, &
PRINT=PRINT, IOBS=IOBS, NCOEF=NCOEF, NRMISS=NRMISS)
!
END

```

Output

```

* * * Case Analysis * * *

```

Obs.	Observed	Predicted	Residual	Leverage	Std. Res.	Jack Res.
	Cook's D	DFFITs	95.0% CI	95.0% CI	95.0% PI	95.0% PI
1	3.0000	3.1286	-0.1286	0.4714	-0.3886	-0.3430
	0.0449	-0.3240	2.2609	3.9962	1.5957	4.6614
2	4.0000	4.1429	-0.1429	0.2857	-0.3714	-0.3273
	0.0184	-0.2070	3.4674	4.8183	2.7100	5.5757
3	5.0000	5.1571	-0.1571	0.1857	-0.3826	-0.3376
	0.0111	-0.1612	4.6126	5.7017	3.7812	6.5331
Y	4	7.0000	6.1714	0.8286	0.1714	2.0000
	0.2759	Inf	5.6482	6.6946	4.8038	7.5391
	5	7.0000	7.1857	-0.1857	0.2429	-0.4689
	0.0235	-0.2366	6.5630	7.8084	5.7770	8.5945
X	6	8.0000	8.0000	0.0000	1.0000	NaN
	NaN	NaN	6.7364	9.2636	6.2129	9.7871
	7	9.0000	9.2143	-0.2143	0.6429	-0.7878
	0.3724	-0.9959	8.2011	10.2275	7.5946	10.8339

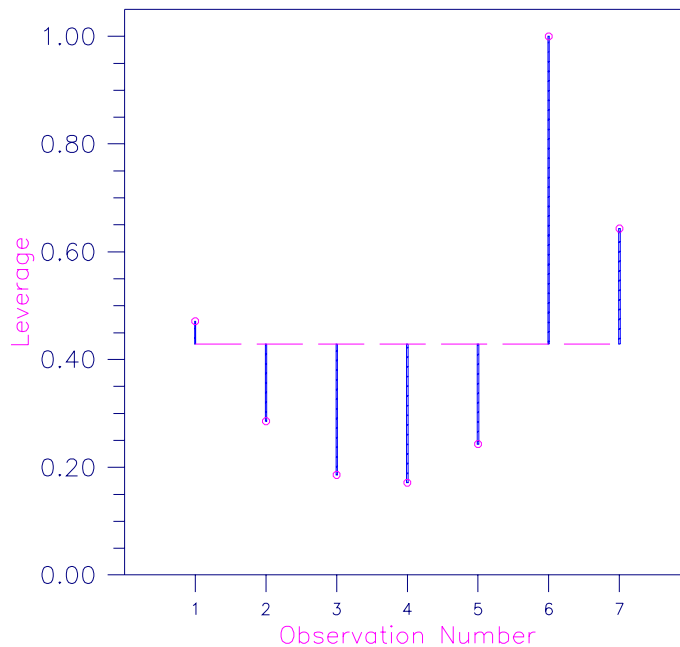


Figure 2-6 Plot of Leverages h_i and the Average ($p/n = 3/7$)

Comments

1. Workspace may be explicitly provided, if desired, by use of R2ASE/DR2ASE. The reference is:

```
CALL R2ASE (IDO, NRX, NCOL, X, LDX, INTCEP, IEF, NCLVAR, INDCL,
NCLVAL, CLVAL, NVEF, INDEF, IDUMMY, IRSP, IWT, IPRED, CONPCM,
CONPCP, PRINT, IOBS, NCOEF, B, R, LDR, DFE, SSE, CASE, LDCASE,
NRMISS, WK)
```

The additional argument is:

WK — Work vector of length $NCOEF + 1$.

2. Informational errors

Type Code

4	1	A weight is negative. Weights must be nonnegative.
3	2	The linear combination of the regression coefficients specified is not estimable within the preset tolerance.
3	3	A leverage much greater than 1.0 was computed. It is set to 1.0.
3	4	A deleted residual mean square much less than 0.0 was computed. It is set to 0.0.
4	5	A number of future observations for the prediction interval is nonpositive. It must be positive.

Description

The general linear model used by routine RCASE is

$$y = X\beta + \varepsilon$$

where y is the $n \times 1$ vector of responses, X is the $n \times p$ matrix of regressors, β is the $p \times 1$ vector of regression coefficients, and ε is the $n \times 1$ vector of errors whose elements are independently normally distributed with mean 0 and variance σ^2/w_i . The model used by RCASE also permits linear equality restrictions on β . From a general linear model fitted using the w_i 's as the weights, routine RCASE computes confidence intervals and statistics for the individual cases that constitute the data set. Let x_i be a column vector containing elements of the i -th row of X . Let $W = \text{diag}(w_1, w_2, \dots, w_n)$. The leverage is defined as

$$h_i = x_i^T (X^T W X)^{-1} x_i w_i$$

(In the case of linear equality restrictions on β , the leverage is defined in terms of the reduced model.) Put $D = \text{diag}(d_1, d_2, \dots, d_p)$ with $d_j = 1$ if the j -th diagonal element of R is positive and 0 otherwise. The leverage is computed as $h_i = (a^T D a) w_i$ where a is a solution to $R^T a = x_i$. The estimated variance of

$$\hat{y}_i = x_i^T \hat{\beta}$$

is given by $h_i s^2 / w_i$, where $s^2 = \text{SSE} / \text{DFE}$. The computation of the remainder of the case statistics follows easily from their definitions. See the chapter introduction ([page 83](#)) for definitions of the case diagnostics.

Often predicted values and confidence intervals are desired for combinations of settings of the effect variables not used in computing the regression fit. This can be accomplished using a single data matrix by including these settings of the variables as part of the data matrix and by setting the response equal to NaN (not a number). NaN can be retrieved by the invocation of the function AMACH(6). The regression routine performing the fit will omit the case, and RCASE will compute a predicted value and confidence interval for the missing response from the given settings of the effect variables.

The type 3 informational errors can occur if the input variables X , R , B and SSE are not consistent with each other or if excessive rounding has occurred in their computation. The type 3 error message with error code 2 arises when X contains a row not in the space spanned by the rows of R . An examination of the model that was fitted and the X for which diagnostics are to be computed is required in order to insure that only linear combinations of the regression coefficients that can be estimated from the fitted model are specified in X . For further details, see the discussion of estimable functions given by Maindonald (1984, pages 166–168) and Searle (1971, pages 180–188).

Additional Example

Example 2

A one-way analysis of covariance model is fitted to the turkey data discussed by Draper and Smith (1981, pages 243–249). The response variable is turkey weight y (in pounds). There are

three groups of turkeys corresponding to the three states where they were reared. The age of a turkey (in weeks) is the covariate. The explanatory variables are group, age, and interaction. The model is

$$y_{ij} = \mu + \alpha_i + \beta x_{ij} + \beta_i x_{ij} + \varepsilon_{ij} \quad i = 1, 2, 3; j = 1, 2, \dots, n_i$$

where $\alpha_3 = 0$ and $\beta_3 = 0$. Routine RGLM (page 128) is used to fit the model. The option IDUMMY = 2 is used. The fitted model gives three separate lines, one for each state where the turkeys were reared. Then, RCASE is used to compute case statistics from the fitted model.

```

USE RCASE_INT
USE RGLM_INT
USE AMACH_INT

INTEGER IDEP, IEF, INTCEP, LDB, LDCASE, LDR, LDSCPE, LDX, &
        MAXB, MAXCL, NCLVAR, NCOL, NROW
PARAMETER (IDEP=1, IEF=3, INTCEP=1, MAXB=6, MAXCL=3, NCLVAR=1, &
        NCOL=3, NROW=13, LDB=MAXB, LDCASE=NROW, LDR=MAXB, &
        LDSCPE=IDEP, LDX=NROW)
!
INTEGER IDUMMY, INDCL(NCLVAR), INDDEP(IDEP), &
        INDEF(4), IOBS, IRANK, IRBEF(IEF+1), IRSP, &
        NCLVAL(NCLVAR), NCOEF, NRMIS, NVEF(IEF)
REAL B(LDB, IDEP), CASE(LDCASE, 12), CLVAL(MAXCL), &
        DFE, R(LDR, MAXB), SCPE(LDSCPE, IDEP), SSE, X(LDX, NCOL)
CHARACTER PRINT
!
DATA (X(1, J), J=1, 3) /25.0, 13.8, 3.0/
DATA (X(2, J), J=1, 3) /28.0, 13.3, 1.0/
DATA (X(3, J), J=1, 3) /20.0, 8.9, 1.0/
DATA (X(4, J), J=1, 3) /32.0, 15.1, 1.0/
DATA (X(5, J), J=1, 3) /22.0, 10.4, 1.0/
DATA (X(6, J), J=1, 3) /29.0, 13.1, 2.0/
DATA (X(7, J), J=1, 3) /27.0, 12.4, 2.0/
DATA (X(8, J), J=1, 3) /28.0, 13.2, 2.0/
DATA (X(9, J), J=1, 3) /26.0, 11.8, 2.0/
DATA (X(10, J), J=1, 3) /21.0, 11.5, 3.0/
DATA (X(11, J), J=1, 3) /27.0, 14.2, 3.0/
DATA (X(12, J), J=1, 3) /29.0, 15.4, 3.0/
DATA (X(13, J), J=1, 3) /23.0, 13.1, 3.0/
DATA INDCL/3/, NVEF/1, 1, 2/, INDEF/3, 1, 1, 3/, INDDEP/2/
!
IDUMMY = 2
CALL RGLM (X, INDCL, NVEF, INDEF, IDEP, INDDEP, MAXCL, B, &
        IDUMMY=IDUMMY, NCLVAL=NCLVAL, CLVAL=CLVAL, IRBEF=IRBEF, &
        R=R, DFE=DFE, SCPE=SCPE)
!
PRINT = 'A'
IRSP = INDDEP(1)
IPRED = 0
PRINT = 'A'
IOBS = 1
NCOEF = IRBEF(IEF+1) - 1
SSE = SCPE(1, 1)
CALL RCASE (X, IRSP, B(1:, 1), R, DFE, SSE, CASE, IEF=IEF, &

```

```
NCLVAR=NCLVAR, INDCL=INDCL, NCLVAL=NCLVAL, CLVAL=CLVAL, &
NVEF=NVEF, INDEF=INDEF, IDUMMY=IDUMMY, IOBS=IOBS, &
PRINT=PRINT, NCOEF=NCOEF)
```

!

END

Output

```

* * * Case Analysis * * *
Obs.   Observed   Predicted   Residual   Leverage   Std. Res.   Jack Res.
      Cook's D     DFFITS     95.0% CI   95.0% CI   95.0% PI   95.0% PI
1     13.8000    13.6000     0.2000     0.2000     0.7040     0.6762
      0.0207     0.3381    13.2641    13.9359    12.7773    14.4227
2     13.3000    13.1901     0.1099     0.3187     0.4192     0.3930
      0.0137     0.2688    12.7661    13.6141    12.3276    14.0526
3     8.9000     9.1418    -0.2418     0.5824    -1.1779    -1.2178
      0.3225    -1.4383     8.5686     9.7149     8.1970    10.0865
4     15.1000    15.2143    -0.1143     0.7143    -0.6732    -0.6444
      0.1888    -1.0189    14.5795    15.8490    14.2309    16.1976
5     10.4000    10.1538     0.2462     0.3846     0.9879     0.9860
      0.1017     0.7795     9.6881    10.6196     9.2701    11.0376
6     13.1000    13.3300    -0.2300     0.7000    -1.3221    -1.4131
      0.6797    -2.1585    12.7016    13.9584    12.3507    14.3093
7     12.4000    12.3900     0.0100     0.3000     0.0376     0.0348
      0.0001     0.0228    11.9786    12.8014    11.5337    13.2463
8     13.2000    12.8600     0.3400     0.3000     1.2795     1.3533
      0.1169     0.8859    12.4486    13.2714    12.0037    13.7163
9     11.8000    11.9200    -0.1200     0.7000    -0.6898    -0.6615
      0.1850    -1.0104    11.2916    12.5484    10.9407    12.8993
10    11.5000    11.8200    -0.3200     0.6000    -1.5930    -1.8472
      0.6344    -2.2623    11.2382    12.4018    10.8700    12.7700
11    14.2000    14.4900    -0.2900     0.3000    -1.0913    -1.1091
      0.0851    -0.7261    14.0786    14.9014    13.6337    15.3463
12    15.4000    15.3800     0.0200     0.6000     0.0996     0.0922
      0.0025     0.1130    14.7982    15.9618    14.4300    16.3300
13    13.1000    12.7100     0.3900     0.3000     1.4676     1.6330
      0.1538     1.0691    12.2986    13.1214    11.8537    13.5663
```

ROTTIN

Computes diagnostics for detection of outliers and influential data points given residuals and the R matrix for a fitted general linear model.

Required Arguments

X— NRX by NCOL matrix containing the data. (Input)

IIND— Independent variable option. (Input)

The absolute value of IIND is the number of independent (explanatory) variables. The sign of IIND specifies the following options:

IIND Meaning

- < 0 The data for the $-IIND$ independent variables are given in the first $-IIND$ columns of X .
- > 0 The data for the $IIND$ independent variables are in the columns of X whose column numbers are given by the elements of $INDIND$.
- = 0 There are no independent variables.

The regressors are the constant regressor (if $INTCEP = 1$) and the independent variables.

INDIND — Index vector of length $IIND$ containing the column numbers of X that are the independent (explanatory) variables. (Input, if $IIND$ is positive)
If $IIND$ is nonpositive, $INDIND$ is not referenced and can be a vector of length one.

R — $INTCEP + |IIND|$ by $INTCEP + |IIND|$ upper triangular matrix containing the R matrix. (Input)

The R matrix can come from a regression fit based on a QR decomposition of the matrix of regressors or based on a Cholesky factorization $R^T R$ of the matrix of sums of squares and crossproducts of the regressors. Elements to the right of a diagonal element of R that is zero must also be zero. A zero row indicates a nonfull rank model. For an R matrix that comes from a regression fit with linear equality restrictions on the parameters, each row of R corresponding to a restriction must have a corresponding diagonal element that is negative. The remaining rows of R must have positive diagonal elements.

DFE — Degrees of freedom for error. (Input)

SSE — Sum of squares for error. (Input)

E — Vector of length NRX with the residuals. (Input)

If a residual is not known, e.g., the value for the dependent (response) variable was missing, the input value of the corresponding element of E should equal NaN (not a number).

OTIN — NRX by 6 matrix containing diagnostics for detection of outliers and influential cases. (Output)

The columns of $OTIN$ contain the following:

Col. Description

- 1 Residual
- 2 Leverage (diagonal element of the 'Hat' matrix)

Col.	Description
3	Standardized residual
4	Jackknife (deleted) residual
5	Cook's Distance
6	DFFITS

Optional Arguments

NRX — Number of rows of data. (Input)
 Default: $NRX = \text{size}(x,1)$.

NCOL — Number of columns in x . (Input)
 Default: $NCOL = \text{size}(x,2)$.

LDX — Leading dimension of x exactly as specified in the dimension statement in the calling program. (Input)
 Default: $LDX = \text{size}(x,1)$.

INTCEP — Intercept option. (Input)
 Default: $INTCEP = 1$.

INTCEP Action

- 0 An intercept is not in the model.
- 1 An intercept is in the model.

IWT — Weighting option. (Input)
 $IWT = 0$ means that all weights are 1.0. For positive IWT , column number IWT of x contains the weights.
 Default: $IWT = 0$.

LDR — Leading dimension of R exactly as specified in the dimension statement in the calling program. (Input)
 Default: $LDR = \text{size}(R,1)$.

LDOTIN — Leading dimension of $OTIN$ exactly as specified in the dimension statement in the calling program. (Input)
 Default: $LDOTIN = \text{size}(OTIN,1)$.

NRMIS — Number of rows of $OTIN$ containing NaN (not a number). (Output)
 If any row of data contains NaN as a value of the independent variable or weight, elements in columns 2 thru 6 of the corresponding row in $OTIN$ are set to NaN. If the residual is missing, elements in columns 3 thru 6 are set to NaN.

FORTRAN 90 Interface

Generic: CALL ROTIN (X, IIND, INDIND, R, DFE, SSE, E, OTIN [,...])

Specific: The specific interface names are S_ROTIN and D_ROTIN.

FORTRAN 77 Interface

Single: CALL ROTIN (NRX, NCOL, X, LDX, INTCEP, IIND, INDIND, IWT, R, LDR, DFE, SSE, E, OTIN, LDOTIN, NRMIS)

Double: The double precision name is DROTIN.

Example 1

A multiple linear regression model is fit and case statistics computed for data discussed by Cook and Weisberg (1982, page 103). The fitted model is

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2$$

Some of the statistics in row 6 of the output matrix OTIN are undefined (0.0/0.0) and are set to NaN (not a number). Some statistics in row 4 of OTIN are infinite and are set to machine infinity. The values of NaN and machine infinity can be retrieved by routine AMACH (or DMACH when using double precision), which is documented in Reference Material.

```
!                               SPECIFICATIONS FOR LOCAL VARIABLES
USE IMSL_LIBRARIES
INTEGER INTCEP, LDB, LDOTIN, LDR, LDSCPE, LDX, NCOEF, NCOL, &
        NDEP, NIND, NROW
PARAMETER (INTCEP=1, NCOL=3, NDEP=1, NIND=2, NROW=7, &
        LDOTIN=NROW, LDSCPE=NDEP, LDX=NROW, &
        NCOEF=INTCEP+NIND, LDB=NCOEF, LDR=NCOEF)
!
INTEGER I, IDEP, IIND, INDDEP(1), INDIND(1), NOUT, NRMIS
REAL B(LDB,NDEP), D(NCOEF), DFE, E(NROW), &
      OTIN(LDOTIN,6), R(LDR,NCOEF), SCPE(LDSCPE,NDEP), &
      SSE, X(LDX,NCOL), XMAX(NCOEF), XMIN(NCOEF)
CHARACTER CLABEL(7)*10, RLABEL(1)*6
!
DATA CLABEL/'Obs.', 'Residual', 'Leverage', 'Std. Res.', &
      'Jack. Res.', 'Cook's D', 'DFFITs'/
DATA RLABEL/'NUMBER'/
!
DATA (X(1,J),J=1,NIND+NDEP) /1.0, 1.0, 3.0/
DATA (X(2,J),J=1,NIND+NDEP) /1.0, 2.0, 4.0/
DATA (X(3,J),J=1,NIND+NDEP) /1.0, 3.0, 5.0/
DATA (X(4,J),J=1,NIND+NDEP) /1.0, 4.0, 7.0/
DATA (X(5,J),J=1,NIND+NDEP) /1.0, 5.0, 7.0/
DATA (X(6,J),J=1,NIND+NDEP) /0.0, 6.0, 8.0/
DATA (X(7,J),J=1,NIND+NDEP) /1.0, 7.0, 9.0/
!
IIND = -NIND
IDEP = -NDEP
```



```

CALL RGIWN (X, IIND, INDIND, IDEP, INDDEP, B, R=R, DFE=DFE, SCPE=SCPE)
SSE = SCPE(1,1)
!
!                               Compute residuals.
DO 10 I=1, NROW
  E(I) = X(I,NCOL) - B(1,1) - SDOT(NIND, B((INTCEP+1):,1), &
    1, X(I:, 1), LDX)
10 CONTINUE
!
CALL ROTIN (X, IIND, INDIND, R, DFE, SSE, E, OTIN, &
  NRMISS=NRMISS)
!
CALL WRRRL ('OTIN', OTIN, RLABEL, CLABEL, FMT='(F10.3)')
CALL UMACH (2, NOUT)
WRITE (NOUT,*) 'NRMISS = ', NRMISS
!
END

```

Output

Obs.	Residual	Leverage	Std. Res.	Jack. Res.	Cook's D	DFFITs
1	-0.129	0.471	-0.389	-0.343	0.045	-0.324
2	-0.143	0.286	-0.371	-0.327	0.018	-0.207
3	-0.157	0.186	-0.383	-0.338	0.011	-0.161
4	0.829	0.171	2.000	Inf	0.276	Inf
5	-0.186	0.243	-0.469	-0.418	0.024	-0.237
6	0.000	1.000	NaN	NaN	NaN	NaN
7	-0.214	0.643	-0.788	-0.742	0.372	-0.996

NRMISS = 1

Comments

1. Workspace may be explicitly provided, if desired, by use of R2TIN/DR2TIN. The reference is:

```
CALL R2TIN (NRX, NCOL, X, LDX, INTCEP, IIND, INDIND, IWT, R,
LDR, DFE, SSE, E, OTIN, LDOTIN, NRMISS, WK)
```

The additional argument is:

WK— Work vector of length INTCEP + |IIND|.

2. Informational errors

Type Code

- | | | |
|---|---|---|
| 3 | 2 | The linear combination of the regression coefficients specified is not estimable within the preset tolerance. |
| 3 | 3 | A leverage much greater than 1.0 was computed. It is set to 1.0. |
| 3 | 4 | A deleted residual mean square much less than 0.0 was computed. It is set to 0.0. |
| 4 | 1 | A weight is negative. Weights must be nonnegative. |

Description

The multiple regression model used by routine ROTIN is

$$y = X\beta + \varepsilon$$

where y is the $n \times 1$ vector of responses, X is the $n \times p$ matrix of regressors, β is the $p \times 1$ vector of regression coefficients, and ε is the $n \times 1$ vector of errors whose elements are independently normally distributed with mean 0 and variance σ^2/w_i . The model used by ROTIN also permits linear equality restrictions on β . From a multiple regression model fit using the w_i 's as the weights, routine ROTIN computes diagnostics for outliers and influential cases. Let x_i be a column vector containing elements of the i -th row of X . Let $W = \text{diag}(w_1, w_2, \dots, w_n)$. The leverage is defined as

$$h_i = x_i^T (X^T W X)^{-1} x_i w_i$$

(In the case of linear equality restrictions on β , the leverage is defined in terms of the reduced model.) Put $D = \text{diag}(d_1, d_2, \dots, d_p)$ with $d_j = 1$ if the j -th diagonal element of R is positive and 0 otherwise. The leverage is computed as $h_i = (a^T D a) w_i$ where a is a solution to $R^T a = x_i$. The computation of the remainder of the case diagnostics follows easily from their definitions. See the chapter introduction (page 83) for definitions of the case diagnostics.

The type 3 informational errors can occur if the input variables X , R , E and SSE are not consistent with each other or if excessive rounding has occurred in their computation. The type 3 error message with error code 2 arises when X contains a row not in the space spanned by the rows of R . An examination of the model that was fitted and the X for which diagnostics are to be computed is required in order to insure that only linear combinations of the regression coefficients that can be estimated from the fitted model are specified in X . For further details, see the discussion of estimable functions given by Maindonald (1984, pages 166–168) and Searle (1971, pages 180–188).

Additional Examples

Example 2

In this example, routine RNLIN (page 304) is first invoked to fit the following nonlinear regression model discussed by Neter, Wasserman, and Kutner (1983, pages 475–478):

$$y_i = \theta_1 e^{\theta_2 x_i} + \varepsilon_i \quad i = 1, 2, \dots, 15$$

Then, ROTIN is used to compute case diagnostics. In addition, the leverage output by ROTIN is used to construct asymptotic confidence intervals on the mean of the nonlinear regression function evaluated at x_i . The asymptotic 95% confidence intervals are computed using the formula:

$$\hat{y}_i \mp t_{.975, DFE} \sqrt{s^2 h_i}$$

where h_i is the computed leverage, $t_{.975, DFE}$ is the 97.5 percentile of the t distribution with DFE degrees of freedom as computed by routine TIN (see Chapter 17, Probability Distribution Functions and Inverses), and s^2 equals SSE/DFE .

```

USE IMSL_LIBRARIES
INTEGER   LDOTIN, LDR, NOBS, NPARM, NRX
PARAMETER (NOBS=15, NPARM=2, NRX=1, LDOTIN=NRX, LDR=NPARM)
!
INTEGER   IDERIV, IDUMMY(1), IEND, IOBS, IRANK, J, NOUT, NRMIS
REAL      A, DE(NPARM, 1), DFE, E(1), FRQ, OTIN(LDOTIN,6), &
          R(LDR,NPARM), SQRT, SSE, THETA(NPARM), WT, Y, &
          YHAT
INTRINSIC SQRT
EXTERNAL EXAMPL
!
DATA THETA/60.0, -0.03/
!
CALL UMACH (2, NOUT)
!
IDERIV = 1
CALL RNLIN (EXAMPL, THETA, R=R, DFE=DFE, SSE=SSE)
!
WRITE (NOUT,*) ' Obs.  Pred.  Res.  Lev. St Res Del Res Cook '// &
              'D DFFIT Conf Interval'
DO 10 IOBS=1, NOBS
  CALL EXAMPL (NPARM, THETA, 0, IOBS, FRQ, WT, E, DE, IEND)
  CALL EXAMPL (NPARM, THETA, 1, IOBS, FRQ, WT, E, DE, IEND)
  CALL EXAMPL (NPARM, THETA, 2, IOBS, FRQ, WT, Y, DE, IEND)
  YHAT = Y + E(1)
  CALL ROTIN (DE, -NPARM, IDUMMY, R, DFE, SSE, E, OTIN, &
             NRX=NRX, LDX=1, INTCEP=0)
  A = TIN(0.975,DFE)*SQRT((SSE/DFE)*OTIN(1,2))
  WRITE (NOUT,'(F5.1,10F7.2)') Y, YHAT, (OTIN(1,J),J=1,6), &
    YHAT - A, YHAT + A
10 CONTINUE
END
!
SUBROUTINE EXAMPL (NPARM, THETA, IOPT, IOBS, FRQ, WT, E, DE, &
                  IEND)
INTEGER   NPARM, IOPT, IOBS, IEND
REAL      THETA(NPARM), FRQ, WT, E(1), DE(NPARM, 1)
!
INTEGER   NOBS
PARAMETER (NOBS=15)
!
REAL      EXP, XDATA(NOBS), YDATA(NOBS)
INTRINSIC EXP
!
DATA YDATA/54.0, 50.0, 45.0, 37.0, 35.0, 25.0, 20.0, 16.0, 18.0, &
      13.0, 8.0, 11.0, 8.0, 4.0, 6.0/
DATA XDATA/2.0, 5.0, 7.0, 10.0, 14.0, 19.0, 26.0, 31.0, 34.0, &
      38.0, 45.0, 52.0, 53.0, 60.0, 65.0/
!
IF (IOBS .LE. NOBS) THEN
  WT = 1.0E0
  FRQ = 1.0E0
  IEND = 0
  IF (IOPT .EQ. 0) THEN

```

```

      E(1) = YDATA(IOBS) - THETA(1)*EXP(THETA(2)*XDATA(IOBS))
ELSE IF (IOPT .EQ. 1) THEN
      DE(1, 1) = -EXP(THETA(2)*XDATA(IOBS))
      DE(2, 1) = -THETA(1)*XDATA(IOBS)*EXP(THETA(2)*XDATA(IOBS))
ELSE IF (IOPT .EQ. 2) THEN
      E(1) = YDATA(IOBS)
END IF
ELSE
      IEND = 1
END IF
RETURN
END

```

Output

Obs.	Pred.	Res.	Lev.	St Res	Del Res	Cook	D	DFFIT	Conf	Interval
54.0	53.86	-0.14	0.40	-0.09	-0.09	0.00	-0.07	51.19	56.53	
50.0	51.92	1.92	0.24	1.13	1.14	0.21	0.65	49.84	54.00	
45.0	45.58	0.58	0.18	0.33	0.32	0.01	0.15	43.79	47.37	
37.0	34.55	-2.45	0.13	-1.34	-1.39	0.13	-0.54	33.04	36.07	
35.0	36.33	1.33	0.11	0.72	0.71	0.03	0.24	34.96	37.70	
25.0	22.37	-2.63	0.11	-1.42	-1.49	0.12	-0.52	21.00	23.75	
20.0	19.06	-0.94	0.12	-0.51	-0.50	0.02	-0.18	17.61	20.51	
16.0	14.82	-1.18	0.12	-0.65	-0.63	0.03	-0.23	13.35	16.29	
18.0	20.74	2.74	0.12	1.50	1.58	0.15	0.58	19.29	22.20	
13.0	12.98	-0.02	0.11	-0.01	-0.01	0.00	0.00	11.56	14.40	
8.0	6.13	-1.87	0.10	-1.01	-1.01	0.06	-0.33	4.81	7.45	
11.0	14.52	3.52	0.08	1.88	2.12	0.15	0.62	13.33	15.70	
8.0	8.81	0.81	0.08	0.43	0.42	0.01	0.12	7.64	9.97	
4.0	2.55	-1.45	0.06	-0.77	-0.75	0.02	-0.19	1.53	3.57	
6.0	7.53	1.53	0.05	0.80	0.79	0.02	0.18	6.61	8.45	

GCLAS

Gets the unique values of each classification variable.

Required Arguments

X — *NROW* by *NCOL* matrix containing the data. (Input)

INDCL — Index vector of length *NCLVAR* containing the column numbers of *X* that are the classification variables. (Input)

MAXCL — An upper bound on the sum of the number of distinct values taken on by each classification variable. (Input)

NCLVAL — Vector of length *NCLVAR* containing the number of values taken on by each classification variable. (Output, if *IDO* = 0 or *IDO* = 1; input/output, if *IDO* = 2 or *IDO* = 3)

NCLVAL(I) is the number of distinct values for the *I*-th classification variable.

CLVAL — Vector of length $NCLVAL(1) + NCLVAL(2) + \dots + NCLVAL(NCLVAR)$ containing the values of the classification variables. (Output, if $IDO = 0$ or $IDO = 1$; input/output, if $IDO = 2$ or $IDO = 3$)

Since in general the length of **CLVAL** will not be known in advance, **MAXCL** (an upper bound for this length) should be used for purposes of dimensioning **CLVAL**. The first $NCLVAL(1)$ variables contain the values of the first classification variable. The next $NCLVAL(2)$ variables contain the values of the second classification variable. ... The last $NCLVAL(NCLVAR)$ variables contain the values of the last classification variable. After invocation of **GCLAS** with $IDO = 3$, **CLVAL** contains the values sorted in ascending order by the classification variable.

Optional Arguments

IDO — Processing option. (Input)
Default: $IDO = 0$.

IDO Action

- | | |
|---|--|
| 0 | This is the only invocation of GCLAS for this data set, and all the data are input at once. |
| 1 | This is the first invocation, and additional calls to GCLAS will be made. Unique values for the classification variables are retrieved from X . |
| 2 | This is an intermediate invocation of GCLAS . Unique values for the classification variables are retrieved from X . |
| 3 | This is the final invocation of GCLAS . Unique values for the classification variables are retrieved from X , and the values in CLVAL are sorted in ascending order for each classification variable. |

NROW — Number of rows of data in **X**. (Input)
Default: $NROW = \text{size}(X,1)$.

NCOL — Number of columns in **X**. (Input)
Default: $NCOL = \text{size}(X,2)$.

LDX — Leading dimension of **X** exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDX = \text{size}(X,1)$.

NCLVAR — Number of classification variables. (Input)
Default: $NCLVAR = \text{size}(INDCL,1)$.

NMISS — Vector of length $NCLVAR$ containing the number of elements of the data containing NaN for any classification variable. (Output, if $IDO = 0$ or $IDO = 1$; input/output if $IDO = 2$ or $IDO = 3$)

FORTRAN 90 Interface

Generic: CALL GCLAS (X, INDCL, MAXCL, NCLVAL, CLVAL [,...])

Specific: The specific interface names are S_GCLAS and D_GCLAS.

FORTRAN 77 Interface

Single: CALL GCLAS (IDO, NROW, NCOL, X, LDX, NCLVAR, INDCL, MAXCL,
 NCLVAL, CLVAL, NMISS)

Double: The double precision name is DGCLAS.

Example

In the following example, the unique values of two classification variables are obtained from a data set XX with six rows. Here, routine GCLAS is invoked repeatedly with one row of the data set input into X at a time. Initially, GCLAS is invoked with IDO = 1, then with IDO = 2 for each of the six rows of data, and finally with IDO = 3.

```
USE GCLAS_INT
USE SCOPY_INT
USE WRRRL_INT
USE WRIRL_INT

INTEGER      LDX, LDXX, MAXCL, NCLVAR, NCOL, NOBS
PARAMETER   (LDX=1, MAXCL=5, NCLVAR=2, NCOL=2, NOBS=6, LDXX=NOBS)
!
INTEGER      I, IDO, INDCL(NCLVAR), NCLVAL(NCLVAR), NMISS(NCLVAR), &
            NROW
REAL         CLVAL(MAXCL), X(LDX,NCOL), XX(LDXX,NCOL)
CHARACTER    CLABEL(2)*8, RLABEL(1)*17
!
DATA INDCL/1, 2/, NCLVAL/2, 3/
DATA (XX(1,J),J=1,NCOL)/10.0, 5.0/
DATA (XX(2,J),J=1,NCOL)/20.0, 15.0/
DATA (XX(3,J),J=1,NCOL)/20.0, 10.0/
DATA (XX(4,J),J=1,NCOL)/10.0, 10.0/
DATA (XX(5,J),J=1,NCOL)/10.0, 15.0/
DATA (XX(6,J),J=1,NCOL)/20.0, 5.0/
!
IDO = 1
NROW = 0
CALL GCLAS (X, INDCL, MAXCL, NCLVAL, CLVAL, IDO=IDO, NROW=NROW, &
           NMISS=NMISS)
IDO = 2
NROW = 1
DO 10 I=1, NOBS
    CALL SCOPY (NCOL, XX(I:,1), LDXX, X(1:,1), LDX)
    CALL GCLAS (X, INDCL, MAXCL, NCLVAL, CLVAL, IDO=IDO, NROW=NROW, &
               NMISS=NMISS)
10 CONTINUE
IDO = 3
NROW = 0
```

```

CALL GCLAS (X, INDCL, MAXCL, NCLVAL, CLVAL, IDO=IDO, &
           NROW=NROW, NMISS=NMISS)
I          = 1
RLABEL(1) = 'Variable  CLVAL:'
CLABEL(1) = 'None'
DO 20 J=1, NCLVAR
  WRITE (RLABEL(1)(9:10),'(I2)') J
  CALL WRRRL (' ', CLVAL(I:), RLABEL, CLABEL, 1, NCLVAL(J), 1)
  I = I + NCLVAL(J)
20 CONTINUE
RLABEL(1) = 'NUMBER'
CLABEL(1) = 'Variable'
CLABEL(2) = 'NMISS'
CALL WRIRL ('%/', NMISS, RLABEL, CLABEL)
END

```

Output

```

Variable 1 CLVAL:  10.00  20.00
Variable 2 CLVAL:   5.00  10.00  15.00

```

```

Variable  NMISS
      1      0
      2      0

```

Comments

Informational error
Type Code

4 1 MAXCL is too small. Increase MAXCL and the dimension of CLVAL.

Description

Routine GCLAS gets the unique values of m (Input in NCLVAR) classification variables. The routine can be used in conjunction with routine GRGLM ([page 225](#)). Routine GRGLM requires the values of the classification variables output by GCLAS in order to generate dummy variables for the general linear model.

In the input array X , missing values for a classification variable can be indicated by NaN (not a number). NAN is represented by AMACH(6). (See the section “Machine-Dependent Constants” found under Reference Material for a further discussion of AMACH, and missing values.) The nonmissing values of the classifications variables are output in CLVAL. If for a particular row of X a value of a classification variable is missing, nonmissing values of the other classification variables are still used. The number of elements equal to NaN for each classification variable is output in NMISS.

GRGLM

Generates regressors for a general linear model.

Required Arguments

X — $NROW$ by $NCOL$ matrix containing the data. (Input)

INDCL — Index vector of length $NCLVAR$ containing the column numbers of X that are the classification variables. (Input)

NCLVAL — Vector of length $NCLVAR$ containing the number of values taken on by each classification variable. (Input)
 $NCLVAL(I)$ is the number of distinct values for the I -th classification variable.

CLVAL — Vector of length $NCLVAL(1) + NCLVAL(2) + \dots + NCLVAL(NCLVAR)$ containing the values of the classification variables. (Input)
The first $NCLVAL(1)$ elements contain the values of the first classification variable. The next $NCLVAL(2)$ elements contain the values of the second classification variable. ...
The last $NCLVAL(NCLVAR)$ elements contain the values of the last classification variable.

NVEF — Vector of length NEF containing the number of variables associated with each effect in the model. (Input)

INDEF — Index vector of length $NVEF(1) + NVEF(2) + \dots + NVEF(NEF)$. (Input)
The first $NVEF(1)$ elements give the column numbers of X for each variable in the first effect. The next $NVEF(2)$ elements give the column numbers for each variable in the second effect. ... The last $NVEF(NEF)$ elements give the column numbers for each variable in the last effect.

NREG — Number of columns in *REG*. (Output)

REG — $NROW$ by $NREG$ matrix containing the regressor variables generated from the matrix X . (Output, if $IDUMMY > 0$)
Since, in general, $NREG$ will not be known in advance, the user may need to invoke *GRGLM* first with $IDUMMY < 0$, dimension REG , and then invoke *GRGLM* with $IDUMMY > 0$.

Optional Arguments

NROW — Number of rows of data in X . (Input)
Default: $NROW = \text{size}(X,1)$.

NCOL — Number of columns in X . (Input)
Default: $NCOL = \text{size}(X,2)$.

LDX — Leading dimension of X exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDX = \text{size}(X,1)$.

NCLVAR — Number of classification variables. (Input)

Default: $NCLVAR = \text{size}(\text{INDCL}, 1)$.

NEF — Number of effects (sources of variation) in the model. (Input)

Default: $NEF = \text{size}(\text{NVEF}, 1)$.

IDUMMY — Dummy variable option. (Input)

Default: $IDUMMY = 1$.

Some indicator variables are defined for the I -th class variable as follows: Let $J = NCLVAL(1) + NCLVAL(2) + \dots + NCLVAL(I - 1)$. $NCLVAL(I)$ indicator variables are defined such that for $K = 1, 2, \dots, NCLVAL(I)$ the K -th indicator variable for observation number $IOBS$ takes the value 1.0 if $X(IOBS, \text{INDCL}(I)) = \text{CLVAL}(J + K)$ and equals 0.0 otherwise. Dummy variables are generated from these indicator variables in one of the three following ways:

IDUMMY Method

-1, 1 The $NCLVAL(I)$ indicator variables are the dummy variables.

-2, 2 The first $NCLVAL(I) - 1$ indicator variables are the dummy variables. The last indicator variable is omitted.

-3, 3 The K -th indicator variable minus the $NCLVAL(I)$ -th indicator variable is the K -th dummy variable ($K = 1, 2, \dots, NCLVAL(I) - 1$).

If $IDUMMY < 0$, only $NREG$ is computed; and X , $CLVAL$, and REG are not referenced.

LDREG — Leading dimension of REG exactly as specified in the dimension statement in the calling program. (Input)

Default: $LDREG = \text{size}(\text{REG}, 1)$.

NRMIS — Number of rows of REG containing NaN (not a number). (Output)

A row of REG contains NaN for a regressor when any of the variables involved in generation of the regressor equals NaN or if a value of one of the classification variables in the model is not given by $CLVAL$.

FORTRAN 90 Interface

Generic: `CALL GRGLM (X, INDCL, NCLVAL, CLVAL, NVEF, INDEF, NREG, REG [, ...])`

Specific: The specific interface names are `S_GRGLM` and `D_GRGLM`.

FORTRAN 77 Interface

Single: CALL GRGLM (NROW, NCOL, X, LDX, NCLVAR, INDCL, NCLVAL,
 CLVAL, NEF, NVEF, INDEF, IDUMMY, NREG, REG, LDREG, NRMISS)

Double: The double precision name is DGRGLM.

Example

In this example, regressors are generated for a two-way analysis-of-covariance model containing all the interaction terms. The model could be fitted by a subsequent invocation of routine RGIVN (see page 117) with INTCEP = 1. The regressors generated with the option IDUMMY = 2 are for the model whose mean function is

$$\mu + \alpha_i + \beta_j + \gamma_{ij} + \delta x_{ij} + \zeta_i x_{ij} + \eta_j x_{ij} + \theta_{ij} x_{ij} \quad i = 1, 2; j = 1, 2, 3$$

where $\alpha_2 = \beta_3 = \gamma_{13} = \gamma_{21} = \gamma_{22} = \gamma_{23} = \zeta_2 = \eta_3 = \theta_{13} = \theta_{21} = \theta_{22} = \theta_{23} = 0$.

```
USE GRGLM_INT
USE UMACH_INT
USE WRRRL_INT

INTEGER      LDREG, LDX, LINDEF, MAXCL, NCLVAR, NCOL, NDREG, NEF, &
             NROW
PARAMETER    (LINDEF=12, MAXCL=5, NCLVAR=2, NCOL=3, NDREG=20, &
             NEF=7, NROW=6, LDREG=NROW, LDX=NROW)
!
INTEGER      IDUMMY, INDCL(NCLVAR), INDEF(LINDEF), J, &
             NCLVAL(NCLVAR), NOUT, NREG, NRMISS, NVEF(NEF)
REAL         CLVAL(MAXCL), REG(LDREG,NDREG), X(LDX,NCOL)
CHARACTER    CLABEL(12)*7, RLABEL(1)*7
!
DATA INDCL/1, 2/, NCLVAL/2, 3/, CLVAL/1.0, 2.0, 1.0, 2.0, 3.0/
DATA NVEF/1, 1, 2, 1, 2, 2, 3/, INDEF/1, 2, 1, 2, 3, 1, 3, 2, 3, &
      1, 2, 3/
DATA (X(1,J),J=1,NCOL)/1.0, 1.0, 1.11/
DATA (X(2,J),J=1,NCOL)/1.0, 2.0, 2.22/
DATA (X(3,J),J=1,NCOL)/1.0, 3.0, 3.33/
DATA (X(4,J),J=1,NCOL)/2.0, 1.0, 4.44/
DATA (X(5,J),J=1,NCOL)/2.0, 2.0, 5.55/
DATA (X(6,J),J=1,NCOL)/2.0, 3.0, 6.66/
DATA RLABEL/'NUMBER'/, CLABEL/' ', 'ALPHA1', 'BETA1', &
      'BETA2', 'GAMMA11', 'GAMMA12', 'DELTA', 'ZETA1', &
      'ETA1', 'ETA2', 'THETA11', 'THETA12'/
!
IDUMMY = 2
CALL GRGLM (X, INDCL, NCLVAL, CLVAL, NVEF, INDEF, NREG, REG, &
           IDUMMY=IDUMMY, NRMISS=NRMISS)
CALL UMACH (2, NOUT)
WRITE (NOUT,*) 'NREG = ', NREG, ' NRMISS = ', NRMISS
CALL WRRRL ('%/REG', REG, RLABEL, CLABEL, NROW, NREG, FMT='(F7.2)')
END
```

Output

NREG = 11 NRMIS = 0

	REG								
	ALPHA1	BETA1	BETA2	GAMMA11	GAMMA12	DELTA	ZETA1	ETA1	ETA2
1	1.00	1.00	0.00	1.00	0.00	1.11	1.11	1.11	0.00
2	1.00	0.00	1.00	0.00	1.00	2.22	2.22	0.00	2.22
3	1.00	0.00	0.00	0.00	0.00	3.33	3.33	0.00	0.00
4	0.00	1.00	0.00	0.00	0.00	4.44	0.00	4.44	0.00
5	0.00	0.00	1.00	0.00	0.00	5.55	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	6.66	0.00	0.00	0.00

	ETA2	THETA11	THETA12
1	0.00	1.11	0.00
2	2.22	0.00	2.22
3	0.00	0.00	0.00
4	0.00	0.00	0.00
5	5.55	0.00	0.00
6	0.00	0.00	0.00

Comments

Let the data matrix $X = (A, B, X_1)$ where A and B are classification variables, and X_1 is a continuous variable. The model containing the effects $A, B, AB, X_1, AX_1, BX_1$ and ABX_1 is specified as follows:

NCLVAR = 2, INDCL = (1, 2), NEF = 7, NVEF = (1, 1, 2, 1, 2, 2, 3), and INDEF = (1, 2, 1, 2, 3, 1, 3, 2, 3, 1, 2, 3).

For this model, suppose NCLVAL(1) = 2, NCLVAL(2) = 3, and CLVAL = (1.0, 2.0, 1.0, 2.0, 3.0). Let $A_1, B_1, B_2,$ and B_3 be the associated indicator variables. Given below, for each IDUMMY option, are the regressors in their order of appearance in REG.

IDUMMY	REG
1	$A_1, A_2, B_1, B_2, B_3, A_1B_1, A_1B_2, A_1B_3, A_2B_1, A_2B_2, A_2B_3, X_1, A_1X_1, A_2X_1, B_1X_1, B_2X_1, B_3X_1, A_1B_1X_1, A_1B_2X_1, A_1B_3X_1, A_2B_1X_1, A_2B_2X_1, A_2B_3X_1$
2	$A_1, B_1, B_2, A_1B_1, A_1B_2, X_1, A_1X_1, B_1X_1, B_2X_1, A_1B_1X_1, A_1B_2X_1$
3	$A_1 - A_2, B_1 - B_3, B_2 - B_3, (A_1 - A_2)(B_1 - B_2), (A_1 - A_2)(B_2 - B_3), X_1, (A_1 - A_2)X_1, (B_1 - B_3)X_1, (B_2 - B_3)X_1, (A_1 - A_2)(B_1 - B_2)X_1, (A_1 - A_2)(B_2 - B_3)X_1$

Within a group of regressors corresponding to an interaction effect, the indicator variables composing the regressors vary most rapidly for the last classification variable, vary next most rapidly for the next to last classification variable, etc.

Description

Routine GRGLM generates regressors for a general linear model from a data matrix. The data matrix can contain classification variables as well as continuous variables.

Regressors for effects composed solely of continuous variables are generated as powers and crossproducts. Consider a data matrix containing continuous variables as columns 3 and 4. The effect indices (3,3) (stored in INDEF) generates a regressor whose i -th value is the square of the i -th value in column 3. The effect indices (3,4) generates a regressor whose i -th value is the product of the i -th value in column 3 with the i -th value in column 4.

Regressors for an effect (source of variation) composed of a single classification variable are generated using indicator variables. Let the classification variable A take on values a_1, a_2, \dots, a_n (stored in CLVAL). From this classification variable, GRGLM creates n indicator variables. For $k = 1, 2, \dots, n$ we have

$$I_k = \begin{cases} 1 & \text{if } A = a_k \\ 0 & \text{otherwise} \end{cases}$$

For each classification variable, another set of variables is created from the indicator variables. We call these new variables *dummy variables*. Dummy variables are generated from the indicator variables in one of three manners:

1. the dummies are the n indicator variables ,
2. the dummies are the first $n - 1$ indicator variables,
- 3 the $n - 1$ dummies are defined in terms of the indicator variables so that for balanced data, the usual summation restrictions are imposed on the regression coefficients.

In particular, for IDUMMY = 1, the dummy variables are $A_k = I_k$ ($k = 1, 2, \dots, n$). For IDUMMY = 2, the dummy variables are $A_k = I_k$ ($k = 1, 2, \dots, n - 1$). For IDUMMY = 3, the dummy variables are $A_k = I_k - I_n$ ($k = 1, 2, \dots, n - 1$). The regressors generated for an effect composed of a single classification variable are the associated dummy variables.

Let m_j be the number of dummies generated for the j -th classification variable. Suppose there are two classification variables A and B with dummies

$$A_1, A_2, \dots, A_{m_1} \text{ and } B_1, B_2, \dots, B_{m_2}$$

respectively. The regressors generated for an effect composed of two classification variables A and B are

$$\begin{aligned} & A \otimes B \\ &= (A_1, A_2, \dots, A_{m_1}) (B_1, B_2, \dots, B_{m_2}) \\ &= (A_1 B_1, A_1 B_2, \dots, A_1 B_{m_2}, A_2 B_1, A_2 B_2, \dots, A_2 B_{m_2}, A_{m_1} B_1, A_{m_1} B_2, \dots, A_{m_1} B_{m_2}) \end{aligned}$$

More generally, the regressors generated for an effect composed of several classification variables and several continuous variables are given by the Kronecker products of variables, where the order of the variables is specified in INDEF. Consider a data matrix containing classification variables in columns 1 and 2 and continuous variables in columns 3 and 4. Label these four columns A, B, X_1 , and X_2 . The regressors generated by the effect indices (1, 2, 3, 3, 4) is $A \otimes B \otimes X_1 X_1 X_2$.

RBEST

Selects the best multiple linear regression models.

Required Arguments

COV — $NVAR$ by $NVAR$ matrix containing the variance-covariance matrix or sum of squares and crossproducts matrix. (Input)
Only the upper triangle of **COV** is referenced. The last column of **COV** must correspond to the dependent variable.

NOBS — Number of observations. (Input)

ICOEFX — Index vector of length $NTBEST + 1$ containing the locations in **COEF** of the first row for each of the best regressions. (Output)
Here, $NTBEST$ is the total number of best regressions found and is given as follows:

ICRIT	NTBEST
< 0	$-NBEST * ICRIT$
1	$NBEST * (NVAR - 1)$
2	$NBEST$
3	$NBEST$

For $I = 1, 2, \dots, NTBEST$, rows $ICOEFX(I)$, $ICOEFX(I) + 1, \dots, ICOEFX(I + 1) - 1$ of **COEF** correspond to the I -th regression.

COEF — $ICOEFX(NTBEST + 1) - 1$ by 5 matrix containing statistics relating to the regression coefficients of the best models. (Output)
An upper bound on the number of rows in **COEF** is given as follows:

ICRIT	Upper Bound on the Number of Rows in COEF
< 0	$-NBEST * ICRIT * (1 - ICRIT)/2$
1	$NBEST * (NVAR - 1) * NVAR/2$
2	$NBEST * (NVAR - 1)$
3	$NBEST * (NVAR - 1)$

Each row corresponds to a coefficient for a particular regression. The regressions are in order of increasing subset size. Within each subset size, the regressions are ordered so that the better regressions appear first. The statistics in the columns are as follows:

Col. Description

- 1 Variable number
- 2 Coefficient estimate
- 3 Estimated standard error of the estimate
- 4 t -statistic for the test that the coefficient is zero
- 5 p -value for the two-sided t test

(Inferences are conditional on the selected models.)

Optional Arguments

NVAR — Number of variables. (Input)
Default: `NVAR = size (COV,2)`.

LDCOV — Leading dimension of `COV` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDCOV = size (COV,1)`.

ICRIT — Criterion option. (Input)
Default: `ICRIT = 1`.

ICRIT	Criterion	NSIZE
< 0	R^2	<code>-ICRIT</code>
1	R^2	<code>NVAR - 1</code>
2	Adjusted R^2	<code>NVAR - 1</code>
3	Mallows C_p	<code>NVAR - 1</code>

Subset sizes 1, 2, ..., `NSIZE` are examined.

NBEST — Number of best regressions to be found. (Input)
If the R^2 criterion is selected, the `NBEST` best regressions for each subset size examined are found. If the adjusted R^2 or Mallows C_p criterion is selected, the `NBEST` best overall regressions are found.
Default: `NBEST = 1`.

NGOOD — Maximum number of good regressions of each subset size to be saved in finding the best regressions. (Input)
`NGOOD` must be greater than or equal to `NBEST`. Normally, `NGOOD` should be less than or equal to 10. It need not ever be larger than the maximum number of subsets for any subset size. Computing time required is inversely related to `NGOOD`.
Default: `NGOOD = 10`.

IPRINT — Printing option. (Input)
Default: `IPRINT = 0`.

IPRINT Action

0 No printing is performed.

1 Printing is performed.

ICRITX— Index vector of length $NSIZE + 1$ containing the locations in **CRIT** of the first element for each subset size. (Output)
(See argument **ICRIT** for a definition of **NSIZE**.) For $I = 1, 2, \dots, NSIZE$, element numbers $ICRITX(I), ICRITX(I) + 1, \dots, ICRITX(I + 1) - 1$ of **CRIT** correspond to the I -th subset size.

CRIT— Vector of length $\max(ICRITX(NSIZE + 1) - 1, NVAR - 1)$ containing in its first $ICRITX(NSIZE + 1) - 1$ elements the criterion values for each subset considered, in increasing subset size order. (Output)
An upper bound on the length of **CRIT** is $\max(NGOOD * NSIZE, NVAR - 1)$. Within each subset size, results are returned in monotone order according to the criterion value with the results for the better regressions given first.

IVARX— Index vector of length $NSIZE + 1$ containing the locations in **INDVAR** of the first element for each subset size. (Output)
For $I = 1, 2, \dots, NSIZE$, element numbers $IVARX(I), IVARX(I) + 1, \dots, IVARX(I + 1) - 1$ of **INDVAR** correspond to the I -th subset size.

INDVAR— Index vector of length $IVARX(NSIZE + 1) - 1$ containing the variable numbers for each subset considered and in the same order as in **CRIT**. (Output)
An upper bound on the length of **INDVAR** is $NGOOD * NSIZE * (NSIZE + 1)/2$.

LDCOEF— Leading dimension of **COEF** exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDCOEF = \text{size}(\text{COEF}, 1)$.

FORTRAN 90 Interface

Generic: `CALL RBEST (COV, NOBS, ICOEFX, COEF [,...])`

Specific: The specific interface names are `S_RBEST` and `D_RBEST`.

FORTRAN 77 Interface

Single: `CALL RBEST (NVAR, COV, LDCOV, NOBS, ICRIT, NBEST, NGOOD, IPRINT, ICRITX, CRIT, IVARX, INDVAR, ICOEFX, COEF, LDCOEF)`

Double: The double precision name is `DRBEST`.

Example

This example uses a data set from Draper and Smith (1981, pages 629–630). This data set is input to the matrix X by routine `GDATA` (see Chapter 19, Utilities). The first four columns

contain the independent variables, and the last column contains the dependent variable. Routine CORVC in Chapter 3, "Correlation," is invoked to compute the corrected sum of squares and crossproducts matrix. Routine RBEST is then invoked to find the best regression for each of the four subset sizes using the R^2 criterion.

```

USE RBEST_INT
USE GDATA_INT
USE CORVC_INT
INTEGER   LDcoef, LDcov, LDX, NBEST, NGOOD, NSIZE, NTBEST, NVAR
PARAMETER (LDX=13, NBEST=1, NGOOD=10, NVAR=5, &
           LDcoef=NBEST*(NVAR-1)*NVAR/2, LDcov=NVAR, &
           NSIZE=NVAR-1, NTBEST=NBEST*(NVAR-1))
!
INTEGER   ICOEFX(NTBEST+1), ICOPT, ICRT, ICRTX(NSIZE+1), &
          INCD(1,1), INDVAR(NGOOD*NSIZE*(NSIZE+1)/2), &
          IPRINT, IVARX(NSIZE+1), NMISS, NOBS, NROW, NVAR1
REAL      COEF(LDcoef,5), COV(LDcov,NVAR), CRIT(NGOOD*NSIZE), &
          SUMWT, X(LDX,NVAR), XMEAN(NVAR)
!
CALL GDATA (5, X, NROW, NVAR1)
!
ICOPT = 1
CALL CORVC (NVAR, X, COV, ICOPT=ICOPT, NOBS=NOBS)
!
IPRINT = 1
CALL RBEST (COV, NOBS, ICOEFX, COEF, IPRINT=IPRINT)
!
END

```

Output

Regressions with 1 variable(s) (R-squared)	
Criterion	Variables
67.5	4
66.6	2
53.4	1
28.6	3

Regressions with 2 variable(s) (R-squared)	
Criterion	Variables
97.9	1 2
97.2	1 4
93.5	3 4
68.0	2 4
54.8	1 3

Regressions with 3 variable(s) (R-squared)	
Criterion	Variables
98.2	1 2 4
98.2	1 2 3
98.1	1 3 4
97.3	2 3 4

Regressions with 4 variable(s) (R-squared)	
Criterion	Variables
98.2	1 2 3 4

Best Regression with 1 variable(s) (R-squared)				
Variable	Coefficient	Standard Error	t-statistic	p-value
4	-0.7382	0.1546	-4.775	0.0006
Best Regression with 2 variable(s) (R-squared)				
Variable	Coefficient	Standard Error	t-statistic	p-value
1	1.468	0.1213	12.10	0.0000
2	0.662	0.0459	14.44	0.0000
Best Regression with 3 variable(s) (R-squared)				
Variable	Coefficient	Standard Error	t-statistic	p-value
1	1.452	0.1170	12.41	0.0000
2	0.416	0.1856	2.24	0.0517
4	-0.237	0.1733	-1.36	0.2054
Best Regression with 4 variable(s) (R-squared)				
Variable	Coefficient	Standard Error	t-statistic	p-value
1	1.551	0.7448	2.083	0.0708
2	0.510	0.7238	0.705	0.5009
3	0.102	0.7547	0.135	0.8959
4	-0.144	0.7091	-0.203	0.8441

Comments

1. Workspace may be explicitly provided, if desired, by use of `R2EST/DR2EST`. The reference is:

```
CALL R2EST (NVAR, COV, LDICOV, NOBS, ICRT, NBEST, NGOOD, IPRINT,
ICRTX, CRIT, IVARX, INDVAR, ICOEFX, COEF, LDICOF, WK, IWK)
```

The additional arguments are as follows:

WK — Work vector of length $NVAR * (2 * NGOOD + 6) + (2 * NVAR^3 + 4 * NVAR)/3$. The first $NVAR - 1$ locations indicate which variables are in the full model. If $IWK(I) = 0$, then variable I is in the full model, otherwise, the variable has been dropped.

IWK — Integer work vector of length $3 * NVAR^2 + 6 * NVAR$.

2. Informational errors

Type Code

- | | | |
|---|---|---|
| 3 | 1 | At least one variable is deleted is from the full model because <code>COV</code> is singular. |
| 4 | 3 | No variables can enter any model. |

Description

Routine `RBEST` finds the best subset regressions for a regression problem with $NVAR - 1$ candidate independent variables. Typically, the intercept is forced into all models and is not a candidate variable. In this case, a sum of squares and crossproducts matrix for the independent and dependent variables corrected for the mean is input for `COV`. Routine `CORVC` in Chapter 3,

“Correlation,” can be used to compute the corrected sum of squares and crossproducts. IMSL routine `RORDM` in Chapter 19, “Utilities,” can be used to reorder this matrix, if required. Other possibilities are

1. The intercept is not in the model. A raw (uncorrected) sum of squares and crossproducts matrix for the independent and dependent variables.

is required for `COV`. `NOBS` must be set to one greater than the number of observations. Routine `MXTXF` (IMSL MATH/LIBRARY) can be used to compute the raw sum of squares and crossproducts matrix.
2. An intercept is to be a candidate variable. A raw (uncorrected) sum of squares and crossproducts matrix for the constant regressor ($= 1$), independent, and dependent variables is required for `COV`. In this case, `COV` contains one additional row and column corresponding to the constant regressor. This row/column contains the sum of squares and crossproducts of the constant regressor with the independent and dependent variables. The remaining elements in `COV` are the same as in the previous case. `NOBS` must be set to one greater than the number of observations.
3. There are m variables to be forced into the models. A sum of squares and crossproducts matrix adjusted for the m variables is required. `NOBS` must be set to m less than the number of observations. Routine `RCOV` (page 113) can be used to compute the adjusted sum of squares and crossproducts matrix. This is accomplished by a regression of the candidate variables on the variables to be forced into the models. The error sum of squares and crossproducts matrix, `SCPE` from `RCOV`, is the input to `COV` in routine `REEST`.

“Best” is defined, on option, by one of three criteria:

1. R^2 (in percent)

$$R^2 = 100 \left(1 - \frac{SSE_p}{SST} \right)$$

2. R_a^2
(adjusted R^2 in percent)

$$R_a^2 = 100 \left[1 - \left(\frac{n-1}{n-p} \right) \frac{SSE_p}{SST} \right]$$

Note that maximizing this criterion is equivalent to minimizing the residual mean square, $SSE_p/(n-p)$.

3. Mallows' C_p statistic

$$C_p = \frac{SSE_p}{s_{\text{NVAR}-1}^2} + 2p - n$$

Here, n is NOBS, and SST is the total sum of squares. SSE_p is the error sum of squares in a model containing p regression parameters including β_0 (or $p - 1$ of the NVAR - 1 candidate variables).

$$s_{NVAR-1}^2$$

is the error mean square from the model with all NVAR - 1 candidate variables in the model. Hocking (1972) and Draper and Smith (1981, pages 296–302) discuss these criteria.

Routine RBEST is based on the algorithm of Furnival and Wilson (1974), this algorithm finds NGOOD candidate regressions for each possible subset size. These regressions are used to identify a set of best regressions. In large problems, many regressions are not computed. They may be rejected without computation based on results for other subsets, this yields an efficient technique for considering all possible regressions.

Programming Notes

Routine RBEST can save considerable CPU time over explicitly computing all possible regressions. However, the routine has some limitations that can cause unexpected results for users that are unaware of the limitations of the software.

1. For $NVAR > -\log_2(\epsilon)$ where ϵ is AMACH(4), (See the section “Machine-Dependent Constants” in Reference Material), some results can be incorrect. This limitation arises because the possible models indicated by the model numbers 1, 2, ..., 2^{NVAR-1} , are stored as floating-point values, for sufficiently large NVAR, the model numbers cannot be stored exactly. On many computers, this means S_RBEST (for $NVAR > 25$) and D_RBEST (for $NVAR > 50$) can produce incorrect results.
2. Routine RBEST eliminates some subsets of candidate variables by obtaining lower bounds on the error sum of squares from fitting larger models. First, the full model containing all NVAR - 1 is fit sequentially using a forward stepwise procedure in which one variable enters the model at a time, and criterion values and model numbers for all the candidate variables that can enter at each step are stored. If linearly dependent variables are removed from the full model, error code 1 is issued. If this error is issued, some submodels that contain variables removed from the full model because of linear dependency can be overlooked, if they have not already been identified during the initial forward stepwise procedure. If error code 1 is issued and you want the variables that were removed from the full model to be considered in smaller models, you may want to rerun the program with a set of linearly independent variables.

RSTEP

Builds multiple linear regression models using forward selection, backward selection, or stepwise selection.

Required Arguments

COV — $NVAR$ by $NVAR$ matrix containing the variance-covariance matrix or sum of squares and crossproducts matrix. (Input)
Only the upper triangle of **COV** is referenced.

NOBS — Number of observations. (Input)

AOV — Vector of length 13 containing statistics relating to the analysis of variance for the final model in this invocation. (Output)

I	AOV(I)
1	Degrees of freedom for regression
2	Degrees of freedom for error
3	Total degrees of freedom
4	Sum of squares for regression
5	Sum of squares for error
6	Total sum of squares
7	Regression mean square
8	Error mean square
9	F -statistic
10	p -value
11	R^2 (in percent)
12	Adjusted R^2 (in percent)
13	Estimated standard deviation of the model error

COEF — $NVAR - 1$ by 5 matrix containing statistics relating to the regression coefficients for the final model in this invocation. (Output)

The rows correspond to the $NVAR - 1$ variables with **LEVEL(I)** nonnegative, i.e., all variables but the dependent variable. The rows are in the same order as the variables in **COV** except that the dependent variable is excluded. Each row corresponding to a variable not in the model is for the model supposing the additional variable was in the model.

Col. Description

- 1 Coefficient estimate
- 2 Estimated standard error of the coefficient estimate
- 3 *t*-statistic for the test that the coefficient is zero
- 4 *p*-value for the two-sided *t* test
- 5 Variance inflation factor. The square of the multiple correlation coefficient for the *I*-th regressor after all others can be obtained from `COEF(I, 5)` by the formula $1.0 - 1.0/\text{COEF}(I, 5)$.

COVS — *NVAR* by *NVAR* matrix that results after `COV` has been swept on the columns corresponding to the variables in the model. (Output, if `INVOKE = 0` or `1`; input/output, if `INVOKE = 2` or `3`)
The estimated variance-covariance matrix of the estimated regression coefficients in the final model can be obtained by extracting the rows and columns of `COVS` corresponding to the independent variables in the final model and multiplying the elements of this matrix by `AOV(8)`. If `COV` is not needed, `COV` and `COVS` can occupy the same storage locations.

Optional Arguments

INVOKE — Invocation option. (Input)
Default: `INVOKE = 0`.

INVOKE Action

- 0 This is the only invocation of `RSTEP` for this variance-covariance matrix. Initialization, stepping, and wrap-up computations are performed.
- 1 This is the first invocation of `RSTEP`, and additional calls to `RSTEP` will be made. Initialization and stepping is performed.
- 2 This is an intermediate invocation of `RSTEP` and stepping is performed.
- 3 This is the final invocation of `RSTEP` and stepping is performed.

NVAR — Number of variables. (Input)
Default: `NVAR = size(COV,2)`.

LDCOV — Leading dimension of `COV` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDCOV = size(COV,1)`.

LEVEL — Vector of length `NVAR` containing levels of priority for variables entering and leaving the regression. (Input)
`LEVEL(I) = -1` means the `I`-th variable is the dependent variable. `LEVEL(I) = 0` means the `I`-th variable is never to enter into the model. Other variables must be assigned a positive value to indicate their level of entry into the model. A variable can enter the model only after all variables with smaller nonzero levels of entry have entered. Similarly, a variable can only leave the model after all variables with higher levels of entry have left. Variables with the same level of entry compete for entry (deletion) at each step.

NFORCE — Variables with levels 1, 2, ..., `NFORCE` are forced into the model as the independent variables. (Input)
Default: `NFORCE = 0`.

NSTEP — Step length option. (Input)
For nonnegative `NSTEP`, `NSTEP` steps are taken. `NSTEP = -1` means stepping continues until completion.
Default: `NSTEP = -1`.

ISTEP — Stepping option. (Input)
Default: `ISTEP = -1`.

ISTEP Action

- 1 An attempt is made to remove a variable from the model (backward step). A variable is removed if its p -value exceeds `POUT`. During initialization, all candidate independent variables enter the model.
- 1 An attempt is made to add a variable to the model (forward step). A variable is added if its p -value is less than `PIN`. During initialization, only the forced variables enter the model.
- 0 A backward step is attempted. If a variable is not removed, a forward step is attempted. This is a stepwise step. Only the forced variables enter the model during initialization.

PIN — Largest p -value for entering variables. (Input)
Variables with p -values less than `PIN` may enter the model. A common choice is `PIN = 0.05`.
Default: `PIN = .05`.

POUT — Smallest p -value for removing variables. (Input)
Variables with p -values greater than `POUT` may leave the model. `POUT` must be greater or equal to `PIN`. A common choice is `POUT = 0.10` (or `2 * PIN`).
Default: `POUT = .10`.

TOL — Tolerance used in determining linear dependence. (Input)
`TOL = 100 * AMACH (4)` is a common choice. See documentation for `AMACH` in the

Reference Material.

Default: `TOL = 1.e-5` for single precision and `2.d - 14` for double precision.

IPRINT — Printing option. (Input)

Default: `IPRINT = 0`.

IPRINT Action

0 No printing is performed.

1 Printing is performed on the final invocation.

2 Printing is performed after each step and on the final invocation.

SCALE — Vector of length `NVAR` containing the initial diagonal entries in `COV`. (Output, if `INVOKE = 0` or `1`; input, if `INVOKE = 2` or `3`)

HIST — Vector of length `NVAR` containing the recent history of variables. (Output, if `INVOKE = 0` or `1`; input/output, otherwise)

HIST(I) Meaning

$k > 0$ `I`-th variable was added to the model during the k -th step.

$k < 0$ `I`-th variable was deleted from the model during the k -th step.

0 `I`-th variable has never been in the model.

0.5 `I`-th variable was added into the model during initialization.

IEND — Completion indicator. (Output)

IEND Meaning

0 Additional steps may be possible.

1 No additional steps are possible.

LDCOEF — Leading dimension of exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDCOEF = size (COEF,1)`.

LDCOVS — Leading dimension of `COVS` exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDCOVS = size (COVS,1)`.

FORTRAN 90 Interface

Generic: CALL RSTEP (COV, NOBS, AOV, COEF, COVS [,...])

Specific: The specific interface names are S_RSTEP and D_RSTEP.

FORTRAN 77 Interface

Single: CALL RSTEP (INVOKE, NVAR, COV, LD COV, LEVEL, NFORCE, NSTEP, ISTEP, NOBS, PIN, POUT, TOL, IPRINT, SCALE, HIST, IEND, AOV, COEF, LD COEF, COVS, LD COVS)

Double: The double precision name is DRSTEP.

Example 1

Both examples use a data set from Draper and Smith (1981, pages 629–630). A corrected sum of squares and crossproducts matrix for this data is given in the DATA statement and can be computed using routine CORVC in Chapter 3, “Correlation”. The first four columns are for the independent variables and the last column is for the dependent variable. Here, RSTEP is invoked using the backward stepping option.

```
USE RSTEP_INT
INTEGER   LD COEF, LD COV, LD COVS, NVAR
PARAMETER (NVAR=5, LD COEF=NVAR, LD COV=NVAR, LD COVS=NVAR)
!
INTEGER   IEND, IPRINT, LEVEL(NVAR), NOBS
REAL      AOV(13), COEF(LD COEF,5), COV(LD COV,NVAR), &
          COVS(LD COVS,NVAR), HIST(NVAR), SCALE(NVAR)
!
DATA COV/415.231, 251.077, -372.615, -290.000, 775.962, 251.077, &
     2905.69, -166.538, -3041.00, 2292.95, -372.615, -166.538, &
     492.308, 38.0000, -618.231, -290.000, -3041.00, 38.0000, &
     3362.00, -2481.70, 775.962, 2292.95, -618.231, -2481.70, &
     2715.76/
DATA LEVEL/4*1, -1/
!
NOBS    = 13
IPRINT  = 2
CALL RSTEP (COV, NOBS, AOV, COEF, COVS, IPRINT=IPRINT)
!
END
```

Output

```
BACKWARD ELIMINATION
STEP 0:  4 variable(s) entered.

Dependent  R-squared   Adjusted   Est. Std. Dev.
Variable   (percent)   R-squared  of Model Error
         5         98.238         97.356         2.446

* * * Analysis of Variance * * *
```


Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Regression	4	2667.9	667.0	111.480	0.0000
Error	8	47.9	6.0		
Total	12	2715.8			

* * * Inference on Coefficients * * *

(Conditional on the Selected Model)

Variable	Coef. Estimate	Standard Error	t-statistic	Prob. of Larger t	Variance Inflation
1	1.551	0.7448	2.082	0.0709	38.5
2	0.510	0.7238	0.704	0.5012	254.4
3	0.102	0.7547	0.135	0.8963	46.9
4	-0.144	0.7091	-0.204	0.8437	282.5

STEP 1 : Variable 3 removed.

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error
5	98.234	97.645	2.309

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Regression	3	2667.8	889.3	166.835	0.0000
Error	9	48.0	5.3		
Total	12	2715.8			

* * * Inference on Coefficients * * *

(Conditional on the Selected Model)

Variable	Coef. Estimate	Standard Error	t-statistic	Prob. of Larger t	Variance Inflation
1	1.452	0.1170	12.410	0.0000	1.07
2	0.416	0.1856	2.242	0.0517	18.78
4	-0.237	0.1733	-1.365	0.2054	18.94

* * * Statistics for Variables Not in the Model * * *

Variable	Coef. Estimate	Standard Error	t-statistic to enter	Prob. of Larger t	Variance Inflation
3	0.102	0.7547	0.135	0.8963	46.87

STEP 2 : Variable 4 removed.

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error
5	97.868	97.441	2.406

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Regression	2	2657.9	1328.9	229.502	0.0000
Error	10	57.9	5.8		
Total	12	2715.8			

* * * Inference on Coefficients * * *

(Conditional on the Selected Model)

Variable	Coef. Estimate	Standard Error	t-statistic	Prob. of Larger t	Variance Inflation
1	1.468	0.1213	12.105	0.0000	1.06
2	0.662	0.0459	14.442	0.0000	1.06

* * * Statistics for Variables Not in the Model * * *

Variable	Coef. Estimate	Standard Error	t-statistic to enter	Prob. of Larger t	Variance Inflation
3	0.250	0.1847	1.354	0.2089	3.14
4	-0.237	0.1733	-1.365	0.2054	18.94

* * * Backward Elimination Summary * * *

Variable	Step	Removed
3	1	
4	2	

Comments

- Workspace may be explicitly provided, if desired, by use of `R2TEP/DR2TEP`. The reference is:

```
CALL R2TEP (INVOKE, NVAR, COV, LDCOV, LEVEL, NFORCE, NSTEP,
ISTEP, NOBS, PIN, POUT, TOL, IPRINT, SCALE,HIST, IEND, AOV,
COEF, LDCOEF, COVS, LDCOVS, SWEPT, IWK)
```

The additional arguments are as follows:

SWEPT — Work vector of length `NVAR` with information to indicate the independent variables in the model. (Output)
`SWEPT(I) = 1.0` indicates that independent variable `I` is in the model. Otherwise, `SWEPT(I) = -1.0`. Routine `RSUBM` (page 251) can be called with the arguments `COVS` and `SWEPT` to obtain the part of `COVS` pertaining to the current model.

IWK — Integer work vector of length $2 * NVAR$.

- Informational errors

Type	Code	Description
3	1	Based on <code>TOL</code> , there are linear dependencies among the variables to be forced.
4	2	No variables entered the model. Elements of <code>AOV</code> are set to NaN.

Description

Routine `RSTEP` builds a multiple linear regression model using forward selection, backward selection, or forward stepwise (with a backward glance) selection. The routine `RSTEP` is designed so that the user can monitor, and perhaps change, the variables added (deleted) to (from) the model after each step. In this case, multiple calls to `RSTEP` (with `INVOKE = 1, 2, 2, ..., 3`) are made. Alternatively, `RSTEP` can be invoked once (with `INVOKE = 0`) in order to perform the stepping until a final model is selected.

Levels of priority can be assigned to the candidate independent variables. All variables with a priority level of 1 must enter the model before any variable with a priority level of 2. Similarly, variables with a level of 2 must enter before variables with a level of 3, etc.

Variables can also be forced into the model. If equal levels of priority are to be assumed, the levels of priority can all be set to 1.

Typically, the intercept is forced into all models and is not a candidate variable. In this case, a sum of squares and crossproducts matrix for the independent and dependent variables corrected for the mean is input for `COV`. Routine `CORVC` in Chapter 3, “Correlation” can be used to compute the corrected sum of squares and crossproducts. Routine `RORDM` in Chapter 19, “Utilities,” can be used to reorder this matrix, if required. Other possibilities are

1. The intercept is not in the model. A raw (uncorrected) sum of squares and crossproducts matrix for the independent and dependent variables is required for `COV`. `NOBS` must be set to one greater than the number of observations. IMSL routine `MXTXF` (IMSL MATH/LIBRARY) can be used to compute the raw sum of squares and crossproducts matrix.
2. An intercept is to be a candidate variable. A raw (uncorrected) sum of squares and crossproducts matrix for the constant regressor (= 1), independent and dependent variables is required for `COV`. In this case, `COV` contains one additional row and column corresponding to the constant regressor. This row/column contains the sum of squares and crossproducts of the constant regressor with the independent and dependent variables. The remaining elements in `COV` are the same as in the previous case. `NOBS` must be set to one greater than the number of observations.

The stepwise regression algorithm is due to Efroymsen (1960). Routine `RSTEP` uses sweeps of `COV` to move variables in and out of the model (Hemmerle 1967, Chapter 3). The `SWEEP` operator discussed by Goodnight (1979) is used. A description of the stepwise algorithm is given also by Kennedy and Gentle (1980, pages 335–340). The advantage of stepwise model building over all possible regressions (see routine `RBEST`, [page 231](#)) is that it is less demanding computationally when the number of candidate independent variables is very large. However, there is no guarantee that the model selected will be the best model (highest R^2) for any subset size of independent variables.

Additional Example

Example 2

This example uses the data set in Example 1. Here, `RSTEP` is invoked using the forward stepwise option.

```
USE RSTEP_INT
INTEGER   LDcoef, LDcov, LDcovs, Nvar
PARAMETER (Nvar=5, LDcoef=Nvar, LDcov=Nvar, LDcovs=Nvar)
!
INTEGER   Iend, Iprint, Istep, level(Nvar), NOBS
REAL     AOV(13), COEF(LDcoef,5), COV(LDcov,Nvar), &
         COVS(LDcovs,Nvar), HIST(Nvar), SCALE(Nvar)
!
```

```

DATA COV/415.231, 251.077, -372.615, -290.000, 775.962, 251.077, &
2905.69, -166.538, -3041.00, 2292.95, -372.615, -166.538, &
492.308, 38.0000, -618.231, -290.000, -3041.00, 38.0000, &
3362.00, -2481.70, 775.962, 2292.95, -618.231, -2481.70, &
2715.76/
DATA LEVEL/4*1, -1/
!
ISTEP = 1
NOBS = 13
IPRINT = 2
CALL RSTEP (COV, NOBS, AOV, COEF, COVS, ISTEP=ISTEP, IPRINT=IPRINT)
!
END

```

Output

FORWARD SELECTION
STEP 0: No variables entered.

```

* * * Statistics for Variables Not in the Model * * *

```

Variable	Coef. Estimate	Standard Error	t-statistic to enter	Prob. of Larger t	Variance Inflation
1	1.869	0.5264	3.550	0.0046	1
2	0.789	0.1684	4.686	0.0007	1
3	-1.256	0.5984	-2.098	0.0598	1
4	-0.738	0.1546	-4.775	0.0006	1

STEP 1 : Variable 4 entered.

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error
5	67.454	64.496	8.964

```

* * * Analysis of Variance * * *

```

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Regression	1	1831.9	1831.9	22.799	0.0006
Error	11	883.9	80.4		
Total	12	2715.8			

```

* * * Inference on Coefficients * * *
(Conditional on the Selected Model)

```

Variable	Coef. Estimate	Standard Error	t-statistic	Prob. of Larger t	Variance Inflation
4	-0.738	0.1546	-4.775	0.0006	1.00

```

* * * Statistics for Variables Not in the Model * * *

```

Variable	Coef. Estimate	Standard Error	t-statistic to enter	Prob. of Larger t	Variance Inflation
1	1.440	0.1384	10.403	0.0000	1.06
2	0.311	0.7486	0.415	0.6867	18.74
3	-1.200	0.1890	-6.348	0.0001	1.00

STEP 2 : Variable 1 entered.

Dependent Variable	R-squared	Adjusted R-squared	Est. Std. Dev.
--------------------	-----------	--------------------	----------------

Variable	(percent)	R-squared	of Model	Error
5	97.247	96.697		2.734

```

* * * Analysis of Variance * * *
Source          DF      Sum of Squares      Mean Square      Overall F      Prob. of
Regression      2      2641.0          1320.5          176.636        0.0000
Error           10       74.8            7.5
Total           12      2715.8

```

```

* * * Inference on Coefficients * * *
(Conditional on the Selected Model)
Variable      Coef.      Standard Error      t-statistic      Prob. of      Variance
              Estimate      Error      to enter      Larger t      Inflation
1              1.440      0.1384      10.403          0.0000          1.06
4             -0.614      0.0486     -12.622          0.0000          1.06

```

```

* * * Statistics for Variables Not in the Model * * *
Variable      Coef.      Standard Error      t-statistic      Prob. of      Variance
              Estimate      Error      to enter      Larger t      Inflation
2              0.416      0.1856          2.242          0.0517          18.78
3             -0.410      0.1992          -2.058          0.0697          3.46

```

```

* * * Forward Selection Summary * * *
Variable      Step Entered
1              2
4              1

```

Example 3

For an extended version of Example 2 that in addition computes the intercept and standard error for the final model from RSTEP, see “Example 2” for routine RSUBM (page 251).

GSWEP

Performs a generalized sweep of a row of a nonnegative definite matrix.

Required Arguments

KROW — Row/column number to be swept. (Input)

A — *N* by *N* nonnegative definite matrix whose row *KROW* is to be swept. (Input/Output)
Only the upper triangle of *A* is referenced.

Optional Arguments

N — Order of the matrix to be swept. (Input)
Default: *N* = size (*A*,2).

LDA — Leading dimension of **A** exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDA = size (A,1)`.

IREV — Reversibility option. (Input)

Default: `IREV = 0`.

IREV Action When Linear Dependence Is Declared

0 Elements of row and column `KROW` of **A** are set to 0.0. Reversibility of the generalized sweep operator is lost.

1 Elements of row and column `KROW` of **A** are left unchanged. Reversibility of the generalized sweep operator is maintained, but some post processing by the user is required. See Comments.

TOL — Tolerance used in determining linear dependence. (Input)

`TOL = 100 * AMACH(4)` is a common choice. See documentation for routine `AMACH` in the Reference Material.

Default: `TOL = 1.e-5` for single precision and `2.d-14` for double precision.

SCALE — Vector of length `N` containing the diagonal scaling matrix used in the tolerance check. (Input)

A common choice for `SCALE(I)` is the `I`-th diagonal element of **A** before any calls to `GSWEP` have been made. If `TOL = 0.0`, `SCALE` is not referenced and can be a vector of length one.

SWEPT — Vector of length `N` with information to indicate what has and has not been swept. (Input/Output)

On the first call to `GSWEP` all elements must equal `-1.0`. On output, `SWEPT(KROW) = 1.0` if the sweep was successful. If a linear dependence is declared, `SWEPT(KROW)` remains equal to `-1.0`.

FORTRAN 90 Interface

Generic: `CALL GSWEP (KROW, A [, ...])`

Specific: The specific interface names are `S_GSWEP` and `D_GSWEP`.

FORTRAN 77 Interface

Single: `CALL GSWEP (KROW, N, A, LDA, IREV, TOL, SCALE, SWEPT)`

Double: The double precision name is `DGSWEP`.

Example

We consider the correlation matrix for the first three regressors from the example used by Berk (1976) and discussed by Frane (1977). The matrix is “nearly” singular. The rows of the correlation matrix are swept sequentially with `KROW` equal 1, 2, 3. With a tolerance of 0.001, the sweeps for 1 and 2 are successful. When a sweep on row 3 is attempted a linear dependence is declared. This is because

$$1 - R_{1,2,3}^2 = 0.0001 < 0.001$$

```
USE IMSL_LIBRARIES
INTEGER    LDA, N
PARAMETER (N=3, LDA=N)
!
INTEGER    ISETNG, KROW
REAL      A(LDA,N), SCALE(N), SQRT, SWEPT(N), TOL
INTRINSIC SQRT
!
A(1,1) = 1.0
A(1,2) = SQRT(0.99)
A(1,3) = 0.1*SQRT(0.99)
A(2,2) = 1.0
A(2,3) = 0.0
A(3,3) = 1.0
TOL     = 0.001
!
                                Copy diagonal of A to SCALE.
CALL SCOPY (N, A(1:,1), LDA+1, SCALE, 1)
!
                                Initialize elements of SWEPT to -1.
SWEPT = -1.0
ISETNG = 4
CALL WROPT (-6, ISETNG, 1)
CALL WRRRN ('A', A, ITRING=1)
CALL WRRRN ('SWEPT', SWEPT)
SWEPT = -1.0
DO 10 KROW=1, 3
    CALL GSWEPT (KROW, A, TOL=TOL, SCALE=SCALE, SWEPT=SWEPT)
    CALL WRRRN ('A', A, ITRING=1)
    CALL WRRRN ('SWEPT', SWEPT)
10 CONTINUE
END
```

Output

```
          A
          1      2      3
1  1.00000  0.99499  0.09950
2          1.00000  0.00000
3                      1.00000
```

```
SWEPT
1  -1.00000
2  -1.00000
3  -1.00000
```

```
          A
          1      2      3
```

```

1  1.00000  0.99499  0.09950
2           0.01000 -0.09900
3                    0.99010

```

```

SWEPT
1  1.00000
2 -1.00000
3 -1.00000

```

```

          A
          1      2      3
1  100.000 -99.499  9.950
2           100.000 -9.900
3                    0.010

```

```

SWEPT
1  1.00000
2  1.00000
3 -1.00000

```

```

          A
          1      2      3
1  100.000 -99.499  0.000
2           100.000  0.000
3                    0.000

```

```

SWEPT
1  1.00000
2  1.00000
3 -1.00000

```

Comments

Say we wish to sweep k different rows of the matrix A . For purposes of discussion, let these be rows 1, 2, ..., k of A . Partition A into its first k rows and columns and the remainder,

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

For a nonsingular A_{11} , successive invocations of `GSWEP` with A and `KROW` equal to 1, 2, ..., k yields

$$\begin{pmatrix} A_{11}^{-1} & A_{11}^{-1}A_{12} \\ \text{---} & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{pmatrix}$$

Only the elements in the upper triangle of A are referenced. Thus, the elements in the lower triangles of the symmetric matrices

$$A_{11}^{-1} \text{ and } A_{22} - A_{21}A_{11}^{-1}A_{12}$$

are not returned. For a singular A_{11} and `IREV` equal to zero, a symmetric g_2 inverse of A_{11} , denoted by

$$A_{11}^{s_2}$$

is used. For a singular A_{11} and IREV not equal to zero, the first k rows of the swept A are not the same as for the IREV equal to one case. However,

$$G = A_{11}^{s_2} \text{ and } H = A_{11}^{s_2} A_{12}$$

can be obtained from the output A as follows:

$$g_{ij} = \begin{cases} 0 & \text{if } s_i + s_j \leq 0 \\ a_{ij} & \text{if } s_i + s_j = 2 \text{ and } i \leq j \\ a_{ji} & \text{if } s_i + s_j = 2 \text{ and } i > j \end{cases}$$

and

$$h_{ij} = \begin{cases} 0 & \text{if } i = j \text{ and } s_i = -1 \\ 1 & \text{if } i = j \text{ and } s_i = 1 \\ 0 & \text{if } i \neq j \text{ and } s_i + s_j \neq 0 \\ a_{ij} & \text{if } i \leq j \text{ and } s_i + s_j = 0 \\ -a_{ji} & \text{if } i > j \text{ and } s_i + s_j = 0 \end{cases}$$

H is the Hermite canonical form (also referred to as the Hermite normal form or a rowechelon form) of A_{11} .

Description

Routine `GSWEEP` computes an upper triangular generalized sweep of a nonnegative definite matrix. The versatility of the SWEEP operator for statistical computations, in particular for regression computations, is discussed by Goodnight (1979).

Routine `GSWEEP` is based on `UTG2SWEEP` and `RUTG2SWEEP` described by Goodnight (1979, pages 157-158). (A misprint appears twice in "Step 5", page 157 of Goodnight's article. The " a_{ij} " should be replaced by " a_{ik} ".) The test for linear dependence is the same as that given by Clarke (1982).

RSUBM

Retrieves a symmetric submatrix from a symmetric matrix.

Required Arguments

A — NA by NA symmetric matrix. (Input)
Only the upper triangle of A is referenced.

$SWEPT$ — Vector of length NA. (Input)
Element $A(I, J)$ is included in submatrix A_{SUB} if and only if $SWEPT(I) > 0.0$ and $SWEPT(J) > 0.0$.

NASUB — Order of submatrix ASUB. (Output)

NASUB equals the number of elements in SWEPT that are greater than zero.

ASUB — NASUB by NASUB symmetric matrix containing a submatrix of A. (Output)

If A is not needed, ASUB and A can share the same storage locations.

Optional Arguments

NA — Order of matrix A. (Input)

Default: NA = size (A,2).

LDA — Leading dimension of A exactly as specified in the dimension statement of the calling program. (Input)

Default: LDA = size (A,1).

LDASUB — Leading dimension of ASUB exactly as specified in the dimension statement of the calling program. (Input)

Default: LDASUB = size (ASUB,1).

FORTRAN 90 Interface

Generic: CALL RSUBM (A, SWEPT, NASUB, ASUB [,...])

Specific: The specific interface names are S_RSUBM and D_RSUBM.

FORTRAN 77 Interface

Single: CALL RSUBM (NA, A, LDA, SWEPT, NASUB, ASUB, LDASUB)

Double: The double precision name is DRSUBM.

Example 1

The 2×2 symmetric submatrix ASUB is retrieved from rows and columns 1 and 4 of the 4×4 symmetric matrix A.

```
USE RSUBM_INT
USE WRRRN_INT
INTEGER    LDA, LDASUB, NA
PARAMETER (LDASUB=2, NA=4, LDA=NA)
!
INTEGER    NASUB
REAL       A(LDA,NA), ASUB(LDASUB,LDASUB), SWEPT(NA)
!
DATA SWEPT/1.0, -1.0, -1.0, 1.0/
DATA A/10.0, 20.0, 40.0, 70.0, 20.0, 30.0, 50.0, 80.0, 40.0, &
      50.0, 60.0, 90.0, 70.0, 80.0, 90.0, 100.0/
!
CALL RSUBM (A, SWEPT, NASUB, ASUB)
```

```
CALL WRRRN ('ASUB', ASUB)
END
```

Output

```
ASUB
   1   2
1  10.0  70.0
2  70.0 100.0
```

Comments

1. Workspace may be explicitly provided, if desired, by use of R2UBM/DR2UBM. The reference is:

```
CALL R2UBM (NA, A, LDA, SWEPT, NASUB, ASUB, LDASUB,
           IWK)
```

The additional argument is:

IWK — Vector of length NASUB.

2. Routine RSUBM can be used after invoking routines GSWEPT ([page 247](#)) and RSTEP ([page 237](#)) in order to retrieve the submatrix for the variables in the model.

Description

Routine RSUBM retrieves a symmetric submatrix from a symmetric matrix A . If elements i and j of the input vector SWEPT are greater than zero, then the ij -th element of A is output in the submatrix ASUB. Otherwise, the ij -th element of A will not be included in ASUB. (Here, $i = 1, 2, \dots, NA$, and $j = 1, 2, \dots, NA$, where NA is the order of A .)

Routine RSUBM can be useful in conjunction with two routines, GSWEPT ([page 247](#)) and RSTEP ([page 237](#)). The routine RSUBM can be used after routine GSWEPT in order to retrieve the submatrix of A that corresponds to the rows/columns that have been successfully swept. In this case, the SWEPT vector output from GSWEPT can be used as the input for the argument SWEPT in RSUBM. Also, RSUBM can be used after routine RSTEP in order to retrieve the submatrix of COVS that corresponds to the independent variables in the final model. In this case, the HIST vector output from RSTEP can be used as the input for the argument SWEPT in RSUBM.

Additional Example

Example 2

This example invokes RSUBM after routine RSTEP ([page 237](#)) in order to retrieve the submatrix of COVS that corresponds to the independent variables in the final stepwise model. With this submatrix, routine BLINF (IMSL MATH/LIBRARY) is used to compute the estimated standard deviation for the intercept in the final model.

A data set from Draper and Smith (1981, pages 629–630) is used. The means and the corrected sum of squares and crossproducts matrix for this data are given in the DATA statements. They

can be computed using routine CORVC in Chapter 3, "Correlation". The first four entries in XMEAN and the first four columns of COV correspond to the independent variables, the last entry in XMEAN and the last column of COV correspond to the dependent variable.

After RSTEP is invoked to obtain a model, the intercept is computed using the formula

$$\hat{\beta}_0 = \bar{y} - \sum_{i=1}^k \hat{\beta}_i \bar{x}_i$$

where k is the number of independent variables in the final model. The estimated standard deviation of the intercept is computed using the formula

$$\text{Est. St. Dev}(\hat{\beta}_0) = \sqrt{s^2(1/n + \bar{x}^T A \bar{x})}$$

where s^2 is the error mean square from the fit (stored in AOV(8)), n is the number of observations, \bar{x} is the subvector of means for the independent variables in the final model (in this case the first mean and the fourth mean), and A is the submatrix (in this case with rows and columns 1 and 4) of the matrix COVS that is output by RSTEP.

```

USE IMSL_LIBRARIES
INTEGER   LDcoef, LDcov, LDcovs, Nvar
PARAMETER (Nvar=5, LDcoef=Nvar, LDcov=Nvar, LDcovs=Nvar)
!
INTEGER   I, Iend, Invoke, Iprint, Istep, J, Level(Nvar), &
          Nforce, Nind, Nobs, Nout, Nstep
REAL      AOV(13), B0, COEF(LDcoef,5), &
          COV(LDcov,Nvar), COVS(LDcovs,Nvar), HIST(Nvar), PIN, &
          POUT, SCALE(Nvar), SEB0, SQRT, TOL, XMEAN(Nvar)
INTRINSIC SQRT
!
DATA COV/415.231, 251.077, -372.615, -290.000, 775.962, 251.077, &
      2905.69, -166.538, -3041.00, 2292.95, -372.615, -166.538, &
      492.308, 38.0000, -618.231, -290.000, -3041.00, 38.0000, &
      3362.00, -2481.70, 775.962, 2292.95, -618.231, -2481.70, &
      2715.76/
DATA XMEAN/7.46154, 48.1538, 11.7692, 30.0000, 95.4231/
DATA LEVEL/4*1, -1/
!
J = 0
Istep = 1
Nobs  = 13
Iprint = 1
CALL RSTEP (COV, Nobs, AOV, COEF, COVS, Nvar=Nvar, Istep=Istep,
            Iprint=Iprint, HIST=HIST)
!
                                Compute intercept
B0 = XMEAN(Nvar)
DO 10 I=1, Nvar - 1
  IF (HIST(I) .GT. 0.0) THEN
    B0      = B0 - XMEAN(I)*COEF(I,1)
    J      = J + 1
    XMEAN(J) = XMEAN(I)
  END IF
10 CONTINUE
!
                                Compute standard error of intercept

```

```

CALL RSUBM (COVS, HIST, NIND, COVS)
SEB0 = 1.0/NOBS + BLINF(COVS, XMEAN, XMEAN, NRA=NIND, NCA=NIND)
SEB0 = SQRT(AOV(8)*SEB0)
!
! Print intercept and standard error
CALL UMACH (2, NOUT)
WRITE (NOUT,99999) ' '
WRITE (NOUT,99999) 'Intercept ', B0
WRITE (NOUT,99999) 'Std. Error', SEB0
99999 FORMAT (1X, A, F10.3)
!
END

```

Output

FORWARD SELECTION

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error
5	97.247	96.697	2.734

```

* * * Analysis of Variance * * *
Source          DF      Sum of Squares      Mean Square      Overall F      Prob. of Larger F
Regression       2      2641.0      1320.5      176.636      0.0000
Error            10       74.8       7.5
Total            12     2715.8

```

```

* * * Inference on Coefficients * * *
(Conditional on the Selected Model)
Variable      Coef. Estimate      Standard Error      t-statistic      Prob. of Larger t      Variance Inflation
1             1.440      0.1384      10.403      0.0000      1.06
4            -0.614      0.0486     -12.622      0.0000      1.06

```

```

* * * Statistics for Variables Not in the Model * * *
Variable      Coef. Estimate      Standard Error      t-statistic      Prob. of Larger t      Variance Inflation
2             0.416      0.1856      2.242      0.0517      18.7
3            -0.410      0.1992     -2.058      0.0697      3.46

```

* * * Forward Selection Summary * * *

Variable	Step Entered
1	2
4	1

Intercept	103.097
Std. Error	2.124

RCURV

Fits a polynomial curve using least squares.

Required Arguments

XDATA — Vector of length *NOBS* containing the *x* values. (Input)

YDATA — Vector of length *NOBS* containing the *y* values. (Input)

B — Vector of length *NDEG* + 1 containing the coefficients

$$\hat{\beta}$$

(Output)

The fitted polynomial is

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 + \dots + \hat{\beta}_k x^k$$

Optional Arguments

NOBS — Number of observations. (Input)

Default: *NOBS* = size(*XDATA*,1).

NDEG — Degree of polynomial. (Input)

Default: *NDEG* = size(*B*,1) - 1.

SSPOLY — Vector of length *NDEG* + 1 containing the sequential sums of squares. (Output)

SSPOLY(1) contains the sum of squares due to the mean. For *i* = 1, 2, ..., *NDEG*,

SSPOLY(*i* + 1) contains the sum of squares due to x^i adjusted for the mean, x , x^2 , ..., and x^{i-1} .

STAT — Vector of length 10 containing statistics described below. (Output)

<i>i</i>	Statistics
1	Mean of <i>x</i>
2	Mean of <i>y</i>
3	Sample variance of <i>x</i>
4	Sample variance of <i>y</i>
5	<i>R</i> -squared (in percent)
6	Degrees of freedom for regression
7	Regression sum of squares
8	Degrees of freedom for error

***i* Statistics**

9 Error sum of squares

10 Number of data points (x, y) containing NaN (not a number) as a x or y value

FORTRAN 90 Interface

Generic: CALL RCURV (XDATA, YDATA, B [,...])

Specific: The specific interface names are S_RCURV and D_RCURV.

FORTRAN 77 Interface

Single: CALL RCURV (NOBS, XDATA, YDATA, NDEG, B, SSPOLY, STAT)

Double: The double precision name is DRCURV.

Example

A polynomial model is fitted to data discussed by Neter and Wasserman (1974, pages 279–285). The data set contains the response variable y measuring coffee sales (in hundred gallons) and the number of self-service coffee dispensers. Responses for fourteen similar cafeterias are in the data set.

```
USE RCURV_INT
USE WRRRN_INT
USE WRRRL_INT

INTEGER      NDEG, NOBS
PARAMETER   (NDEG=2, NOBS=14)

!
REAL        B(NDEG+1), SSPOLY(NDEG+1), STAT(10), XDATA(NOBS), &
            YDATA(NOBS)
CHARACTER   CLABEL(11)*15, RLABEL(1)*4

!
DATA RLABEL/'NONE'/, CLABEL/' ', 'Mean of X', 'Mean of Y', &
      'Variance X', 'Variance Y', 'R-squared', &
      'DF Reg.', 'SS Reg.', 'DF Error', 'SS Error', &
      'Pts. with NaN'/
DATA XDATA/0., 0., 1., 1., 2., 2., 4., 4., 5., 5., 6., 6., 7., &
      7./
DATA YDATA/508.1, 498.4, 568.2, 577.3, 651.7, 657.0, 755.3, &
      758.9, 787.6, 792.1, 841.4, 831.8, 854.7, 871.4/

!
CALL RCURV (XDATA, YDATA, B, SSPOLY=SSPOLY, STAT=STAT)

!
CALL WRRRN ('B', B, 1, NDEG+1, 1)
CALL WRRRN ('SSPOLY', SSPOLY, 1, NDEG+1, 1)
CALL WRRRL ('%/STAT', STAT, RLABEL, CLABEL, 1, 10, 1, FMT='(2W10.4)')
END
```

Output

```
      B
      1      2      3
503.3  78.9  -4.0
```

```
      SSPOLY
      1      2      3
7077152.0  220644.2  4387.7
```

```
      STAT
Mean of X  Mean of Y  Variance X  Variance Y  R-squared  DF Reg.
   3.571    711.0     6.418    17364.8    99.69      2
```

```
SS Reg.  DF Error  SS Error  Pts. with NaN
225031.9    11      710.5      0
```

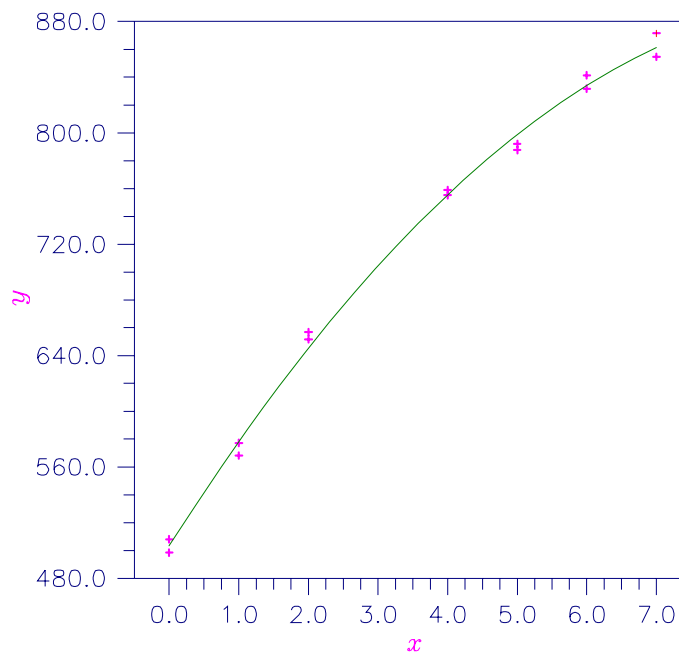


Figure 2-7 Plot of Data and Second Degree Polynomial Fit

Comments

1. Workspace may be explicitly provided, if desired, by use of R2URV/DR2URV. The reference is:

```
CALL R2URV (NOBS, XDATA, YDATA, NDEG, B, SSPOLY,
           STAT, WK, IWK)
```

The additional arguments are as follows:

WK — Work vector of length $11 * \text{NOBS} + 11 * \text{NDEG} + 5 + (\text{NDEG} + 1) * (\text{NDEG} + 3)$.

IWK — Work vector of length **NOBS**.

2. Informational errors

Type	Code	Description
4	3	Each (x, y) point contains NaN (not a number). There are no valid data.
4	7	The x values are constant. At least $\text{NDEG} + 1$ distinct x values are needed to fit a NDEG polynomial.
3	4	The y values are constant. A zero order polynomial is fit. High order coefficients are set to zero.
3	5	There are too few observations to fit the desired degree polynomial. High order coefficients are set to zero.
3	6	A perfect fit was obtained with a polynomial of degree less than NDEG . High order coefficients are set to zero.

3. If NDEG is greater than 10, the accuracy of the results may be questionable.

Description

Routine **RCURV** computes estimates of the regression coefficients in a polynomial (curvilinear) regression model. In addition to the computation of the fit, **RCURV** computes some summary statistics. Sequential sums of squares attributable to each power of the independent variable (stored in **SSPOLY**) are computed. These are useful in assessing the importance of the higher order powers in the fit. Draper and Smith (1981, pages 101–102) and Neter and Wasserman (1974, pages 278–287) discuss the interpretation of the sequential sums of squares. The statistic R^2 (stored in **STAT(5)**) is the percentage of the sum of squares of y about its mean explained by the polynomial curve. Specifically,

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} 100\%$$

where

$$\hat{y}_i$$

is the fitted y value at x_i and

$$\bar{y}$$

(stored in **STAT(2)**) is the mean of y . This statistic is useful in assessing the overall fit of the curve to the data. R^2 must be between 0% and 100%, inclusive. $R^2 = 100\%$ indicates a perfect fit to the data.

Routine **RCURV** computes estimates of the regression coefficients in a polynomial model using orthogonal polynomials as the regressor variables. This reparameterization of the polynomial model in terms of orthogonal polynomials has the advantage that the loss of accuracy resulting from forming powers of the x -values is avoided. All results are returned to the user for the original model.

The routine `RCURV` is based on the algorithm of Forsythe (1957). A modification to Forsythe's algorithm suggested by Shampine (1975) is used for computing the polynomial coefficients. A discussion of Forsythe's algorithm and Shampine's modification appears in Kennedy and Gentle (1980, pages 342–347).

RPOLY

Analyzes a polynomial regression model.

Required Arguments

X — `NOBS` by `NCOL` matrix containing the data. (Input)

IRSP — Column number `IRSP` of *X* contains the data for the response (dependent) variable. (Input)

IND — Column number `IND` of *X* contains the data for the independent (explanatory) variable. (Input)

MAXDEG — Maximum degree of polynomial to be fit. (Input)

NDEG — Degree of final polynomial regression. (Output)

COEF — `NDEG + 1` by 4 matrix containing statistics relating to the coefficients of the polynomial model. (Output)

Row 1 corresponds to the intercept. Row $1 + i$ corresponds to the coefficient of x^i . The columns are described as follows:

Col. Description

1 Estimated coefficient

$$\hat{\beta}$$

2 Estimated standard error of the estimated coefficient

3 *t*-statistic for the test the coefficient is zero

4 *p*-value for the two-sided *t* test

Optional Arguments

NOBS — Number of observations. (Input)
Default: `NOBS = size(x,1)`.

NCOL — Number of columns in *X*. (Input)
Default: `NCOL = size(x,2)`.

LDX — Leading dimension of X exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDX = \text{size}(X,1)$.

IFRQ — Frequency option. (Input)
 $IFRQ = 0$ means that all frequencies are 1.0. For positive $IFRQ$, column number $IFRQ$ of X contains the frequencies. If $X(i, IFRQ) = 0.0$, none of the remaining elements of row i of X are referenced, and updating of statistics is skipped for row i .
Default: $IFRQ = 0$.

IWT — Weighting option. (Input)
 $IWT = 0$ means that all weights are 1.0. For positive IWT , column number IWT of X contains the weights, and the computed prediction interval uses $AOV(8) = X(i, IWT)$ for the estimated variance of a future response.
Default: $IWT = 0$.

IPRED — Prediction interval option. (Input)
 $IPRED = 0$ means that prediction intervals are desired for a single future response. For positive $IPRED$, column number $IPRED$ of X contains the number of future responses for which a prediction interval is desired on the average of the future responses.
Default: $IPRED = 0$.

CONPCM — Confidence level for two-sided interval estimates on the mean in percent. (Input)
Default: $CONPCM = 95.0$.

CONPCP — Confidence level for two-sided prediction intervals in percent. (Input)
Default: $CONPCP = 95.0$.

ICRIT — Criterion option. (Input)
Default: $ICRIT = 0$.

ICRIT	Meaning
--------------	----------------

- | | |
|---|---|
| 0 | Fit a $MAXDEG$ -th degree polynomial. |
| 1 | Fit the lowest degree polynomial with an R^2 (in percent) of at least $CRIT$. |
| 2 | Fit the lowest degree polynomial with a lack-of-fit F test not significant at level $CRIT$ percent. |

CRIT — Criterion in percent. (Input, if $ICRIT = 1$ or $ICRIT = 2$, not referenced if $ICRIT = 0$)
Default: $CRIT = 95.0$.

ICRIT Meaning of CRIT

- 1 R^2 (in percent) that the fitted polynomial must achieve. A common choice is 95.0.
- 2 Significance level (in percent) for the lack-of-fit test that the fitted polynomial must not exceed. A common choice is 5.0.

LOF— Lack of fit option. (Input)
If $ICRIT = 2$, LOF must equal 1.
Default: $LOF = 0$.

LOF Action

- 0 $TLOF$ is not computed.
- 1 $TLOF$ is computed.

IPRINT— Printing option. (Input)
Default: $IPRINT = 0$.

IPRINT Action

- 0 No printing is performed.
- 1 AOV , $SQSS$, $COEF$, $TLOF$ are printed.
- 2 AOV , $SQSS$, $COEF$, $TLOF$, unusual cases in $CASE$ and plots of the data, and the fitted polynomial are printed.
- 3 AOV , $SQSS$, $COEF$, $TLOF$, $CASE$, plots of the data, the fitted polynomial, and the residuals are printed.

AOV— Vector of length 15 that contains statistics relating to the analysis of variance.
(Output)

***i* $AOV(i)$**

- 1 Degrees of freedom for the model
- 2 Degrees of freedom for error
- 3 Total (corrected) degrees of freedom
- 4 Sum of squares for the model
- 5 Sum of squares for error

<i>i</i>	AOV(<i>i</i>)
6	Total (corrected) sum of squares
7	Model mean square
8	Error mean square
9	Overall <i>F</i> -statistic
10	<i>p</i> -value
11	R^2 (in percent)
12	Adjusted R^2 (in percent)
13	Estimate of the standard deviation
14	Overall response mean
15	Coefficient of variation (in percent)

SQSS — NDEG by 4 matrix containing sequential statistics for the polynomial model.
(Output)

Row *i* corresponds to x^i ($i = 1, 2, \dots, \text{NDEG}$). The columns are described as follows:

Col. Description

1	Degrees of freedom
2	Sum of squares
3	<i>F</i> -statistic
4	<i>p</i> -value

LDSQSS — Leading dimension of SQSS exactly as specified in the dimension statement in the calling program. (Input)

Default: LDSQSS = size (SQSS,1).

LDCOEF — Leading dimension of COEF exactly as specified in the dimension statement in the calling program. (Input)

Default: LDCOEF = size (COEF,1).

TLOF — NDEG by 4 matrix containing tests of lack of fit for each degree of the polynomial.
(Output, if LOF = 1)

Row *i* corresponds to x^i ($i = 1, 2, \dots, \text{NDEG}$). The columns are described as follows:

Col. Description

- 1 Degrees of freedom
- 2 Lack-of-fit sum of squares
- 3 F test for lack of fit of the polynomial model of degree i
- 4 p -value for the F test

If $\text{LOF} = 0$, TLOF is not referenced and can be a 1 by 1 array.

LDTLOF — Leading dimension of TLOF exactly as specified in the dimension statement in the calling program. (Input)

Default: $\text{LDTLOF} = \text{size}(\text{TLOF}, 1)$.

CASE — NOBS by 12 matrix containing the case statistics. (Output)
Columns 1 through 12 contain the following:

Col. Description

- 1 Observed response
- 2 Predicted response
- 3 Residual
- 4 Leverage
- 5 Standardized residual
- 6 Jackknife residual
- 7 Cook's distance
- 8 DFFITS
- 9, 10 Confidence interval on the mean
- 11, 12 Prediction interval

LDCASE — Leading dimension of CASE exactly as specified in the dimension statement in the calling program. (Input)

Default: $\text{LDCASE} = \text{size}(\text{CASE}, 1)$.

NRMIS — Number of rows of CASE containing NaN (not a number). (Output)

FORTRAN 90 Interface

Generic: CALL RPOLY (X, IRSP, IND, MAXDEG, NDEG, COEF [,...])

Specific: The specific interface names are S_RPOLY and D_RPOLY.

FORTRAN 77 Interface

Single: CALL RPOLY (NOBS, NCOL, X, LDX, IRSP, IND, IFRQ, IWT, IPRED, CONPCM, CONPCP, MAXDEG, ICRT, CRIT, LOF, IPRINT, NDEG, AOV, SQSS, LDSQSS, COEF, LDCOEF, TLOF, LDTLOF, CASE, LDCASE, NRMISS)

Double: The double precision name is DRPOLY.

Example

A polynomial model is fitted to data discussed by Neter and Wasserman (1974, pages 279–285). The data set contains the response variable y measuring coffee sales (in hundred gallons) and the number of self-service coffee dispensers. Responses for fourteen similar cafeterias are in the data set. Some of the cafeterias have the same number of dispensers so that lack of fit of the model can be assessed.

```
USE RPOLY_INT
INTEGER    LDCASE, LDCOEF, LDSQSS, LDTLOF, LDX, MAXDEG, NCOL, &
           NOBS
PARAMETER (MAXDEG=2, NCOL=2, NOBS=14, LDCASE=NOBS, &
           LDCOEF=MAXDEG+1, LDSQSS=MAXDEG, LDTLOF=MAXDEG, &
           LDX=NOBS)
!
INTEGER    ICRT, IFRQ, IND, IPRED, IPRINT, IRSP, IWT, LOF, &
           NDEG, NRMISS
REAL      AOV(15), CASE(LDCASE,12), COEF(LDCOEF,4), CONPCM, &
           CONPCP, CRIT, SQSS(LDSQSS,4), TLOF(LDTLOF,4), &
           X(LDX,NCOL)
!
DATA (X(1,J),J=1,2) /0.0, 508.1/
DATA (X(2,J),J=1,2) /5.0, 787.6/
DATA (X(3,J),J=1,2) /0.0, 498.4/
DATA (X(4,J),J=1,2) /1.0, 568.2/
DATA (X(5,J),J=1,2) /2.0, 651.7/
DATA (X(6,J),J=1,2) /7.0, 854.7/
DATA (X(7,J),J=1,2) /2.0, 657.0/
DATA (X(8,J),J=1,2) /4.0, 755.3/
DATA (X(9,J),J=1,2) /6.0, 831.8/
DATA (X(10,J),J=1,2) /4.0, 758.9/
DATA (X(11,J),J=1,2) /5.0, 792.1/
DATA (X(12,J),J=1,2) /6.0, 841.4/
DATA (X(13,J),J=1,2) /7.0, 871.4/
DATA (X(14,J),J=1,2) /1.0, 577.3/
!
IRSP      = 2
IND       = 1
```

```

LOF      = 1
IPRINT  = 1
CALL RPOLY (X, IRSP, IND, MAXDEG, NDEG, COEF, LOF=LOF, IPRINT=IPRINT, &
          AOV=AOV, SQSS=SQSS, TLOF=TLOF, CASE=CASE, NRMISS=NRMISS)
!
END

```

Output

R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
99.685	99.628	8.037	711.0	1.13

```

* * * Analysis of Variance * * *

```

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Regression	2	225031.9	112515.9	1741.748	0.0000
Residual	11	710.6	64.6		
Corrected Total	13	225742.5			

```

* * * Inference on Coefficients * * *

```

Coef.	Estimate	Standard Error	t-statistic	Prob. of Larger t
1	503.3	4.791	105.054	0.0000
2	78.9	3.453	22.865	0.0000
3	-4.0	0.482	-8.242	0.0000

```

* * * Sequential Statistics * * *

```

Degree of Polynomial	Degrees of Freedom	Sum of Squares	F-statistic	Prob. of Larger F
1	1	220644.1	3415.574	0.0000
2	1	4387.7	67.922	0.0000

```

* * * Tests of Lack of Fit * * *

```

Degree of Polynomial	Degrees of Freedom	Sum of Squares	F-statistic	Prob. of Larger F
1	5	4793.7	22.031	0.0004
2	4	406.0	2.332	0.1547

Comments

1. Workspace may be explicitly provided, if desired, by use of R2OLY/DR2OLY. The reference is:

```

CALL R2OLY (NOBS, NCOL, X, LDX, IRSP, IND, IFRQ,
          IWT, IPRED, CONPCM, CONPCP, MAXDEG, ICRT, CRIT, LOF, IPRINT,
          NDEG, AOV, SQSS, LDSQSS, COEF, LDcoef, TLOF, LDTLOF, CASE,
          LDCASE, NRMISS, WK, IWK)

```

The additional arguments are as follows:

WK — Work vector of length $\text{MAXDEG}^2 + 8 * \text{MAXDEG} + 8 * \text{NOBS} + 5$

IWK — Work vector of length NOBS.

2.	Informational errors	
	Type	Code
	4	1 An invalid weight is encountered. Weights must be nonnegative.
	4	2 An invalid frequency is encountered. Frequencies must be nonnegative.
	4	7 The independent variable is constant. At least two distinct settings of the independent variable are needed.
	4	8 The number of future observations for a prediction interval must be positive.
	3	4 The response is constant. A zero degree polynomial is fit.
	3	5 There are too few observations to fit the desired degree polynomial. NDEG is set to one less than the number of valid observations.
	3	6 A perfect fit to the data was obtained with a polynomial of lower degree than MAXDEG.

Description

Routine `RPOLY` computes estimates of the regression coefficients in a polynomial (curvilinear) regression model. The degree of the polynomial can be specified, or the degree of the polynomial can be determined by `RPOLY` under one of two criteria:

1. If some of the x settings are repeated, the lowest degree polynomial can be fit whose lack of fit is not significant at a specified level.
2. The lowest degree polynomial can be fitted with an R^2 that meets a specified lower bound.

In addition to the computation of the fit, `RPOLY` computes and prints summary statistics (analysis of variance, sequential sums of squares, t tests for the coefficients, tests for lack of fit), case statistics (diagnostics for individual cases, confidence and prediction intervals), and plots (data, fitted data, and residuals).

Routine `RPOLY` computes estimates of the regression coefficients in a polynomial regression model using orthogonal polynomials. The reparameterization of the polynomial model in terms of orthogonal polynomials has the advantage that the loss of accuracy resulting from forming powers of the x settings is avoided. All results are returned to the user for the original model.

Often a predicted value and a confidence interval are desired for a setting of the independent variable not used in computing the regression fit. This is accomplished by including an extra row in the data matrix with the desired setting of the independent variable and with the response set equal to NaN (not a number). NaN can be retrieved by `AMACH(6)`, which is documented in the Reference Material. The row of the data matrix containing NaN will be omitted from the computations for determining the regression fit, and a prediction and a confidence interval for the missing response will be computed from the given setting of the independent variable.

Routine `RPOLY` is based on the algorithm of Forsythe (1957). A modification to Forsythe's algorithm suggested by Shampine (1975) is used for computing the polynomial coefficients. A discussion of Forsythe's algorithm and Shampine's modification appears in Kennedy and Gentle (1980, pages 342–347). A modification to Forsythe's algorithm is made for the inclusion of weights (Kelly 1967, page 68).

RCOMP

Generates an orthogonal central composite design.

Required Arguments

XMIN — Vector of length *NVAR* with the minimum values. (Input)
XMIN(*i*) is the minimum for the *i*-th variable.

XMAX — Vector of length *NVAR* with the maximum values. (Input)
XMAX(*i*) is the maximum for the *i*-th variable.

NCENTR — Number of center points. (Input)
NCENTR must be greater than 0.

X — *NPTS* by *NVAR* matrix containing the orthogonal central composite design. (Output)
Design settings for variable *I* are contained in column *I* of *X*. (*I* = 1, 2, ..., *NVAR*)

Optional Arguments

NVAR — Number of explanatory variables. (Input)
NVAR must be greater than or equal to 2 and less than or equal to 12.
Default: *NVAR* = size (*XMIN*,1).

IFREP — Option for the fractional replicate of the 2^{NVAR} design selected. (Input)
IFREP is referenced only if *NVAR* is greater than or equal to 5. In the following table, the design points in the fractional replicate part of the design are defined using modulo 2 arithmetic. Each variable is coded 0 or 1 to represent the low and high values of the variable.
Default: *IFREP* = 0.

NVAR **Defining Equation(s)**

$$5 \quad x_1 + x_2 + x_3 + x_4 + x_5 = \begin{cases} 0 & \text{if } IFREP = 0 \\ 1 & \text{if } IFREP = 1 \end{cases}$$

$$6 \quad x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = \begin{cases} 0 & \text{if } IFREP = 0 \\ 1 & \text{if } IFREP = 1 \end{cases}$$

$$7 \quad x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = \begin{cases} 0 & \text{if } IFREP = 0 \\ 1 & \text{if } IFREP = 1 \end{cases}$$

$$\begin{aligned}
8 \quad & \begin{pmatrix} x_1 + x_2 + x_3 + x_4 + x_5, \\ x_4 + x_5 + x_6 + x_7 + x_8 \end{pmatrix} = \begin{cases} (0,0) & \text{if IFREP} = 0 \\ (0,1) & \text{if IFREP} = 1 \\ (1,0) & \text{if IFREP} = 2 \\ (1,1) & \text{if IFREP} = 3 \end{cases} \\
9 \quad & \begin{pmatrix} x_1 + x_2 + x_3 + x_4 + x_5 + x_6, \\ x_4 + x_5 + x_6 + x_7 + x_8 + x_9 \end{pmatrix} = \begin{cases} (0,0) & \text{if IFREP} = 0 \\ (0,1) & \text{if IFREP} = 1 \\ (1,0) & \text{if IFREP} = 2 \\ (1,1) & \text{if IFREP} = 3 \end{cases} \\
10 \quad & \begin{pmatrix} x_1 + x_2 + x_3 + x_4 + x_5 + x_6, \\ x_1 + x_2 + x_3 + x_7 + x_8 + x_9, \\ x_1 + x_2 + x_4 + x_5 + x_7 + x_8 + x_{10} \end{pmatrix} = \begin{cases} (0,0,0) & \text{if IFREP} = 0 \\ (0,0,1) & \text{if IFREP} = 1 \\ \vdots & \\ (1,1,1) & \text{if IFREP} = 7 \end{cases} \\
11 \quad & \begin{pmatrix} x_1 + x_2 + x_3 + x_4 + x_5 + x_6, \\ x_1 + x_2 + x_6 + x_9 + x_{10}, \\ x_1 + x_5 + x_6 + x_7 + x_{10} + x_{11}, \\ x_1 + x_3 + x_5 + x_8 + x_{11} \end{pmatrix} = \begin{cases} (0,0,0,0) & \text{if IFREP} = 0 \\ (0,0,0,1) & \text{if IFREP} = 1 \\ \vdots & \\ (1,1,1,1) & \text{if IFREP} = 15 \end{cases} \\
12 \quad & \begin{pmatrix} x_1 + x_2 + x_3 + x_4 + x_9 + x_{10}, \\ x_1 + x_2 + x_5 + x_6 + x_9 + x_{11}, \\ x_1 + x_4 + x_5 + x_7 + x_{11} + x_{12}, \\ x_1 + x_2 + x_7 + x_8 + x_{10} + x_{11} \end{pmatrix} = \begin{cases} (0,0,0,0) & \text{if IFREP} = 0 \\ (0,0,0,1) & \text{if IFREP} = 1 \\ \vdots & \\ (1,1,1,1) & \text{if IFREP} = 15 \end{cases}
\end{aligned}$$

NPTS — Number of design points. (Output)

NVAR	NPTS
2 thru 4	$2^{\text{NVAR}} + 2 * \text{NVAR} + \text{NCENTR}$
5 thru 7	$2^{\text{NVAR}-1} + 2 * \text{NVAR} + \text{NCENTR}$
8 or 9	$2^{\text{NVAR}-2} + 2 * \text{NVAR} + \text{NCENTR}$
10	$2^{\text{NVAR}-3} + 2 * \text{NVAR} + \text{NCENTR}$
11 or 12	$2^{\text{NVAR}-4} + 2 * \text{NVAR} + \text{NCENTR}$

LDX — Leading dimension of X exactly as specified in the dimension statement in the calling program. (Input)

Default: LDX = size (X,1).

FORTRAN 90 Interface

Generic: CALL RCOMP (XMIN, XMAX, NCENTR, X [, ...])

Specific: The specific interface names are S_RCOMP and D_RCOMP.

FORTRAN 77 Interface

Single: CALL RCOMP (NVAR, XMIN, XMAX, NCENTR, IFREP, NPTS, X, LDX)

Double: The double precision name is DRCOMP.

Example

This example uses two variables and their respective minimum and maximum values to generate an orthogonal central composite design with four center points.

```
USE RCOMP_INT
USE UMACH_INT
USE WRRRN_INT

PARAMETER (NVAR=2, NCENTR=4, LDX=2**NVAR+2*NVAR+NCENTR)
REAL      X(LDX,NVAR), XMAX(NVAR), XMIN(NVAR)
DATA      XMIN /251.0,73.0/ XMAX/295.0, 87.0/
!
CALL RCOMP (XMIN, XMAX, NCENTR, X, NPTS=NPTS)
!
CALL UMACH (2, NOUT)
WRITE (NOUT,*) 'NPTS = ', NPTS
CALL WRRRN ('X', X)
END
```

Output

NPTS = 12

	X	
	1	2
1	291.2	85.8
2	291.2	74.2
3	254.8	85.8
4	254.8	74.2
5	273.0	80.0
6	273.0	80.0
7	273.0	80.0
8	273.0	80.0
9	251.0	80.0
10	295.0	80.0
11	273.0	73.0
12	273.0	87.0

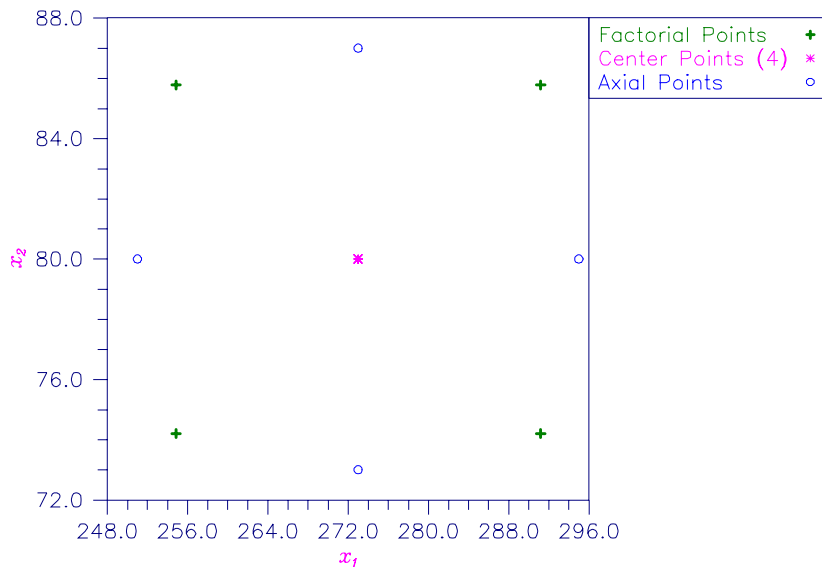


Figure 2-8 Orthogonal Central Composite Design With Four Center Points

Description

Routine `RCOMP` generates an orthogonal central composite design from the minimum and maximum value for each of n (input in `NVAR`) variables, where $2 \leq n \leq 12$. An orthogonal central composite design is a 2^{-k} replicate of a 2^n factorial design, i.e., a 2^{n-k} fractional factorial, augmented by $2n$ axial points and m (input in `NCENTR`) center points. The values of n and k used by `RCOMP` are given by the following table:

N	k
2, 3, 4	0
5, 6, 7	1
8, 9	2
10	3
11, 12	4

The fractional factorial part of all designs generated by `RCOMP` are of resolution V or greater. This means the fractions allow the overall mean, all the main effects, and all the two-factor interactions to be estimated. For a further discussion, see John (1971, pages 148–157).

Experimental designs for fitting a second-order response surface must contain at least three levels of each variable in order for the regression coefficients to be estimated. Orthogonal central composite designs provide a useful alternative to the 3^n factorial design, which can require an excessive number of design points. On a *per observation basis*, the orthogonal central composite design is no worse than the 3^n factorial design with regard to efficiency for estimating the regression coefficients of the square and crossproduct variables (see Meyers

1971, pages 134–136). The design assumes three factor and higher-way interactions are negligible.

Meyers (1971, chapter 7) and John (1971, pages 204–206) discuss the generation of the design. The number of design points (stored in `NPTS`) is

$2^{n-k} + 2n + m$. Each variable in the design appears at five different levels. For a second-order response surface model with the x variables coded $\{-\alpha, -1, 0, 1, \alpha\}$ and with pure quadratic terms corrected for the mean

$$c = \frac{2^{n-k} + 2\alpha^2}{2^{n-k} + 2n + m}$$

the design produces a diagonal $X^T X$ matrix. Let

$$\alpha = \left(\frac{\left(\sqrt{2^{n-k} + 2n + m} - \sqrt{2^{n-k}} \right)^2 2^{n-k}}{4} \right)^{1/4}$$

and let the minimum and maximum value of the j -th variable be denoted by x_{1j} and x_{2j} , respectively. The following table gives the formulas for the coded and decoded variable settings:

Coded Setting for Variable j	Decoded Setting for Variable j
$-\alpha$	x_{1j}
-1	$\frac{x_{1j} - x_{2j}}{2\alpha} + \frac{x_{1j} + x_{2j}}{2}$
0	$\frac{x_{1j} + x_{2j}}{2}$
1	$\frac{x_{2j} - x_{1j}}{2\alpha} + \frac{x_{1j} + x_{2j}}{2}$
α	x_{2j}

RFORP

Fits an orthogonal polynomial regression model.

Required Arguments

X — NOBS by NCOL matrix containing the data. (Input)

IRSP — Column number *IRSP* of X contains the data for the response (dependent) variable. (Input)

IND — Column number **IND** of **X** contains the data for the independent (explanatory) variable. (Input)

MAXDEG — Maximum degree of polynomial to be fit. (Input)

NDEG — Degree of final polynomial regression. (Output)

A — Vector of length **MAXDEG** containing constants used to generate orthogonal polynomials. (Output)
Only the first **NDEG** elements of **A** are referenced.

B — Vector of length **MAXDEG** containing constants used to generate orthogonal polynomials. (Output)
Only the first **NDEG** elements of **B** are referenced.

SCOEF — Vector of length $1 + \text{MAXDEG}$ containing the regression coefficients α of the fitted model using the scaled version of $x(z)$. (Output)
Only the first $1 + \text{NDEG}$ elements of **SCOEF** are referenced.

$$\hat{\alpha}_0 = \text{SCOEF}(1)$$

is the estimated intercept and equals the response mean.

$$\hat{\alpha}_i = \text{SCOEF}(1+i)$$

contains the estimated coefficient for the i -th order orthogonal polynomial using the scaled version of $x(z)$.

D — Vector of length $\text{MAXDEG} + 1$ containing the diagonal elements of the (diagonal) sums of squares and crossproducts matrix. (Output)
The sum of squares due to the i -th degree orthogonal polynomial is given by

$$D(i+1) * \hat{\alpha}_i^2$$

Only the first $\text{NDEG} + 1$ elements of **D** are referenced.

Optional Arguments

NOBS — Number of observations. (Input)
Default: **NOBS** = size (**X**,1).

NCOL — Number of columns in **X**. (Input)
Default: **NCOL** = size (**X**,2).

LDX — Leading dimension of **X** exactly as specified in the dimension statement in the calling program. (Input)
Default: **LDX** = size (**X**,1).

IFRQ — Frequency option. (Input)

$IFRQ = 0$ means that all frequencies are 1.0. For positive $IFRQ$, column number $IFRQ$ of X contains the frequencies. If $X(i, IFRQ) = 0.0$, none of the remaining elements of row i of X are referenced, and updating of statistics is skipped for row i .

Default: $IFRQ = 0$.

IWT — Weighting option. (Input)

$IWT = 0$ means that all weights are 1.0. For positive IWT , column number IWT of X contains the weights.

Default: $IWT = 0$.

ICRIT — Criterion option. (Input)

Default: $ICRIT = 0$.

ICRIT	Meaning
--------------	----------------

0	Fit a $MAXDEG$ -th degree polynomial.
---	---------------------------------------

1	Fit the lowest degree polynomial with an R^2 (in percent) of at least $CRIT$.
---	--

2	Fit the lowest degree polynomial with a lack-of-fit F test not significant at level $CRIT$ percent.
---	---

CRIT — Criterion in percent. (Input, if $ICRIT = 1$ or $ICRIT = 2$)

Default: $CRIT = 95.0$.

ICRIT	Meaning of CRIT
--------------	------------------------

1	R^2 (in percent) that the fitted polynomial must achieve. A common choice is 95.0.
---	--

2	Significance level (in percent) for the lack-of-fit test that the fitted polynomial must not exceed. A common choice is 5.0.
---	--

LOF — Lack-of-fit option. (Input)

If $ICRIT = 2$, LOF must equal 1.

Default: $LOF = 0$.

LOF	Action
------------	---------------

0	DFPE and SSPE are not computed.
---	---------------------------------

1	DFPE and SSPE are computed.
---	-----------------------------

SMULTC — Multiplicative constant used to compute a scaled version of x , say z , on the interval -2 to 2 , inclusive. (Output)

SADDC — Additive constant used to compute a scaled version of $x(z)$ on the interval -2 to 2 , inclusive. (Output)

DFE — Degrees of freedom for error. (Output)

SSE — Sum of squares for error. (Output)

DFPE — Degrees of freedom for pure error. (Output, if LOF = 1)

SSPE — Sum of squares for pure error. (Output, if LOF = 1)

NRMIS — Number of rows of data encountered that contain any missing values for the independent, response, weight, or frequency variables. (Output)
NaN (not a number) is used as the missing value code. Any row of x containing NaN as a value of the independent, response, weight, or frequency variables is omitted from the fit.

FORTRAN 90 Interface

Generic: CALL RFORP (X, IRSP, IND, MAXDEG, NDEG, A, B, SCOE, D [,...])

Specific: The specific interface names are S_RFORP and D_RFORP.

FORTRAN 77 Interface

Single: CALL RFORP (NOBS, NCOL, X, LDX, IRSP, IND, IFRQ, IWT, MAXDEG, ICRT, CRIT, LOF, NDEG, SMULTC, SADDC, A, B, SCOE, D, DFE, SSE, DFPE, SSPE, NRMIS)

Double: The double precision name is DRFORP.

Example

A polynomial model is fitted to data discussed by Neter and Wasserman (1974, pages 279–285). The data set contains the response variable y measuring coffee sales (in hundred gallons) and the number of self-service coffee dispensers. Responses for fourteen similar cafeterias are in the data set, some of the cafeterias have the same number of dispensers so that lack of fit of the model can be assessed.

```
USE RFORP_INT
USE UMACH_INT
USE WRRRN_INT

INTEGER LDX, MAXDEG, NCOL, NOBS
PARAMETER (MAXDEG=2, NCOL=2, NOBS=14, LDX=NOBS)
!
INTEGER IND, IRSP, LOF, NDEG, NOUT, NRMIS
REAL A(MAXDEG), B(MAXDEG), CRIT, D(MAXDEG+1), DFE, DFPE, &
SADDC, SCOE(MAXDEG+1), SMULTC, SSE, SSPE, X(LDX,NCOL)
!
```

```

DATA (X(1,J),J=1,2) /0.0, 508.1/
DATA (X(2,J),J=1,2) /5.0, 787.6/
DATA (X(3,J),J=1,2) /0.0, 498.4/
DATA (X(4,J),J=1,2) /1.0, 568.2/
DATA (X(5,J),J=1,2) /2.0, 651.7/
DATA (X(6,J),J=1,2) /7.0, 854.7/
DATA (X(7,J),J=1,2) /2.0, 657.0/
DATA (X(8,J),J=1,2) /4.0, 755.3/
DATA (X(9,J),J=1,2) /6.0, 831.8/
DATA (X(10,J),J=1,2) /4.0, 758.9/
DATA (X(11,J),J=1,2) /5.0, 792.1/
DATA (X(12,J),J=1,2) /6.0, 841.4/
DATA (X(13,J),J=1,2) /7.0, 871.4/
DATA (X(14,J),J=1,2) /1.0, 577.3/
!
IRSP = 2
IND = 1
LOF = 1
CALL RFORP (X, IRSP, IND, MAXDEG, NDEG, A, B, SCOEf, D, LOF=LOF, &
            SMULTC=SMULTC, SADDC=SADDC, DFE=DFE, SSE=SSE, &
            DFPE=DFPE, SSPE=SSPE, NRMISS=NRMISS)
!
CALL UMACH (2, NOUT)
WRITE (NOUT,*) 'NDEG = ', NDEG
CALL WRRRN ('A', A, 1, NDEG, 1)
CALL WRRRN ('B', B, 1, NDEG, 1)
WRITE (NOUT,*) 'SMULTC = ', SMULTC
WRITE (NOUT,*) 'SADDC = ', SADDC
CALL WRRRN ('SCOEf', SCOEf, 1, NDEG+1, 1)
CALL WRRRN ('D', D, 1, NDEG+1, 1)
WRITE (NOUT,*) 'DFE = ', DFE
WRITE (NOUT,*) 'SSE = ', SSE
WRITE (NOUT,*) 'DFPE = ', DFPE
WRITE (NOUT,*) 'SSPE = ', SSPE
WRITE (NOUT,*) 'NRMISS = ', NRMISS
END

```

Output

NDEG = 2

```

      A
      1      2
0.04082 -0.07996

```

```

      B
      1      2
0.000  1.946
SMULTC =  0.571429
SADDC = -2.00000

```

```

      SCOEf
      1      2      3
711.0  90.0 -12.2

```

D

	1	2	3
	14.00	27.24	29.69
DFE =		11.0000	
SSE =		710.594	
DFPE =		7.00000	
SSPE =		304.626	
NRMISS =	0		

Comments

1. Workspace may be explicitly provided, if desired, by use of R2ORP/DR2ORP. The reference is:

```
CALL R2ORP (NOBS, NCOL, X, LDX, IRSP, IND, IFRQ, IWT, MAXDEG,
ICRIT, CRIT, LOF, NDEG, SMULTC, SADDC, A, B, SCOE, D, DFE, SSE,
DFPE, SSPE, NRMISS, WK, IWK)
```

The additional arguments are as follows:

WK — Work vector of length $8 * \text{NOBS}$.

IWK — Work vector of length NOBS .

2. Informational errors

Type Code

3	4	The response variable is constant. A zero order polynomial is fit. High order coefficients are set to zero.
3	5	There are too few observations to fit the desired degree polynomial. High order coefficients are set to zero.
3	6	A perfect fit is obtained with a polynomial of lower degree than MAXDEG.
4	1	An invalid weight is encountered.
4	2	An invalid frequency is encountered.
4	3	Each row of x contains a missing value.
4	7	The independent variable is constant. At least two distinct settings of the independent variable are needed.

3. The orthogonal polynomials evaluated at each scaled x value (z) are computed from A and B as follows:

$$\text{POLY}(I, 1) = Z(I) - A(1)$$

$$\text{POLY}(I, 2) = (Z(I) - A(2)) * \text{POLY}(I, 1) - B(2)$$

$$\text{POLY}(I, J) = (Z(I) - A(J)) * \text{POLY}(I, J - 1) - B(J) * \text{POLY}(I, J - 2)$$

for $J = 3$ through NDEG .

Description

Routine `RFORP` computes estimates of the regression coefficients in a polynomial regression model using orthogonal polynomials. The reparameterization of the polynomial model in terms of orthogonal polynomials has the advantage that the loss of accuracy resulting from forming powers of the x values is avoided. The design of `RFORP` assumes that further computations such

as summary statistics or case statistics are needed. For this reason, the results returned by `RFORP` are for the reparameterized model in terms of orthogonal polynomials. This enables computational accuracy to be maintained for the subsequent computations. Routine `RSTAP` (page 279) can be used to compute summary statistics for the original polynomial model given the results from `RFORP`. Routine `RCASP` (page 285) can be used to compute case statistics for the original polynomial model given the results from `RFORP`.

The degree of the polynomial can be specified, or the degree of the polynomial can be determined by `RFORP` under one of two criteria:

1. If some of the x values are repeated, the lowest degree polynomial can be fitted whose lack of fit is not significant at a specified level.
2. The lowest degree polynomial can be fitted with an R^2 that meets a specified lower bound.

Routine `RFORP` is based on the algorithm of Forsythe (1957). A modification to Forsythe's algorithm is made for the inclusion of weights (Kelly 1967, page 68).

Let x_i be a value of the independent variable. The x_i 's are scaled to the interval $[-2, 2]$ for computational accuracy. The scaled version of the independent variable is computed by the formula $z_i = mx_i + c$. The multiplicative scaling constant m (stored in `SMULTC`) is

$$m = \frac{4}{\max_i(x_i) - \min_i(x_i)}$$

The additive constant c (stored in `SADDC`) is

$$c = \frac{2(\min_i(x_i) + \max_i(x_i))}{\min_i(x_i) - \max_i(x_i)}$$

Orthogonal polynomials are evaluated using the three-term recurrence relationship

$$p_j(z) = (z - a_j)p_{j-1}(z) - b_j p_{j-2}(z)$$

beginning with the initial polynomials

$$p_0(z) = 1 \quad \text{and} \quad p_1(z) = z - a_1$$

The a_j 's and b_j 's (stored in `A` and `B`) are computed to make the $p_j(z)$'s orthogonal with respect to the set of weights w_i , and over the set z_i .

The fitted model is

$$\hat{y}_i = \hat{\alpha}_0 + \hat{\alpha}_1 p_1(z_i) + \cdots + \hat{\alpha}_k p_k(z_i)$$

The

$$\hat{\alpha}_j \text{'s}$$

(stored in `SCOEFF`) are computed (Shampine 1975) by

$$\hat{\alpha}_j = \frac{\sum_{i=1}^n e_i w_i p_j(z_i)}{d_j}$$

where $e_i = y_i - p_{j-1}(z_i)$ and

$$d_j = \sum_{i=1}^n w_i [p_j(z_i)]^2$$

The d_j 's (stored in **D**) can be used to compute the sum of squares due to the j -th orthogonal polynomial by

$$Q_j = d_j \hat{\alpha}_j^2$$

A more complete description of Forsythe's algorithm and the modification of Shampine appears in Kennedy and Gentle (1980, pages 342–347).

RSTAP

Computes summary statistics for a polynomial regression model given the fit based on orthogonal polynomials.

Required Arguments

- A** — Vector of length **NDEG** containing constants used to generate orthogonal polynomials. (Input)
- B** — Vector of length **NDEG** containing constants used to generate orthogonal polynomials. (Input)
- SMULTC** — Multiplicative constant used to compute the scaled version of x , say z , on the interval -2 to 2 , inclusive. (Input)
- SADDC** — Additive constant used to compute the scaled version of $x(z)$ on the interval -2 to 2 , inclusive. (Input)
- SCOEF** — Vector of length **NDEG** + 1 containing the regression coefficients of the fitted model using the scaled version of the original data. (Input)
SCOEF(1) is the estimated intercept. **SCOEF**(1 + i) contains the estimated coefficient for the i -th order orthogonal polynomial using z .
- D** — Vector of length **NDEG** + 1 containing the diagonal elements of the (diagonal) sums of squares and crossproducts matrix. (Input)
- DFE** — Degrees of freedom for error. (Input)
- SSE** — Sum of squares for error. (Input)
- COEF** — **NDEG** + 1 by 4 matrix containing statistics relating to the coefficients of the polynomial model. (Output)

Row 1 corresponds to the intercept. Row $1 + i$ corresponds to the coefficient of x^i . The columns are described as follows:

Col. Description

- 1 Estimated coefficient
- 2 Estimated standard error of estimated coefficient
- 3 t -statistic for the test that the coefficient is zero
- 4 p -value for the two-sided t test

Optional Arguments

NDEG — Degree of the polynomial regression. (Input)
Default: NDEG = size (A,1).

LOF — Lack of fit test option. (Input)
Default: LOF = 0.

LOF Action

- 0 No lack of fit test is performed.
- 1 Lack of fit test is performed.

DFPE — Degrees of freedom for pure error. (Input, if LOF = 1)
If LOF = 0, DFPE is not referenced.
Default: DFPE = 1.0.

SSPE — Sum of squares for pure error. (Input, if LOF = 1)
If LOF = 0, SSPE is not referenced.
Default: SSPE = 0.0.

IPRINT — Printing option. (Input)
Default: IPRINT = 0.

IPRINT Action

- 0 No printing is performed.
- 1 AOV, SQSS, COEF are printed.

AOV — Vector of length 15 that contains statistics relating to the analysis of variance.
(Output)

I	AOV(I)
1	Degrees of freedom for the model
2	Degrees of freedom for error
3	Total (corrected) degrees of freedom
4	Sum of squares for the model
5	Sum of squares for error
6	Total (corrected) sum of squares
7	Model mean square
8	Error mean square
9	Overall F -statistic
10	p -value
11	R^2 (in percent)
12	Adjusted R^2 (in percent)
13	Estimate of the standard deviation
14	Overall mean of y
15	Coefficient of variation (in percent)

SQSS — NDEG by 4 matrix containing sequential statistics for the polynomial model.
(Output)

Row i corresponds to x^i ($i = 1, 2, \dots, \text{NDEG}$). The columns are described as follows:

Col. Description

1	Degrees of freedom
2	Sum of squares
3	F -statistic
4	p -value

LDSQSS — Leading dimension of SQSS exactly as specified in the dimension statement of the calling program. (Input)

Default: LDSQSS = size (SQSS, 1).

LDCOEF — Leading dimension of COEF exactly as specified in the dimension statement of the calling program. (Input)

Default: LDCOEF = size (COEF,1).

TLOF — NDEG by 4 matrix containing tests of lack of fit for each degree of the polynomial. (Output, if LOF = 1)

If LOF = 0, TLOF is not referenced and can be a 1 by 1 array. Row i corresponds to x^i ($i = 1, 2, \dots, \text{NDEG}$). The columns are described as follows:

Col.	Description
------	-------------

- | | |
|---|--|
| 1 | Degrees of freedom |
| 2 | Lack of fit sum of squares |
| 3 | F test for lack of fit of the polynomial model of degree i |
| 4 | p -value for the F test |

LDTLOF — Leading dimension of TLOF exactly as specified in the dimension statement of the calling program. (Input)

Default: LDTLOF = size (TLOF, 1).

FORTRAN 90 Interface

Generic: CALL RSTAP (A, B, SMULTC, SADDC, SCOEF, D, DFE, SSE,
COEF [,...])

Specific: The specific interface names are S_RSTAP and D_RSTAP.

FORTRAN 77 Interface

Single: CALL RSTAP (NDEG, A, B, SMULTC, SADDC, SCOEF, D, DFE, SSE,
LOF, DFPE, SSPE, IPRINT, AOV, SQSS, LDSQSS, COEF, LDCOEF,
TLOF, LDTLOF)

Double: The double precision name is DRSTAP.

Example

A polynomial model is fitted to data discussed by Neter and Wasserman (1974, pages 279–285). The data set contains the response variable y measuring coffee sales (in hundred gallons) and the number of self-service coffee dispensers. Responses for fourteen similar cafeterias are in the

data set and some of the cafeterias have the same number of dispensers so that lack of fit of the model can be assessed.

```

USE RSTAP_INT
USE RFORP_INT
INTEGER   LDcoef, LDSQSS, LDTLOF, LDX, MAXDEG, NCOL, NOBS
PARAMETER (MAXDEG=2, NCOL=2, NOBS=14, LDcoef=MAXDEG+1, &
           LDSQSS=MAXDEG, LDTLOF=MAXDEG, LDX=NOBS)
!
INTEGER   IND, IPRINT, IRSP, LOF, NDEG, NRMIS
REAL     A(MAXDEG), AOV(15), B(MAXDEG), COEF(MAXDEG+1,4), &
         CRIT, D(MAXDEG+1), DFE, DFPE, SADDC, SCOEf(MAXDEG+1), &
         SMULTC, SQSS(LDSQSS,4), SSE, SSPE, TLOF(MAXDEG,4), &
         X(LDX,NCOL)
!
DATA (X(1,J),J=1,2) /0.0, 508.1/
DATA (X(2,J),J=1,2) /5.0, 787.6/
DATA (X(3,J),J=1,2) /0.0, 498.4/
DATA (X(4,J),J=1,2) /1.0, 568.2/
DATA (X(5,J),J=1,2) /2.0, 651.7/
DATA (X(6,J),J=1,2) /7.0, 854.7/
DATA (X(7,J),J=1,2) /2.0, 657.0/
DATA (X(8,J),J=1,2) /4.0, 755.3/
DATA (X(9,J),J=1,2) /6.0, 831.8/
DATA (X(10,J),J=1,2) /4.0, 758.9/
DATA (X(11,J),J=1,2) /5.0, 792.1/
DATA (X(12,J),J=1,2) /6.0, 841.4/
DATA (X(13,J),J=1,2) /7.0, 871.4/
DATA (X(14,J),J=1,2) /1.0, 577.3/
!
IRSP = 2
IND = 1
LOF = 1
CALL RFORP (X, IRSP, IND, MAXDEG, NDEG, A, B, SCOEf, D, LOF=LOF, &
           SMULTC=SMULTC, SADDC=SADDC, DFE=DFE, SSE=SSE, &
           DFPE=DFPE, SSPE=SSPE)
!
IPRINT = 1
CALL RSTAP (A, B, SMULTC, SADDC, SCOEf, D, DFE, SSE, COEF, &
           NDEG=NDEG, LOF=LOF, DFPE=DFPE, SSPE=SSPE, IPRINT=IPRINT, &
           AOV=AOV, SQSS=SQSS, TLOF=TLOF)
END

```

Output

R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
99.685	99.628	8.037	711.0	1.13

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Regression	2	225031.9	112515.9	1741.748	0.0000
Residual	11	710.6	64.6		
Corrected Total	13	225742.5			

```

* * * Inference on Coefficients * * *
Coef.      Estimate      Standard      Prob. of
           Error t-statistic Larger |t|
1           503.3         4.791        105.054    0.0000
2            78.9         3.453         22.865    0.0000
3            -4.0         0.482         -8.242    0.0000

* * * Sequential Statistics * * *
Degree of  Degrees of      Sum of      Prob. of
Polynomial Freedom    Squares F-statistic Larger F
1           1        220644.1    3415.574    0.0000
2           1         4387.7      67.922     0.0000

* * * Tests of Lack of Fit * * *
Degree of  Degrees of      Sum of      Prob. of
Polynomial Freedom    Squares F-statistic Larger F
1           5         4793.7     22.031     0.0004
2           4         406.0      2.332     0.1547

```

Comments

Workspace may be explicitly provided, if desired, by use of R2TAP/DR2TAP.
The reference is:

CALL R2TAP (NDEG, A, B, SMULTC, SADDC, SCOE, D, DFE, SSE, LOF, DFPE,
SSPE, IPRINT, AOV, SQSS, LDSQSS, COEF, LDCOE, TLOF, LDTLOF, WK)

The additional argument is:

WK — Work vector of length $(NDEG + 1) * (NDEG + 7)$.

Description

Routine RSTAP transforms a polynomial regression model, fitted using orthogonal polynomials, into a polynomial function of the original independent variable. In addition, summary statistics (analysis of variance, t tests, tests for lack of fit) are computed. Results from routine RFORP (see 272), which produces the fit using orthogonal polynomials, are used for input.

The fitted model from RFORP is

$$\hat{y}_i = \hat{\alpha}_0 p_0(z_i) + \hat{\alpha}_1 p_1(z_i) + \cdots + \hat{\alpha}_k p_k(z_i)$$

where the z_i 's are the settings of the independent variable x scaled to the interval $[-2, 2]$ and

where the $p_j(z)$'s are the orthogonal polynomials. The " $X^T X$ " matrix for this model is a diagonal matrix with elements d_j (stored in D). The orthogonal polynomials can be expressed as

$$p_j(z) = \sum_{m=0}^j \delta_{jm} z^m$$

First, RSTAP computes

$$\hat{\gamma}_j = \sum_{m=j}^k \hat{\alpha}_m \delta_{mj}$$

to produce the fit for the polynomial function in terms of the scaled independent variable as given by

$$\hat{y}_i = \hat{\gamma}_0 + \hat{\gamma}_1 z_i + \cdots + \hat{\gamma}_k z_i^k$$

The variances and covariances for the estimated coefficients in this model are given by

$$\text{cov}(\hat{\gamma}_i, \hat{\gamma}_j) = \sigma^2 \sum_{m=\max(i,j)}^k \delta_m \delta_{jm} d_j^{-1} \quad i, j = 0, 1, \dots, m$$

Second, RSTAP computes

$$\hat{\beta}_j$$

as a linear combination of the

$$\hat{\gamma}_j \text{'s}$$

by the formula

$$\hat{\beta}_j = \sum_{m=0}^{k-j} (-1)^m 2^{2j+m} \binom{j+m}{m} \left(\frac{\min_i x_i + \max_i x_i}{\max_i x_i - \min_i x_i} \right)^m \left(\frac{1}{\max_i x_i - \min_i x_i} \right)^{j+m} \hat{\gamma}_{j+m}$$

in order to produce the fit for the polynomial function in terms of the original independent variable as given by

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + \cdots + \hat{\beta}_k x_i^k$$

The variance of

$$\hat{\beta}_j$$

computed from the variances and covariances of the

$$\hat{\gamma}_j \text{'s}$$

using the usual formula for computing variances of linear combinations of correlated random variables. The sequential sum of squares due to x^j (stored in SQSS) is computed by

$$Q_j = d_j \hat{\alpha}_j^2$$

RCASP

Computes case statistics for a polynomial regression model given the fit based on orthogonal polynomials.

Required Arguments

X — NOBS by NCOL matrix containing the data. (Input)

IRSP — Column number `IRSP` of `X` contains the data for the response (dependent) variable. (Input)

IND — Column number `IND` of `X` contains the data for the independent (explanatory) variable. (Input)

NDEG — Degree of the polynomial regression. (Input)

SMULTC — Multiplicative constant used to compute a scaled version of x on the interval -2 to 2 , inclusive. (Input)

SADDC — Additive constant used to compute a scaled version of x on the interval -2 to 2 , inclusive. (Input)

A — Vector of length `NDEG` containing constants used to generate orthogonal polynomials. (Input)

B — Vector of length `NDEG` containing constants used to generate orthogonal polynomials. (Input)

SCOEF — Vector of length `NDEG + 1` containing the regression coefficients

$$\hat{\alpha}$$

of the fitted model using the scaled version of $x(z)$. (Input)

$$\hat{\alpha}_0 = \text{SCOEF}(1)$$

is the estimated intercept and equals the response mean.

$$\hat{\alpha}_i = \text{SCOEF}(1+i)$$

contains the estimated coefficient for the i -th order orthogonal polynomial using the scaled version of $x(z)$.

D — Vector of length `NDEG + 1` containing the diagonal elements of the (diagonal) sums of squares and crossproducts matrix. (Input)

SSE — Sum of squares for error. (Input)

DFE — Degrees of freedom for error. (Input)

CASE — `NOBS` by 12 matrix containing the case statistics. (Output)
Columns 1 through 12 contain the following:

Col. Description

- 1 Observed response
- 2 Predicted response
- 3 Residual
- 4 Leverage
- 5 Standardized residual
- 6 Jackknife residual
- 7 Cook's distance
- 8 DFFITS
- 9, 10 Confidence interval on the mean
- 11, 12 Prediction interval

Optional Arguments

NOBS — Number of observations. (Input)
Default: NOBS = size (x,1).

NCOL — Number of columns in x. (Input)
Default: NCOL = size (x,2).

LDX — Leading dimension of x exactly as specified in the dimension statement in the calling program. (Input)
Default: LDX = size (x,1).

IWT — Weighting option. (Input)
IWT = 0 means that all weights are 1.0. For positive IWT, column number IWT of x contains the weights, and the computed prediction interval uses $SSE/(DFE * x(i, IWT))$ for the estimated variance of a future response.
Default: IWT = 0.

IPRED — Prediction interval option. (Input)
IPRED = 0 means that prediction intervals are desired for a single future response. For positive IPRED, column number IPRED of x contains the number of future responses for which a prediction interval is desired on the average of the future responses.
Default: IPRED = 0.

CONPCM — Confidence level for two-sided interval estimates on the mean, in percent.

(Input)

Default: CONPCM = 95.0.

CONPCP — Confidence level for two-sided prediction intervals, in percent. (Input)

Default: CONPCP = 95.0.

PRINT — Printing option. (Input)

Default: PRINT = 'N'.

PRINT is a character string indicating what is to be printed. The PRINT string is composed of one-character print codes to control printing. These print codes are given as follows:

PRINT(I:I)	Printing that Occurs
'A'	All
'N'	None
'1'	Observed response
'2'	Predicted response
'3'	Residual
'4'	Leverage
'5'	Standardized residual
'6'	Jackknife residual
'7'	Cook's distance
'8'	DFFITs
'M'	Confidence interval on the mean
'P'	Prediction interval
'X'	Influential cases (unusual "x-value")
'Y'	Outlier cases (unusual "y-value")

The concatenated print codes 'A', 'N', '1', ..., 'P' that comprise the PRINT string give the combination of statistics to be printed. Concatenation of these codes with print codes 'X' or 'Y' restricts printing to cases determined to be influential or outliers. Here are a few examples:

PRINT Printing Action

- 'A' All.
- 'N' None.
- '46' Leverage and jackknife residual for all cases.
- 'AXY' All statistics are printed for cases that are highly influential or are outliers.
- '46XY' Leverage and jackknife residual are printed for cases that are highly influential or are outliers.

LDCASE — Leading dimension of **CASE** exactly as specified in the dimension statement in the calling program. (Input)
Default: **LDCASE** = size (**CASE**,1).

NRMISS — Number of rows of **CASE** containing NaN (not a number). (Output)

FORTRAN 90 Interface

Generic: `CALL RCASP (X, IRSP, IND, NDEG, SMULTC, SADDC, A, B, SCOEF, D, SSE, DFE, CASE [,...])`

Specific: The specific interface names are **S_RCASP** and **D_RCASP**.

FORTRAN 77 Interface

Single: `CALL RCASP (NOBS, NCOL, X, LDX, IRSP, IND, IWT, IPRED, CONPCM, CONPCP, NDEG, SMULTC, SADDC, A, B, SCOEF, D, SSE, DFE, PRINT, CASE, LDCASE, NRMISS)`

Double: The double precision name is **DRCASP**.

Example

A polynomial model is fitted to data discussed by Neter and Wasserman (1974, pages 279–285). The data set contains the response variable *y* measuring coffee sales (in hundred gallons) and the number of self-service coffee dispensers. Responses for fourteen similar cafeterias are in the data set.

```
USE RCASP_INT
USE RFORP_INT

INTEGER    LDCASE, LDcoef, LDSQSS, LDTLOF, LDX, MAXDEG, NCOL, &
           NOBS

PARAMETER (MAXDEG=2, NCOL=2, NOBS=14, LDCASE=NOBS, &
           LDcoef=MAXDEG+1, LDSQSS=MAXDEG, LDTLOF=MAXDEG, &
           LDX=NOBS)
```

!

```

INTEGER      ICRIT, IFRQ, IND, IPRED, IRSP, IWT, LOF, NDEG, NRMIS
REAL        A(MAXDEG), B(MAXDEG), CASE(LDCASE,12), CONPCM, &
CONPCP, CRIT, D(MAXDEG+1), DFE, DFPE, SADDC, &
SCOEF(MAXDEG+1), SMULTC, SSE, SSPE, X(LDX,NCOL)
CHARACTER   PRINT*1
!
DATA (X(1,J),J=1,2) /0.0, 508.1/
DATA (X(2,J),J=1,2) /5.0, 787.6/
DATA (X(3,J),J=1,2) /0.0, 498.4/
DATA (X(4,J),J=1,2) /1.0, 568.2/
DATA (X(5,J),J=1,2) /2.0, 651.7/
DATA (X(6,J),J=1,2) /7.0, 854.7/
DATA (X(7,J),J=1,2) /2.0, 657.0/
DATA (X(8,J),J=1,2) /4.0, 755.3/
DATA (X(9,J),J=1,2) /6.0, 831.8/
DATA (X(10,J),J=1,2) /4.0, 758.9/
DATA (X(11,J),J=1,2) /5.0, 792.1/
DATA (X(12,J),J=1,2) /6.0, 841.4/
DATA (X(13,J),J=1,2) /7.0, 871.4/
DATA (X(14,J),J=1,2) /1.0, 577.3/
!
IRSP = 2
IND = 1
LOF = 1
CALL RFORP (X, IRSP, IND, MAXDEG, NDEG, A, B, SCOEF, D, LOF=LOF, &
SMULTC=SMULTC, SADDC=SADDC, DFE=DFE, SSE=SSE)
!
PRINT = 'A'
CALL RCASP (X, IRSP, IND, NDEG, SMULTC, SADDC, A, B, SCOEF, D, SSE, &
DFE, CASE, PRINT=PRINT)
!
END

```

Output

```

* * * Case Analysis * * *

```

Obs.	Observed	Predicted	Residual	Leverage	Std. Res.	Jack. Res
	Cook's D	DFFITS	95.0% CI	95.0% CI	95.0% PI	95.0% PI
1	508.1000	503.3459	4.7541	0.3554	0.7367	0.7204
	0.0997	0.5349	492.8003	513.8916	482.7510	523.9409
2	787.6000	798.8150	-11.2150	0.1429	-1.5072	-1.6132
	0.1262	-0.6586	792.1288	805.5012	779.9034	817.7266
3	498.4000	503.3459	-4.9460	0.3554	-0.7664	-0.7511
	0.1079	-0.5577	492.8003	513.8916	482.7510	523.9409
4	568.2000	578.3177	-10.1177	0.1507	-1.3660	-1.4293
	0.1104	-0.6021	571.4498	585.1857	559.3412	597.2943
5	651.7000	645.3505	6.3495	0.1535	0.8586	0.8476
	0.0446	0.3609	638.4200	652.2810	626.3513	664.3498
6	854.7000	861.4297	-6.7297	0.3650	-1.0508	-1.0563
	0.2116	-0.8008	850.7420	872.1175	840.7617	882.0978
7	657.0000	645.3505	11.6495	0.1535	1.5753	1.7069
	0.1500	0.7268	638.4200	652.2810	626.3513	664.3498
8	755.3000	755.5992	-0.2992	0.1897	-0.0414	-0.0394
	0.0001	-0.0191	747.8945	763.3038	736.3040	774.8943
9	831.8000	834.0919	-2.2919	0.1429	-0.3080	-0.2949
	0.0053	-0.1204	827.4056	840.7782	815.1804	853.0035

10	758.9000	755.5992	3.3008	0.1897	0.4562	0.4392
	0.0162	0.2125	747.8945	763.3038	736.3040	774.8943
11	792.1000	798.8150	-6.7150	0.1429	-0.9024	-0.8942
	0.0452	-0.3650	792.1288	805.5012	779.9034	817.7266
12	841.4000	834.0919	7.3081	0.1429	0.9821	0.9804
	0.0536	0.4002	827.4056	840.7782	815.1804	853.0035
13	871.4000	861.4297	9.9703	0.3650	1.5567	1.6809
	0.4643	1.2745	850.7420	872.1175	840.7617	882.0978
14	577.3000	578.3177	-1.0178	0.1507	-0.1374	-0.1311
	0.0011	-0.0552	571.4498	585.1857	559.3412	597.2943

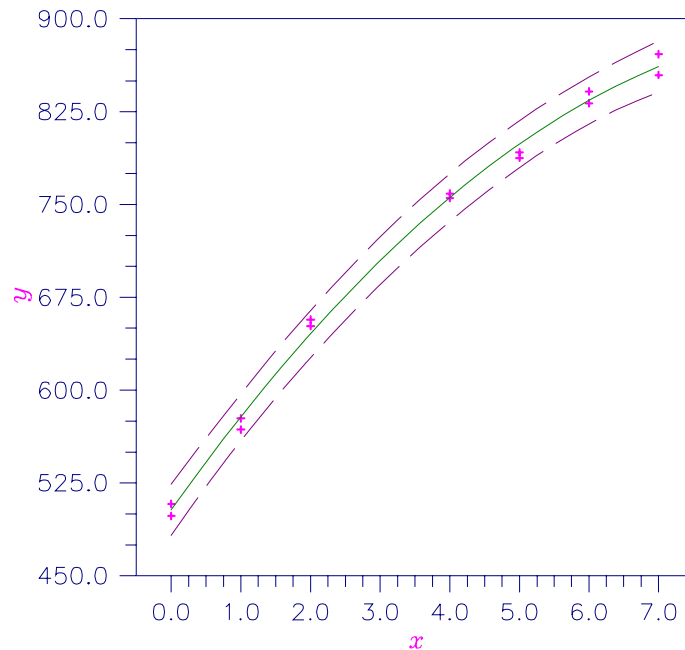


Figure 2-9 Second Degree Polynomial Fit With 95% One-at-a-Time Prediction Intervals

Comments

1. Workspace may be explicitly provided, if desired, by use of R2ASP/DR2ASP. The reference is:

```
CALL R2ASP (NOBS, NCOL, X, LDX, IRSP, IND, IWT, IPRED, CONPCM,
CONPCP, NDEG, SMULTC, SADD, A, B, SCOE, D, SSE, DFE, PRINT,
CASE, LDCASE, NRMISS, WK)
```

The additional argument is:

WK — Work vector of length NDEG + 1.

2. Informational errors
Type Code

4	1	A weight is negative. Weights must be nonnegative
4	8	The number of future observations for a prediction interval must be positive.
3	9	A leverage much greater than one is computed. It is set to one.
3	10	A deleted residual mean square much less than zero is computed. It is set to 0.0.

Description

Routine `RCASP` assumes a polynomial model

$$y_i = \beta_0 + \beta_1 x_i + \cdots + \beta_k x_i^k + \varepsilon_i \quad i = 1, 2, \dots, n$$

where the observed values of the y_i 's constitute the response, the x_i 's are the settings of the independent variable, the β_j 's are the regression coefficients and the ε_i 's are the errors that are independently distributed normal with mean 0 and variance σ^2/w_i . Given the results of a polynomial regression, fitted using orthogonal polynomials and weights w_i , routine `RCASP` produces predicted values, residuals, confidence intervals, prediction intervals, and diagnostics for outliers and influential cases.

Often a predicted value and confidence interval are desired for a setting of the independent variable not used in computing the regression fit. This can be accomplished by including the independent variable setting as part of the data matrix and by setting the response equal to NaN (not a number). NaN can be retrieved by `AMACH(6)`.

Results from routine `RFORP` ([page 272](#)), which produces the fit using orthogonal polynomials, are used for input. The fitted model from `RFORP` is

$$\hat{y}_i = \hat{\alpha}_0 p_0(z_i) + \hat{\alpha}_1 p_1(z_i) + \cdots + \hat{\alpha}_k p_k(z_i)$$

where the z_i 's are settings of the independent variable x scaled to the interval $[-2, 2]$ and where the $p_j(z)$'s are the orthogonal polynomials. The " $X^T X$ " matrix for this model is a diagonal matrix with elements d_j (stored in `D`). The case statistics are easily computed from this model and are equal to those from the original polynomial model with the β_j 's as the regression coefficients.

The leverage is computed as

$$h_i = w_i \sum_{j=0}^k d_j^{-1} p_j^2(z_i)$$

The estimated variance of

$$\hat{y}_i$$

is given by $h_i \sigma^2 / w_i$. The computation of the remainder of the case statistics follows easily from their definitions. See the chapter introduction ([page 83](#)) for definitions of the case diagnostics.

OPOLY

Generates orthogonal polynomials with respect to x -values and specified weights.

Required Arguments

- X* — Vector of length *N* containing the *x*-values. (Input)
- NDEG* — Degree of highest degree orthogonal polynomial to be generated. (Input)
- SMULTC* — Multiplicative constant used to compute a scaled version of *x* on the interval -2 to 2 , inclusive. (Output)
- SADDC* — Additive constant used to compute a scaled version of *x* on the interval -2 to 2 , inclusive. (Output)
- SX* — Vector of length *N* containing the scaled version of *x* on the interval -2 to 2 , inclusive, computed as follows: $SX(i) = SMULTC * X(i) + SADDC$ where $i = 1, 2, \dots, N$. (Output)
If *X* is not needed, *SX* and *X* can occupy the same storage locations.
- A* — Vector of length *NDEG* containing constants used to generate orthogonal polynomials. (Output)
- B* — Vector of length *NDEG* containing constants used to generate orthogonal polynomials. (Output)
- POLY* — Matrix, *N* by *NDEG*, containing the orthogonal polynomials evaluated at *SX(i)* for $i = 1, 2, \dots, N$. (Output)

Optional Arguments

- N* — Number of *x*-values. (Input)
Default: $N = \text{size}(\text{POLY}, 1)$.
- IWT* — Weighting option. (Input)
 $IWT = 0$ means that all weights are 1.0. For $IWT = 1$, *WT* contains the weights.
Default: $IWT = 0$.
- WT* — Vector of length *N* containing the weights. (Input, if $IWT = 1$)
If $IWT = 0$, *WT* is not referenced and can be a vector of length one.
- LDAPOLY* — Leading dimension of *POLY* exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDAPOLY = \text{size}(\text{POLY}, 1)$.

FORTRAN 90 Interface

- Generic: CALL OPOLY (X, NDEG, SMULTC, SADDC, SX, A, B, POLY [, ...])
- Specific: The specific interface names are S_OPOLY and D_OPOLY.

FORTRAN 77 Interface

Single: CALL OPOLY (N, X, IWT, WT, NDEG, SMULTC, SADDC, SX, A, B,
 POLY, LDPOLY)

Double: The double precision name is DOPOLY.

Example

First-degree and second-degree orthogonal polynomials are generated using equally spaced x values 1, 2, ..., 12. (Equally spaced x values are not required by OPOLY.)

```
USE OPOLY_INT
USE UMACH_INT
USE WRRRN_INT

INTEGER      LDPOLY, N, NDEG
PARAMETER   (N=12, NDEG=2, LDPOLY=N)
!
INTEGER      NOUT
REAL         A(NDEG), B(NDEG), POLY(LDPOLY,NDEG), SADDC, SMULTC, &
            SX(N), WT(1), X(N)
!
DATA X/1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, &
    12.0/
!
CALL OPOLY (X, NDEG, SMULTC, SADDC, SX, A, B, POLY)
CALL UMACH (2, NOUT)
WRITE (NOUT,99999) SMULTC, SADDC
99999 FORMAT (' SMULTC = ', F7.3, ' SADDC = ', F7.3)
CALL WRRRN ('A', A, 1, NDEG, 1)
CALL WRRRN ('B', B, 1, NDEG, 1)
CALL WRRRN ('POLY', POLY)
END
```

Output

SMULTC = 0.364 SADDC = -2.364

 A
 1 2
-5.960E-08 -1.009E-07

 B
 1 2
0.000 1.576

 POLY
 1 2
1 -2.000 2.424
2 -1.636 1.102
3 -1.273 0.044
4 -0.909 -0.749
5 -0.545 -1.278
6 -0.182 -1.543
7 0.182 -1.543

8	0.545	-1.278
9	0.909	-0.749
10	1.273	0.044
11	1.636	1.102
12	2.000	2.424

Comments

1. Informational error
Type Code
3 8 N must be greater than NDEG in order for higher order polynomials to be nonzero. Columns N + 1 through NDEG of POLY are set to zero.
2. The orthogonal polynomials evaluated at each scaled X value are computed from A and B as follows:
POLY(I, 1) = SX(I) - A(1)
POLY(I, 2) = (SX(I) - A(2)) * POLY(I, 1) - B(2)
POLY(I, J) = (SX(I) - A(J)) * POLY(I, J - 1) - B(J) * POLY(I, J - 2) for J = 3 through NDEG.
3. If NDEG is greater than 10, the accuracy of the results may be questionable.

Description

Routine OPOLY generates orthogonal polynomials over a set of x_i 's and with respect to weights w_i . The routine OPOLY is based on the algorithm of Forsythe (1957). (See also Kennedy and Gentle 1980.) A modification to Forsythe's algorithm is made for the inclusion of weights (Kelly 1967, page 68).

Let x_i be a value of the independent variable. The x_i 's are scaled to the interval $[-2, 2]$ for computational accuracy. The scaled version of the independent variable is computed by the formula $z_i = mx_i + c$. The multiplicative scaling constant m (stored in SMULTC) is

$$m = \frac{4}{\max_i(x_i) - \min_i(x_i)}$$

The additive constant c (stored in SADDC) is

$$c = \frac{2(\min_i(x_i) + \max_i(x_i))}{\min_i(x_i) - \max_i(x_i)}$$

Orthogonal polynomials are generated using the three-term recurrence relationship

$$p_j(z) = (z - a_j)p_{j-1}(z) - b_j p_{j-2}(z)$$

beginning with the initial polynomials

$$p_0(z) = 1 \quad \text{and} \quad p_1(z) = z - a_1$$

The a_j 's and b_j 's (stored in \mathbf{A} and \mathbf{B}) are computed to make the $p_j(z)$'s orthogonal, with respect to the the set of weights w_i , and over the set z_i .

GCSCP

Generates centered variables, squares, and crossproducts.

Required Arguments

X — NRX by NVAR matrix containing the data. (Input)

XMEAN — Vector of length NVAR containing the means of the variables. (Input)

CSCP — NRX by $\text{NVAR} * (\text{NVAR} + 3)/2$ matrix containing the centered variables, squares, and crossproducts. (Output)

Columns	Description
1 to NVAR	Centered variables
$\text{NVAR} + 1$ to $2 * \text{NVAR}$	Squared variables
$2 * \text{NVAR} + 1$ to $\text{NVAR} * (\text{NVAR} + 3)/2$	Crossproducts

If \mathbf{X} is not needed, \mathbf{X} and the first NVAR columns of \mathbf{CSCP} may occupy the same storage locations.

Optional Arguments

IDO — Processing option. (Input)
Default: $\text{IDO} = 0$.

IDO	Action
------------	---------------

0	This is the only invocation of GCSCP for this data set, and all the data are input at once.
---	--

1	This is the first invocation, and additional calls to GCSCP will be made. Initialization and updating for the data in \mathbf{X} are performed.
---	--

2	This is an intermediate or final invocation of GCSCP and updating for the data in \mathbf{X} is performed.
---	---

NRX — Number of rows of data in \mathbf{X} . (Input)
Default: $\text{NRX} = \text{size}(\mathbf{X}, 1)$.

NVAR — Number of variables. (Input)
Default: $NVAR = \text{size}(x,2)$.

LDX — Leading dimension of x exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDX = \text{size}(x,1)$.

ICEN — Centering option. (Input)
If $IDO = 1$ or $IDO = 2$, **ICEN** must equal 0.
Default: $ICEN = 0$.

ISUB Action

- 0 CSCP contains the centered variables in columns 1 through $NVAR$. Square and crossproduct variables are generated from these centered variables in the remaining columns of CSCP.
- 1 First, the action taken when $ICEN = 0$ is performed. Next, the means of the square and crossproduct variables are subtracted from the square and crossproduct variables.

SCPM — Vector of length $NVAR * (NVAR + 1)/2$ containing the means of the generated square and crossproduct variables. (Output, if $IDO = 0$ or 1; input/output, if $IDO = 2$)

Elements	Description
1 to $NVAR$	Squared variable means
$NVAR+1$ to $NVAR * (NVAR + 1)/2$	Crossproduct variable means

LDCSCP — Leading dimension of CSCP exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDCSCP = \text{size}(CSCP,1)$.

NRMIS — Number of rows of data encountered in calls to GCSCP that contain any missing values for the variables. (Output, if $IDO = 0$ or 1; input/output, if $IDO = 2$)
NaN (not a number) is used as the missing value code.
Default: $NRMIS = 0$.

NVOBS — Number of valid observations. (Output, if $IDO = 0$ or 1; input/output, if $IDO = 2$)
Number of rows of data encountered in calls to GCSCP that do not contain any missing values for the variables.

FORTRAN 90 Interface

Generic: `CALL GCSCP (X, XMEAN, CSCP [, ...])`

Specific: The specific interface names are S_GCSCP and D_GCSCP.

FORTRAN 77 Interface

Single: CALL GCSCP (IDO, NRX, NVAR, X, LDX, ICEN, XMEAN, SCPM, CSCP, LDCSCP, NRMISS, NVOBS)

Double: The double precision name is DGCSCP.

Example 1

With data containing 4 rows and 3 variables, GCSCP is used to center the variables and to generate (using the centered variables) the square and crossproduct variables. The data is input in one invocation (IDO = 0), and the generated squared and crossproduct variables are centered (ICEN = 1). On output, SCPM contains the means in standard order, i.e.,

$$\overline{z_1^2}, \overline{z_2^2}, \overline{z_3^2}, \overline{z_1 z_2}, \overline{z_1 z_3}, \overline{z_2 z_3}$$

Also, CSCP contains the variables in standard order, i.e.,

$$z_1, z_2, z_3, z_1^2 - \overline{z_1^2}, z_2^2 - \overline{z_2^2}, z_3^2 - \overline{z_3^2}, z_1 z_2 - \overline{z_1 z_2}, z_1 z_3 - \overline{z_1 z_3}, z_2 z_3 - \overline{z_2 z_3}$$

```

USE GCSCP_INT
USE UMACH_INT
USE WRRRN_INT

INTEGER    LDCSCP, LDX, NRX, NVAR
PARAMETER (NRX=4, NVAR=3, LDCSCP=NRX, LDX=NRX)
!
INTEGER    NOUT, NRMISS, NVOBS
REAL      CSCP(LDCSCP, NVAR*(NVAR+3)/2), SCPM(NVAR*(NVAR+1)/2), &
          X(LDX, NVAR), XMEAN(NVAR)
!
DATA (X(1,J), J=1, NVAR) / 10.0, 8.0, 11.0 /
DATA (X(2,J), J=1, NVAR) / 5.0, 15.0, 1.0 /
DATA (X(3,J), J=1, NVAR) / 3.0, 2.0, 4.0 /
DATA (X(4,J), J=1, NVAR) / 6.0, 3.0, 4.0 /
DATA XMEAN / 6.0, 7.0, 5.0 /
!
ICEN = 1
CALL GCSCP (X, XMEAN, CSCP, ICEN=ICEN, SCPM=SCPM, NRMISS=NRMISS, &
           NVOBS=NVOBS)
!
CALL UMACH (2, NOUT)
WRITE (NOUT, *) ' NRMISS = ', NRMISS
CALL WRRRN ('SCPM', SCPM, 1, NVAR*(NVAR+1)/2, 1)
CALL WRRRN ('CSCP', CSCP)
END

```

Output

NRMISS = 0

SCPM

	1	2	3	4	5	
6.50	26.50	13.50	2.75	7.75	-4.25	

					CSCP				
	1	2	3	4	5	6	7	8	9
1	4.00	1.00	6.00	9.50	-25.50	22.50	1.25	16.25	10.25
2	-1.00	8.00	-4.00	-5.50	37.50	2.50	-10.75	-3.75	-27.75
3	-3.00	-5.00	-1.00	2.50	-1.50	-12.50	12.25	-4.75	9.25
4	0.00	-4.00	-1.00	-6.50	-10.50	-12.50	-2.75	-7.75	8.25

Comments

Crossproduct variables are ordered as follows: (1, 2), (1, 3), ..., (1, NVAR), (2, 3), (2, 4), ..., (2, NVAR), ..., (NVAR - 1, NVAR).

Programming Notes

Routine `GCSCP` centers a data set consisting of independent variable settings and generates (using the centered variables) the settings for all possible squared and crossproduct variables in standard order. The routine `GCSCP` is designed so that you can partition a large data set into submatrices (requiring less space) and make multiple calls to `GCSCP` (with `IDO = 1, 2, 2, ..., 2`). Alternatively, one invocation of `GCSCP` (with `IDO = 0`) can be made with the entire data set contained in `X`.

Let n be the number of rows in the entire data set, and let m (stored in `NVAR`) be the number of variables. Let x_{ij} be the i -th setting of the j -th variable ($i = 1, 2, \dots, n; j = 1, 2, \dots, m$). Denote the means (stored in `XMEAN`) by

$$\bar{x}_j \quad (j = 1, 2, \dots, m)$$

The settings of the j -th centered variable (stored in the j -th column of `CSCP`) are given by

$$z_{ij} = x_{ij} - \bar{x}_j$$

The settings of the j -th squared variable (stored in the $(m + j)$ -th column of `CSCP`) are given by

$$\begin{cases} z_{ij}^2 & \text{if ICEN} = 0 \\ z_{ij}^2 - \bar{z}_j^2 & \text{if ICEN} = 1 \end{cases}$$

where

$$\bar{z}_j^2 = \sum_{i=1}^n \frac{z_{ij}^2}{n}$$

(stored in the $(m + j)$ -th column of `SCPM`) is the mean of the j -th squared variable. The settings of the jk crossproduct variable (stored in the

$$k - j + mj - \frac{j(j-1)}{2}$$

column of CSCP) are given by

$$\begin{cases} \overline{z_{ij}z_{ik}} & \text{if ICEN} = 0 \\ \overline{z_{ij}z_{ik}} - \overline{z_j}z_k & \text{if ICEN} = 1 \end{cases}$$

where

$$\overline{z_j z_k} = \frac{\sum_{i=1}^n z_{ij} z_{ik}}{n}$$

(stored in the

$$k - j + mj - \frac{j(j-1)}{2}$$

location of SCFM) is the mean of the jk -th ($j < k$) crossproduct variable.

Additional Example

Example 2

With data containing 4 rows and 3 variables, GCSCP is used to center the variables and to generate (using the centered variables) the square and crossproduct variables. The data is input in multiple invocations (IDO = 1, 2, 2, 2). Here, the square and crossproduct variables, generated using the centered variables, cannot be centered (ICEN = 0).

```

USE GCSCP_INT
USE UMACH_INT
USE WRRRN_INT

INTEGER    LDCSCP, LDX, NRX, NVAR
PARAMETER (LDX=4, NRX=1, NVAR=3, LDCSCP=NRX)
!

INTEGER    I, IDO, MISS, NOUT, NRMISS, NVOBS
REAL       CSCP(LDCSCP,NVAR*(NVAR+3)/2), SCPM(NVAR*(NVAR+1)/2), &
           X(LDX,NVAR), XMEAN(NVAR)
!

DATA (X(1,J),J=1,NVAR)/10.0, 8.0, 11.0/
DATA (X(2,J),J=1,NVAR)/ 5.0, 15.0, 1.0/
DATA (X(3,J),J=1,NVAR)/ 3.0, 2.0, 4.0/
DATA (X(4,J),J=1,NVAR)/ 6.0, 3.0, 4.0/
DATA XMEAN/6.0, 7.0, 5.0/
!

CALL UMACH (2, NOUT)
MISS = 0
DO 10 I=1, 4
  IF (I .EQ. 1) THEN
    IDO = 1

```

```

ELSE
  IDO = 2
END IF
CALL GCSCP (X(I:,1:), XMEAN, CSCP, IDO=IDO, NRX=NRX, SCPM=SCPM, &
           NRMISS=NRMISS, NVOBS=NVOBS)
MISS = MISS + NRMISS
CALL WRRRN ('CSCP', CSCP)
10 CONTINUE
CALL WRRRN ('SCPM', SCPM, 1, NVAR*(NVAR+1)/2, 1)
WRITE (NOUT,*) ' MISS = ', MISS
END

```

Output

```

              CSCP
  1      2      3      4      5      6      7      8      9
4.00    1.00    6.00   16.00   1.00   36.00   4.00   24.00   6.00

              CSCP
  1      2      3      4      5      6      7      8      9
-1.00   8.00  -4.00   1.00   64.00   16.00  -8.00   4.00  -32.00

              CSCP
  1      2      3      4      5      6      7      8      9
-3.00  -5.00  -1.00   9.00   25.00   1.00   15.00   3.00   5.00

              CSCP
  1      2      3      4      5      6      7      8      9
 0.00  -4.00  -1.00   0.00   16.00   1.00   0.00   0.00   4.00

              SCPM
  1      2      3      4      5      6
 6.50  26.50  13.50   2.75   7.75  -4.25
MISS =    0

```

TCSCP

Transforms coefficients from a second order response surface model generated from squares and crossproducts of centered variables to a model using uncentered variables.

Required Arguments

XMEAN — Vector of length *NVAR* containing the means of the variables. (Input)

SCPM — Vector of length $NVAR(NVAR + 1)/2$ containing the means of the generated square and crossproduct variables. (Input)

Elements

1 to *NVAR*

NVAR+ 1 to *NVAR* * (*NVAR* + 1)/2

Description

Squared variable means

Crossproduct variable means

BC — Vector of length $NVAR * (NVAR + 3)/2 + 1$ containing the coefficients for the centered variables. (Input)

Here, the fitted model is

$$\hat{y} = BC(1) + \sum_{j=1}^{NVAR} BC(j) * z_j + \sum_{j=1}^{NVAR} BC(j) * (z_j^2 - SCPM(j)) \\ + \sum_{j=1}^{NVAR} \sum_{k=j+1}^{NVAR} B(1 + NVAR + m) * (z_j z_k - SCPM(m(j,k)))$$

where $z_j = x_j - XMEAN(j)$ and $m_{jk} = j * NVAR - j(j - 1)/2 + k - j$. These regression coefficients can come from a regression using variables generated by routine GCSCP (page 296) with the option ICEN = 1.

B — Vector of length $NVAR * (NVAR + 3)/2 + 1$ containing the coefficients of the uncentered variables. (Output)

Here, the model uses the original x variables, i.e.,

$$\hat{y} = B(1) + \sum_{j=1}^{NVAR} B(j) * x_j + \sum_{j=1}^{NVAR} B(j) * x_j^2 \\ + \sum_{j=1}^{NVAR} \sum_{k=j+1}^{NVAR} B(1 + NVAR + m) * x_j x_k$$

Optional Arguments

NVAR — Number of variables. (Input)

Default: $NVAR = \text{size}(XMEAN, 1)$.

FORTRAN 90 Interface

Generic: `CALL TCSCP (XMEAN, SCPM, BC, B [, ...])`

Specific: The specific interface names are `S_TCSCP` and `D_TCSCP`.

FORTRAN 77 Interface

Single: `CALL TCSCP (NVAR, XMEAN, SCPM, BC, B)`

Double: The double precision name is `DTCSKP`.

Example

This example transforms coefficients from a second-order response surface model with three independent variables fitted using squares and crossproducts of centered variables into a model using the original uncentered variables.

```
USE TCSCP_INT
USE WRRRN_INT
```

```

INTEGER      NVAR
PARAMETER    (NVAR=3)
!
REAL         B (NVAR*(NVAR+3)/2+1), BC (NVAR*(NVAR+3)/2+1), &
             SCPM (NVAR*(NVAR+1)/2), XMEAN (NVAR)
!
DATA XMEAN/10.0, 11.0, 6.0/
DATA SCPM/12.0, 5.0, 2.0, 3.0, 7.0, 1.0/
DATA BC/1.0, 2.0, 3.0, 0.0, 5.0, 0.0, 7.0, 0.0, 9.0, 10.0/
!
CALL TCSCP (XMEAN, SCPM, BC, B)
!
CALL WRRRN ('B', B, 1, NVAR*(NVAR+3)/2+1, 1)
!
END

```

Output

				B				
1	2	3	4	5	6	7	8	
1753.0	-152.0	-57.0	-284.0	5.0	0.0	7.0	0.0	
9	10							
9.0	10.0							

Comments

Crossproduct variables are ordered as follows: (1, 2), (1, 3), ..., (1, NVAR), (2, 3), (2, 4), ..., (2, NVAR), ..., (NVAR - 1, NVAR).

Description

Routine `TCSCP` transforms coefficients from a second-order response surface model fitted using squares and crossproducts of centered variables into a model using the original uncentered variables. Let x_{ij} be the i -th setting of the j -th variable ($i = 1, 2, \dots, n; j = 1, 2, \dots, m$). Denote the means (stored in `XMEAN`) by

$$\bar{x}_j \quad (j = 1, 2, \dots, m)$$

The settings of the j -th centered variable are given by

$$z_{ij} = x_{ij} - \bar{x}_j$$

The settings of the j -th squared variable are given by

$$z_{ij}^2 - \bar{z}_j^2$$

where

$$\bar{z}_j^2 = \frac{\sum_{i=1}^n z_{ij}^2}{n}$$

(stored in $(m + j)$ -th column of `SCPM`) is the mean of the j -th squared variable. The settings of the jk crossproduct variable are given by

$$z_{ij}z_{ik} - \overline{z_j z_k}$$

where

$$\overline{z_j z_k} = \sum_{i=1}^n \frac{z_{ij}z_{ik}}{n}$$

(stored in the

$$k - j + mj - \frac{j(j-1)}{2}$$

location of SCPM) is the mean of the jk -th ($j < k$) crossproduct variable. The fitted model is

$$\hat{y}_i = \hat{\alpha}_0 + \sum_{j=1}^m \hat{\alpha}_j z_{ij} + \sum_{j=1}^m \hat{\alpha}_{jj} (z_{ij}^2 - \overline{z_j^2}) + \sum_{j=2}^{m-1} \sum_{k=j+1}^m \hat{\alpha}_{jk} (z_{ij}z_{ik} - \overline{z_j z_k})$$

TCSCP transforms the

$$\hat{\alpha}_j \text{'s, the } \hat{\alpha}_{jj} \text{'s, and the } \hat{\alpha}_{jk} \text{'s}$$

to regression coefficients for the original independent variables. The fitted transformed model is

$$\hat{y}_i = \hat{\beta}_0 + \sum_{j=1}^m \hat{\beta}_j x_{ij} + \sum_{j=1}^m \hat{\beta}_{jj} x_{ij}^2 + \sum_{j=2}^{m-1} \sum_{k=j+1}^m \hat{\beta}_{jk} x_{ij} x_{ik}$$

where

$$\begin{aligned} \hat{\beta}_0 &= \hat{\alpha}_0 - \sum_{j=1}^m \hat{\alpha}_j \overline{x_j} - \sum_{j=1}^m \hat{\alpha}_{jj} \overline{z_j^2} - \sum_{j=1}^{m-1} \sum_{k=j+1}^m \hat{\alpha}_{jk} \overline{z_j z_k} + \sum_{j=1}^m \hat{\alpha}_{jj} \overline{x_j^2} \\ &\quad + \sum_{j=1}^{m-1} \sum_{k=j+1}^m \hat{\alpha}_{jk} \overline{x_j x_k} \\ \hat{\beta}_j &= \hat{\alpha}_j - 2\hat{\alpha}_{jj} \overline{x_j} - \sum_{k=1}^{m-1} \hat{\alpha}_{kj} \overline{x_k} - \sum_{k=j+1}^m \hat{\alpha}_{jk} \overline{x_k} \\ \hat{\beta}_{jj} &= \hat{\alpha}_{jj} \\ \hat{\beta}_{jk} &= \hat{\alpha}_{jk} \end{aligned}$$

RNLIN

Fits a nonlinear regression model.

Required Arguments

FUNC — User-supplied SUBROUTINE to return the weight, frequency, residual, and optionally the derivative of the residual at the given parameter vector THETA for a single observation. The usage is:

CALL FUNC (NPARM, THETA, IOPT, IOBS, FRQ, WT, E, DE, IEND),
where

NPARM – Number of unknown parameters in the regression function. (Input)

THETA – Vector of length NPARM containing parameter values. (Input)

IOPT – Function/derivative evaluation option. (Input)

IOPT Meaning

0 Evaluate the function.

1 Evaluate the derivative.

If *IDERIV* = 0, only *IOPT* = 0 is used.

IOBS – Observation number. (Input)

The function is evaluated at the *IOBS*-th observation.

FRQ – Frequency for the observation. (Output)

WT – Weight for the observation. (Output)

Use *WT* = 1.0 for equal weighting (unweighted least squares).

E – Error (residual) for the *IOBS*-th observation. (Output, if *IOPT* = 0)

DE – Vector of length NPARM containing the partial derivatives of the residual for the *IOBS*-th observation. (Output, if *IOPT* = 1)

If *IDERIV* = 0, *DE* is not referenced and can be a vector of length one.

IEND – Completion indicator. (Output)

IEND Meaning

0 *IOBS* is less than or equal to the number of observations.

1 *IOBS* is greater than the number of observations. *WT*, *FRQ*, *E*, and *DE* are not output.

FUNC must be declared EXTERNAL in the calling program.

THETA — Vector of length NPARM containing parameter values. (Input/Output)

On input, *THETA* must contain the initial estimate. On output, *THETA* contains the final estimate.

Optional Arguments

NPARM — Number of unknown parameters in the regression function. (Input)

Default: *NPARM* = size (*THETA*,1).

IDERIV — Derivative option. (Input)

Default: *IDERIV* = 0.

IDERIV Meaning

0 Derivatives are obtained by finite differences.

1 Derivatives are supplied by *FUNC*.

R — *NPARM* by *NPARM* upper triangular matrix containing the *R* matrix from a *QR* decomposition of the Jacobian. (Output)

LDR — Leading dimension of R exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDR= size (R, 1)`.

IRANK — Rank of R . (Output)

`IRANK` less than `NPARM` may indicate the model is overparameterized.

DFE — Degrees of freedom for error. (Output)

SSE — Sums of squares for error. (Output)

FORTRAN 90 Interface

Generic: `CALL RNLIN (FUNC, THETA [, ...])`

Specific: The specific interface names are `S_RNLIN` and `D_RNLIN`.

FORTRAN 77 Interface

Single: `CALL RNLIN (FUNC, NPARM, IDERIV, THETA, R, LDR, IRANK, DFE, SSE)`

Double: The double precision name is `DRNLIN`.

Example 1

This example uses data discussed by Neter, Wasserman, and Kutner (1983, pages 475–478). A nonlinear model

$$y_i = \theta_1 e^{\theta_2 x_i} + \varepsilon_i \quad i = 1, 2, \dots, 15$$

is fitted. The option `IDERIV = 0` is used.

The user must supply a `SUBROUTINE` to return the residual, weight, and frequency for a single observation at the given value of the regression parameter vector θ . This subroutine, called `EXAMPL` here, must be declared `EXTERNAL` in the calling program and must have the specified calling sequence.

```
USE RNLIN_INT
USE UMACH_INT
USE WRRRN_INT

INTEGER    LDR, NOBS, NPARM
PARAMETER (NOBS=15, NPARM=2, LDR=NPARM)
!
INTEGER    IRANK, NOUT
REAL      DFE, R(LDR, NPARM), SSE, THETA(NPARM)
EXTERNAL  EXAMPL
!
DATA THETA/60.0, -0.03/
!
```



```

CALL UMACH (2, NOUT)
!
CALL RNLIN (EXAMPL, THETA, R=R, IRANK=IRANK, DFE=DFE, SSE=SSE)
WRITE (NOUT,*) 'THETA = ', THETA
WRITE (NOUT,*) 'IRANK = ', IRANK, ' DFE = ', DFE, ' SSE = ', &
SSE
CALL WRRRN ('R', R)
END
!
SUBROUTINE EXAMPL (NPARM, THETA, IOPT, IOBS, FRQ, WT, E, DE, &
IEND)
INTEGER NPARM, IOPT, IOBS, IEND
REAL THETA(NPARM), FRQ, WT, E, DE(1)
!
INTEGER NOBS
PARAMETER (NOBS=15)
!
REAL EXP, XDATA(NOBS), YDATA(NOBS)
INTRINSIC EXP
!
DATA YDATA/54.0, 50.0, 45.0, 37.0, 35.0, 25.0, 20.0, 16.0, 18.0, &
13.0, 8.0, 11.0, 8.0, 4.0, 6.0/
DATA XDATA/2.0, 5.0, 7.0, 10.0, 14.0, 19.0, 26.0, 31.0, 34.0, &
38.0, 45.0, 52.0, 53.0, 60.0, 65.0/
!
IF (IOBS .LE. NOBS) THEN
WT = 1.0E0
FRQ = 1.0E0
IEND = 0
E = YDATA(IOBS) - THETA(1)*EXP(THETA(2)*XDATA(IOBS))
ELSE
IEND = 1
END IF
RETURN
END

```

Output

```

THETA = 58.6045 -3.95835E-02
IRANK = 2 DFE = 13.0000 SSE = 49.4593

R
1 1.9 1139.8
2 0.0 1139.7

```

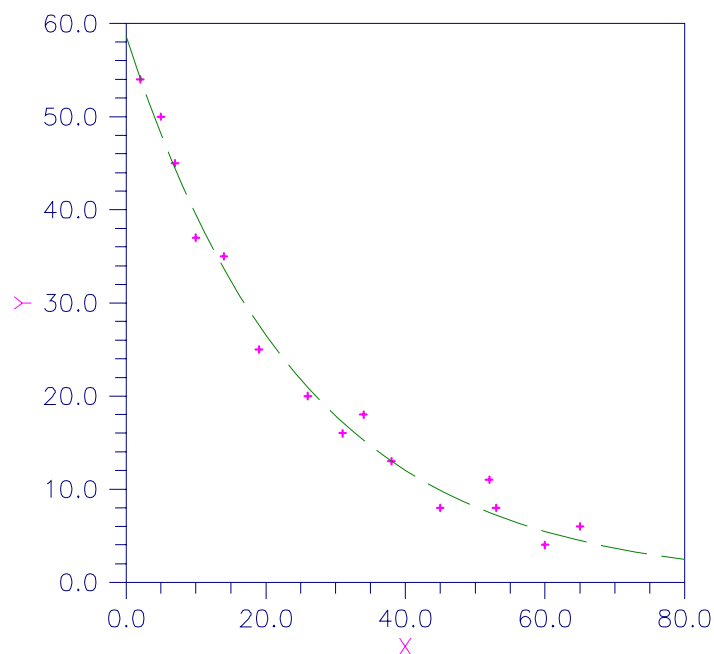


Figure 2-10 Plot of the Nonlinear Fit

Comments

1. Workspace may be explicitly provided, if desired, by use of R2LIN/DR2LIN. The reference is:

```
CALL R2LIN (FUNC, NPARM, IDERIV, THETA, R, LDR, IRANK, DFE, SSE,
IPARAM, RPARAM, SCALE, IWK, WK)
```

The additional arguments are as follows:

IPARAM — Vector of length 6 containing convergence parameters. (Input/Output)
On input, set `IPARAM(1) = 0` for default convergence parameter settings. If `IPARAM(1) = 0`, the remaining elements of `IPARAM`, and the arguments `RPARAM` and `SCALE` need not be initialized.

I	Name	IPARAM(I)
1	INIT	Initialization flag. (Input) INIT = 0 means use default settings for IPARAM, RPARAM, and SCALE. INIT = 1 means use the input IPARAM and RPARAM settings.
2	NDIGIT	Number of good digits in the residuals. (Input, if IPARAM(1) = 1)

3	ITER	Number of iterations. (Input/Output, if IPARAM(1) = 1; output, otherwise) On input, this is the maximum number of iterations allowed. The default is 100. On output, it is the actual number of iterations.
4	NFCN	Number of SSE evaluations. (Input/Output, if IPARAM(1) = 1; output, if IPARAM(1) = 0) On input, this is the maximum number of evaluations allowed. The default is 400. On output, it is the actual number of evaluations.
5	NJAC	Number of Jacobian evaluations. (Input, if IPARAM(1) = 1 and IDERIV = 1; output, if IDERIV = 1) On input, this is the maximum number of Jacobian evaluations allowed. The default is 100. On output, it is the number of Jacobian evaluations.
6	MODE	Scaling option. (Input, if IPARAM(1) = 1) If IPARAM(6) = 1, the values for SCALE are set internally. The default is 1. Otherwise, SCALE must be input.

RPARAM — Vector of length 7 containing convergence parameters. (Input, if IPARAM(1) = 1)

In the following table, the default settings are given in parentheses.

EPS = AMACH(4) (see the documentation for IMSL routine AMACH).

I	Name	RPARAM(I)
1	FJACTL	Scaled gradient tolerance (SQRT(EPS) for single precision; $\text{EPS}^{1/3}$ for double precision) Convergence is declared if $ \text{WK}(I) * \max\{ \text{THETA}(I) , 1.0/\text{SCALE}(I)\}/\text{SSE}$ is less than FJACTL for $I = 1, 2, \dots, \text{NPARM}$, where $\text{WK}(I)$ is the I -th component of the gradient vector.
2	STEPTL	Scaled step tolerance ($\text{EPS}^{2/3}$) Convergence is declared if $ \text{WK}(\text{NPARM} + I) /\max\{ \text{THETA}(I) , 1.0/\text{SCALE}(I)\}$ is less than STEPTL for $I = 1, 2, \dots, \text{NPARM}$, where $\text{WK}(\text{NPARM} + I)$ is the I -th component of the last step.
3	RFTOL	Relative function tolerance ($\max\{10^{-10}, \text{EPS}^{2/3}\}$ for single precision, $\max\{10^{-20}, \text{EPS}^{2/3}\}$ for double precision) Convergence is declared if the change in SSE is less than or equal to RFTOL * SSE in absolute value.

- | | | |
|---|--------|--|
| 4 | AFTOL | Absolute function tolerance ($\max\{10^{-20}, \text{EPS}^2\}$ for single precision; $\max\{10^{-40}, \text{EPS}^2\}$ for double precision)

Convergence is declared if SSE is less than AFTOL. |
| 5 | FALSTL | False convergence tolerance ($100.0 * \text{EPS}$) |
| 6 | STEPMX | Maximum allowable step size ($1000 * \max(\text{TOL1}, \text{TOL2})$ where $\text{TOL1} = \text{SNRM2}(\text{NPARM}, \text{SCXTH}, 1)$; $\text{TOL2} = \text{SNRM2}(\text{NPARM}, \text{SCALE}, 1)$ and SCXTH is the elementwise product of SCALE and THETA, i.e., $\text{SCXTH}(\text{I}) = \text{SCALE}(\text{I}) * \text{THETA}(\text{I})$.) |
| 7 | DELTA | Size of initial trust region radius (based on the initial scaled Cauchy step) |

SCALE — Vector of length NPARM. (Input/Output, if IPARAM(1) = 1 and IPARAM(6) = 0; output, if IPARAM(6) = 1)

A common choice is to set all elements of SCALE to 1.0. If good starting values for THETA are known and nonzero, a good choice is $\text{SCALE}(\text{I}) = 1.0/|\text{THETA}(\text{I})|$. Otherwise, for example, if THETA(I) is known to be in the interval $(-10^5, 10^5)$, set $\text{SCALE}(\text{I}) = 10^{-5}$. Or, for example, if THETA(I) is known to be in the interval $(10^3, 10^5)$, set $\text{SCALE}(\text{I}) = 10^{-4}$.

IWK — Work vector of length NPARM.

WK — Work vector of length $11 * \text{NPARM} + 4$. (Output)

The first NPARM components of WK are the gradient at the solution. The second NPARM components of WK give the last step.

2. Informational errors

Type Code

- | | | |
|---|---|---|
| 3 | 1 | Both the scaled actual and predicted reductions in the function are less than or equal to the relative function convergence tolerance. |
| 3 | 2 | The iterates appear to be converging to a noncritical point. Incorrect gradient information, a discontinuous function, or stopping tolerances being too tight may be the cause. |
| 4 | 3 | Maximum number of iterations is exceeded. |
| 4 | 4 | Maximum number of function evaluations is exceeded. |
| 4 | 5 | Maximum number of Jacobian evaluations is exceeded for IDERIV = 1. |
| 3 | 6 | Five consecutive steps of the same size have been taken. Either the function is unbounded below, or it has a finite asymptote in some direction, or the stepsize is too small. |
| 2 | 7 | Scaled step tolerance is satisfied, the current point may be an approximate local solution, or the algorithm is making very slow progress and is not near a solution, or STEPTL is too big. |

3. The first stopping criterion for `RNLIN` occurs when `SSE` is less than the absolute function tolerance. The second stopping criterion occurs when the norm of the scaled gradient is less than the given gradient tolerance. The third stopping criterion occurs when the scaled distance between the last two steps is less than the step tolerance. The third stopping criterion also generates error code 7. The fourth stopping criterion occurs when the relative change in `SSE` is less than `RFTOL`. The fourth stopping criterion also generates error code 1. See Dennis and Schnabel (1983, pages 159–161, 278–280, and 347–348) for a discussion of stopping criteria and choice of tolerances.
4. To use some nondefault convergence parameters, first call `R8LIN`, then reset the corresponding convergence parameters to the desired value and call `R2LIN`. For example, the following code could be used if nondefault convergence parameters are to be used:

```

!
      CALL R8LIN (IPARAM, RPARAM)
! R8LIN outputs IPARAM(1) = 1 to indicate some
! nondefault convergence parameters are to be set.
! R8LIN outputs the remaining elements of IPARAM
! and RPARAM as their default values.
!
! Set some nondefault convergence parameters.
      IPARAM(3) = 20
      IPARAM(6) = 0
      SCALE(1) = 0.1
      SCALE(2) = 10.0
!
      CALL R2LIN (FUNC, NPARAM, IDERIV, THETA, R, &
                LDR, IRANK, DFE, SSE, IPARAM, &
                RPARAM, SCALE, IWK, WK)

```

If double precision is being used, then `DR8LIN` and `DR2LIN` are called and `RPARAM` is declared double precision.

Description

Routine `RNLIN` fits a nonlinear regression model using least squares. The nonlinear regression model is

$$y_i = f(x_i; \theta) + \varepsilon_i \quad i = 1, 2, \dots, n$$

where the observed values of the y_i 's constitute the responses or values of the dependent variable, the known x_i 's are the vectors of the values of the independent (explanatory) variables, θ is the vector of p regression parameters, and the ε_i 's are independently distributed normal errors with mean zero and variance σ^2 . For this model, a least squares estimate of θ is also a maximum likelihood estimate of θ .

The residuals for the model are

$$e_i(\theta) = y_i - f(x_i; \theta) \quad i = 1, 2, \dots, n$$

A value of θ that minimizes

$$\sum_{i=1}^n [e_i(\theta)]^2$$

is a least squares estimate of θ . Routine `RNLIN` is designed so that these residuals are input one at a time from a user-supplied subroutine. This permits `RNLIN` to handle the case when n is large and the data cannot reside in an array but must reside on some secondary storage device.

Routine `RNLIN` is based on MINPACK routines `LMDIF` and `LMDER` by Moré et al. (1980). The routine `RNLIN` uses a modified Levenberg-Marquardt method to generate a sequence of approximations to a minimum point. Let

$$\hat{\theta}_c$$

be the current estimate of θ . A new estimate is given by

$$\hat{\theta}_c + s_c$$

where s_c is a solution to

$$(J(\hat{\theta}_c)^T J(\hat{\theta}_c) + \mu_c I) s_c = J(\hat{\theta}_c)^T e(\hat{\theta}_c)$$

Here,

$$J(\hat{\theta}_c)$$

is the Jacobian evaluated at

$$\hat{\theta}_c$$

The algorithm uses a “trust region” approach with a step bound of δ_c . A solution of the equations is first obtained for $\mu_c = 0$. If $\|s_c\|_2 < \delta_c$, this update is accepted. Otherwise, μ_c is set to a positive value and another solution is obtained. The method is discussed by Levenberg (1944), Marquardt (1963), and Dennis and Schnabel (1983, pages 129–147, 218–338).

If `IDERIV` = 0, forward finite differences are used to estimate the Jacobian numerically. If `IDERIV` = 1, the Jacobian is computed analytically via the user-supplied subroutine. With `IDERIV` = 0 and single precision arithmetic, the estimate of the Jacobian may be so poor that the algorithm terminates at a noncritical point. In such instances, `IDERIV` = 1 or double precision arithmetic is recommended.

Routine `RNLIN` does not actually store the Jacobian but uses fast Givens transformations to construct an orthogonal reduction of the Jacobian to upper triangular form (stored in `R`). The reduction is based on fast Givens transformations (see routines `SROTMG` and `SROTM`, Golub and Van Loan 1983, pages 156–162, Gentleman 1974). This method has two main advantages: (1) the loss of accuracy resulting from forming the crossproduct matrix used in the equations for s_c is avoided, and (2) the $n \times p$ Jacobian need not be stored saving space when $n > p$.

A weighted least squares fit can also be performed. This is appropriate when the variance of ε_i in the nonlinear regression model is not constant but instead is σ^2/w_i . Here, the w_i 's are weights input via the user-supplied subroutine. For the weighted case, `RNLIN` computes a minimum weighted sum of squares for error (stored in `SSE`).

Programming Notes

Nonlinear regression allows substantial flexibility over linear regression because the user can specify the functional form of the model. This added flexibility can cause unexpected convergence problems for users that are unaware of the limitations of the software. Also, in many cases, there are possible remedies that may not be immediately obvious. The following is a list of possible convergence problems and some remedies that the user can try. There is not a one-to-one correspondence between the problems and the remedies. Remedies for some problems may also be relevant for the other problems.

1. A local minimum is found. Try a different starting value. Good starting values often can be obtained by fitting simpler models. For example, for a nonlinear function

$$f(x; \theta) = \theta_1 e^{\theta_2 x}$$

good starting values can be obtained from the estimated linear regression coefficients

$$\hat{\beta}_0 \text{ and } \hat{\beta}_1$$

from a simple linear regression of $\ln y$ on $\ln x$. The starting values for the nonlinear regression in this case would be

$$\theta_1 = e^{\hat{\beta}_0} \text{ and } \theta_2 = \hat{\beta}_1$$

If an approximate linear model is not clear, then simplify the model by reducing the number of nonlinear regression parameters. For example, some nonlinear parameters for which good starting values are known could be set to these values in order to simplify the model for computing starting values for the remaining parameters.

2. The estimate of θ is incorrectly returned as the same or very close to the initial estimate
 - The scale of the problem may be orders of magnitude smaller than the assumed default of 1 causing premature stopping. For example, in single precision if SSE is less than $AMACH(4)**2$, the routine stops. See Example 3, which shows how to shut down some of the stopping criteria that may not be relevant for your particular problem and which also shows how to improve the speed of convergence by the input of the scale of the model parameters.
 - The scale of the problem may be orders of magnitude larger than the assumed default causing premature stopping. The information with regard to the input of the scale of the model parameters in Example 3 is also relevant here. In addition, the maximum allowable step size, $RPARAM(6)$ in Example 3, may need to be increased.
 - The residuals are input with accuracy much less than machine accuracy causing premature stopping because a local minimum is found. Again see Example 3 to see generally how to change some default tolerances. If you cannot improve the precision of the computations of the residual, you need to set $IPARAM(2)$ to indicate the actual number of good digits in the residuals.

3. The model is discontinuous as a function of θ . You may have a mistake in the subroutine you supplied. Note that the function $f(x; \theta)$ can be a discontinuous function of x .
4. The R matrix returned by `RNLIN` is inaccurate. Use the double precision version `DRNLIN`. If `IDERIV = 1`, check your derivatives or try using `IDERIV = 0`. If `IDERIV = 0`, try using `IDERIV = 1`.
5. Overflow occurs during the computations. Print out θ and the residual in the subroutine you supplied. Make sure the code you supply does not overflow at some value of θ .
6. The estimate of θ is going to infinity. You may need to reparameterize or change your function. For example, a parameterization in terms of the reciprocals may be appropriate.
7. Some components of θ are outside known bounds. Routine `RNLIN` does not handle bounds on the parameters, but you can artificially impose some by setting the residuals unusually large outside the bounds. Although this introduces a discontinuity in the model function, this often works and allows you to use `RNLIN` without having to resort to a more general nonlinear optimization routine.

Additional Examples

Example 2

This example fits the model in Example 1 with the option `IDERIV = 1`.

```

USE RNLIN_INT
USE UMACH_INT
USE WRRRN_INT

INTEGER    LDR, NOBS, NPARM
PARAMETER (NOBS=15, NPARM=2, LDR=NPARM)
!
INTEGER    IDERIV, IRANK, NOUT
REAL       DFE, R(LDR, NPARM), SSE, THETA(NPARM)
EXTERNAL   EXAMPL
!
DATA THETA/60.0, -0.03/
!
CALL UMACH (2, NOUT)
!
IDERIV = 1
CALL RNLIN (EXAMPL, THETA, IDERIV=IDERIV, R=R, IRANK=IRANK, DFE=DFE, &
           SSE=SSE)
WRITE (NOUT, *) 'THETA = ', THETA
WRITE (NOUT, *) 'IRANK = ', IRANK, ' DFE = ', DFE, ' SSE = ', &
           SSE
CALL WRRRN ('R', R)
END
!
SUBROUTINE EXAMPL (NPARM, THETA, IOPT, IOBS, FRQ, WT, E, DE, &

```



```

                                IEND)
INTEGER      NPARM, IOPT, IOBS, IEND
REAL        THETA(NPARM), FRQ, WT, E, DE(NPARM)
!
INTEGER      NOBS
PARAMETER   (NOBS=15)
!
REAL        EXP, XDATA(NOBS), YDATA(NOBS)
INTRINSIC   EXP
!
DATA YDATA/54.0, 50.0, 45.0, 37.0, 35.0, 25.0, 20.0, 16.0, 18.0, &
      13.0, 8.0, 11.0, 8.0, 4.0, 6.0/
DATA XDATA/2.0, 5.0, 7.0, 10.0, 14.0, 19.0, 26.0, 31.0, 34.0, &
      38.0, 45.0, 52.0, 53.0, 60.0, 65.0/
!
IF (IOBS .LE. NOBS) THEN
  WT   = 1.0E0
  FRQ  = 1.0E0
  IEND = 0
  IF (IOPT .EQ. 0) THEN
    E = YDATA(IOBS) - THETA(1)*EXP(THETA(2)*XDATA(IOBS))
  ELSE
    DE(1) = -EXP(THETA(2)*XDATA(IOBS))
    DE(2) = -THETA(1)*XDATA(IOBS)*EXP(THETA(2)*XDATA(IOBS))
  END IF
ELSE
  IEND = 1
END IF
RETURN
END

```

Output

```

THETA =      58.6034      -3.95812E-02
IRANK =   2   DFE =      13.0000   SSE =      49.4593

      R
      1      2
1      1.9  1140.1
2      0.0  1139.9

```

Example 3

This example fits the model in Example 1, but the data for y is 10^{-10} times the values in Example 1. In order to solve this problem without rescaling y , we use some nondefault convergence tolerances and scales. This is accomplished by invoking routine `R8LIN`, setting some elements of `IPARAM`, `RPARAM`, and `SCALE`, and then invoking `R2LIN`. Here, we set the absolute function tolerance to 0.0. The default value would cause the program to terminate after one iteration because the residual sum of squares is roughly 10^{-19} . Also, we set the relative function tolerance to 0.0. The gradient stopping condition is properly scaled for this problem so we leave it at its default value. Finally, we set `SCALE(I)` equal to the absolute value of the reciprocal of the starting value.

Note in the output that the estimate of θ_1 is 10^{-10} times the estimate in Example 1. Note also that the invocation of R2LIN in place of RNLIN allows the printing of additional information that is output in IPARAM (number iterations and number of SSE evaluations) and output in WK (gradient at solution and last step).

```

USE UMACH_INT
USE R8LIN_INT
USE R2LIN_INT
USE WROPT_INT
USE WRRRN_INT

INTEGER    LDR, NOBS, NPARM
PARAMETER  (NOBS=15, NPARM=2, LDR=NPARM)

!

INTEGER    IPARAM(6), IRANK, IWK(NPARM), NOUT
REAL       ABS, DFE, R(LDR,NPARM), RPARAM(7), SCALE(NPARM), SSE, &
           THETA(NPARM), WK(11*NPARM+4)

INTRINSIC  ABS
EXTERNAL   EXAMPL

!

DATA THETA/6.0E-9, -0.03/

!

CALL UMACH (2, NOUT)

!

CALL R8LIN (IPARAM, RPARAM)
RPARAM(3) = 0.0
RPARAM(4) = 0.0
IPARAM(6) = 0
SCALE(1)  = 1.0/ABS(THETA(1))
SCALE(2)  = 1.0/ABS(THETA(2))
CALL R2LIN (EXAMPL, NPARM, IDERIV, THETA, R, LDR, IRANK, DFE, &
           SSE, IPARAM, RPARAM, SCALE, IWK, WK)
WRITE (NOUT,*) 'THETA = ', THETA
WRITE (NOUT,*) 'IRANK = ', IRANK, ' DFE = ', DFE, ' SSE = ', &
           SSE
WRITE (NOUT,*) 'Number of iterations = ', IPARAM(3)
WRITE (NOUT,*) 'Number of SSE evaluations = ', IPARAM(4)
ISETNG=2
CALL WROPT (-6, ISETNG, 1)
CALL WRRRN ('Gradient at solution', WK, 1, NPARM, 1)
CALL WRRRN ('Last step taken', WK((NPARM+1):), 1, NPARM, 1)
CALL WRRRN ('R', R)
END

!

SUBROUTINE EXAMPL (NPARM, THETA, IOPT, IOBS, FRQ, WT, E, DE, &
                 IEND)
INTEGER    NPARM, IOPT, IOBS, IEND
REAL       THETA(NPARM), FRQ, WT, E, DE(1)

!

INTEGER    NOBS
PARAMETER  (NOBS=15)

!

REAL       EXP, XDATA(NOBS), YDATA(NOBS)
INTRINSIC  EXP
!

```

```

DATA YDATA/54.0E-10, 50.0E-10, 45.0E-10, 37.0E-10, 35.0E-10, &
      25.0E-10, 20.0E-10, 16.0E-10, 18.0E-10, 13.0E-10, 8.0E-10, &
      11.0E-10, 8.0E-10, 4.0E-10, 6.0E-10/
DATA XDATA/2.0, 5.0, 7.0, 10.0, 14.0, 19.0, 26.0, 31.0, 34.0, &
      38.0, 45.0, 52.0, 53.0, 60.0, 65.0/
!
IF (IOBS .LE. NOBS) THEN
  WT   = 1.0E0
  FRQ  = 1.0E0
  IEND = 0
  E    = YDATA(IOBS) - THETA(1)*EXP(THETA(2)*XDATA(IOBS))
ELSE
  IEND = 1
END IF
RETURN
END

```

Output

```

THETA =      5.86076E-09      -3.95879E-02
RANK =  2  DFE =      13.0000  SSE =      4.94593E-19
Number of iterations =      5
Number of SSE evaluations =     13

```

```

      Gradient at solution
           1           2
6.86656E-14  -1.73762E-20

```

```

      Last step taken
           1           2
-3.24588E-14   3.65805E-07

```

```

           R
           1           2
1  1.87392E+00  1.13981E-07
2  0.00000E+00  1.13934E-07

```

Example 4

For an extended version of Example 2 that in addition computes the estimated asymptotic variance-covariance matrix of the estimated nonlinear regression parameters, see Example 2 for routine `RCOVB` ([page 164](#)). The example also computes confidence intervals for the parameters.

Example 5

For an extended version of Example 2 that in addition computes standardized residuals, leverages, and confidence intervals on the mean response, see Example 2 for routine `ROTIN` ([page 215](#)).

RLAV

Fits a multiple linear regression model using the least absolute values criterion.

Required Arguments

X — *NOBS* by *NCOL* matrix containing the data. (Input)

IIND — Independent variable option. (Input)

The absolute value of *IIND* is the number of independent (explanatory) variables. The sign of *IIND* specifies the following options:

IIND Meaning

- < 0 The data for the $-IIND$ independent variables are given in the first $-IIND$ columns of *X*.
- > 0 The data for the *IIND* independent variables are in the columns of *X* whose column numbers are given by the elements of *INDIND*.
- = 0 There are no independent variables.

The regressors are the constant regressor (if *INTCEP* = 1) and the independent variables.

INDIND — Index vector of length *IIND* containing the column numbers of *X* that are the independent (explanatory) variables. (Input, if *IIND* is positive)

If *IIND* is negative, *INDIND* is not referenced and can be a vector of length one.

IRSP — Column number *IRSP* of *X* contains the data for the response (dependent) variable. (Input)

B — Vector of length *INTCEP* + $|IIND|$ containing a *LAV* solution for the regression coefficients. (Output)

If *INTCEP* = 1, *B*(1) contains the intercept estimate. *B*(*INTCEP* + *I*) contains the coefficient estimate for the *I*-th independent variable.

Optional Arguments

NOBS — Number of observations. (Input)

Default: *NOBS* = size (*X*,1).

NCOL — Number of columns in *X*. (Input)

Default: *NCOL* = size (*X*,2).

LDX — Leading dimension of *X* exactly as specified in the dimension statement in the calling program. (Input)

Default: *LDX* = size (*X*,1).

INTCEP — Intercept option. (Input)

Default: *INTCEP* = 1.

INTCEP Action

0 An intercept is not in the model.

1 An intercept is in the model.

IRANK — Rank of the matrix of regressors. (Output)

If **IRANK** is less than **INTCEP** + **|IIND|**, linear dependence of the regressors was declared.

SAE — Sum of the absolute values of the errors. (Output)

ITER — Number of iterations performed. (Output)

NRMISS — Number of rows of data containing NaN (not a number) for the dependent or independent variables. (Output)

If a row of data contains NaN for any of these variables, that row is excluded from the computations.

FORTRAN 90 Interface

Generic: CALL RLAV (X, IIND, INDIND, IRSP, B [...])

Specific: The specific interface names are **S_RLAV** and **D_RLAV**.

FORTRAN 77 Interface

Single: CALL RLAV (NOBS, NCOL, X, LDX, INTCEP, IIND, INDIND, IRSP, B, IRANK, SAE, ITER, NRMISS)

Double: The double precision name is **DRLAV**.

Example

A straight line fit to a data set is computed under the LAV criterion.

```
!                               SPECIFICATIONS FOR PARAMETERS
USE RLAV_INT
USE UMACH_INT
USE WRRRL_INT
INTEGER    LDX, NCOEF, NCOL, NOBS
PARAMETER (NCOEF=2, NCOL=2, NOBS=8, LDX=NOBS)
!
INTEGER    IIND, INDIND(1), IRANK, IRSP, ITER, NOUT, &
           NRMISS
REAL       B(NCOEF), SAE, X(LDX,NCOL)
CHARACTER  CLABEL(1)*4, RLABEL(1)*4
!
DATA (X(1,J),J=1,NCOL) /1.0, 1.0/
DATA (X(2,J),J=1,NCOL) /4.0, 5.0/
```

```

DATA (X(3,J),J=1,NCOL) /2.0, 0.0/
DATA (X(4,J),J=1,NCOL) /2.0, 2.0/
DATA (X(5,J),J=1,NCOL) /3.0, 1.5/
DATA (X(6,J),J=1,NCOL) /3.0, 2.5/
DATA (X(7,J),J=1,NCOL) /4.0, 2.0/
DATA (X(8,J),J=1,NCOL) /5.0, 3.0/
!
IIND = -1
IRSP = 2
!
CALL RLAV (X, IIND, INDIND, IRSP, B, IRANK=IRANK, SAE=SAE, &
          ITER=ITER, NRMISS=NRMISS)
!
CALL UMACH (2, NOUT)
RLABEL(1) = 'B ='
CLABEL(1) = 'NONE'
CALL WRRRL (' ', B, RLABEL, CLABEL, 1, NCOEF, 1, FMT='(F6.2)')
WRITE (NOUT,*) 'IRANK = ', IRANK
WRITE (NOUT,*) 'SAE = ', SAE
WRITE (NOUT,*) 'ITER = ', ITER
WRITE (NOUT,*) 'NRMISS = ', NRMISS
END

```

Output

```

B =      0.50      0.50
IRANK =      2
SAE =      6.00000
ITER =      2
NRMISS =      0

```

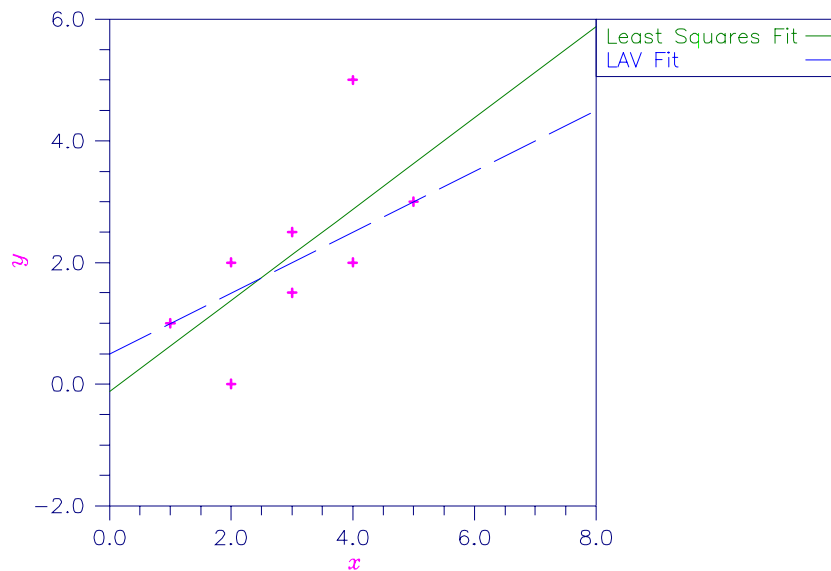


Figure 2-11 Least Squares and Least Absolute Value Fitted Lines

Comments

1. Workspace may be explicitly provided, if desired, by use of `R2AV/DR2AV`. The reference is:

```
CALL R2AV (NOBS, NCOL, X, LDX, INTCEP, IIND, INDIND,  
IRSP, B, IRANK, SAE, ITER, NRMISS, IWK, WK)
```

The additional arguments are as follows:

IWK — Work vector of length `NOBS`

WK — Work vector of length `NOBS * (|IIND| + 5) + 2 * |IIND| + 4`

2. Informational error
Type Code

3 1 The solution may not be unique.

Description

Routine `RLAV` computes estimates of the regression coefficients in a multiple linear regression model. The criterion satisfied is the minimization of the sum of the absolute values of the deviations of the observed response y_i from the fitted response

$$\hat{y}_i$$

for a set on n observations. Under this criterion, known as the L_1 or LAV (least absolute value) criterion, the regression coefficient estimates minimize

$$\sum_{i=1}^n |y_i - \hat{y}_i|$$

The estimation problem can be posed as a linear programming problem. The special nature of the problem, however, allows for considerable gains in efficiency by the modification of the usual simplex algorithm for linear programming. These modifications are described in detail by Barrodale and Roberts (1973, 1974).

In many cases, the algorithm can be made faster by computing a least-squares solution prior to the invocation of `RLAV`. This is particularly useful when a least-squares solution has already been computed. The procedure is as follows:

1. Fit the model using least squares and compute the residuals from this fit.
2. Fit the residuals from Step 1 on the regressor variables in the model using `RLAV`.
3. Add the two estimated regression coefficient vectors from Steps 1 and 2. The result is an L_1 solution.

When multiple solutions exist for a given problem, routine `RLAV` may yield different estimates of the regression coefficients on different computers, however, the sum of the absolute values of

the residuals should be the same (within rounding differences). The informational error indicating nonunique solutions may result from rounding accumulation. Conversely, because of rounding the error may fail to result even when the problem does have multiple solutions.

RLLP

Fits a multiple linear regression model using the L_p norm criterion.

Required Arguments

X — NOBS by NCOL matrix containing the data. (Input)

IIND — Independent variable option. (Input)

IIND Meaning

< 0 The first $-IIND$ columns of *X* contain the independent (explanatory) variables.

> 0 The *IIND* independent variables are specified by the column numbers in *INDIND*.

= 0 There are no independent variables.

There are $NCOEF = INTCEP + |IIND|$ regressors—the constant regressor (if $INTCEP = 1$) and the independent variables.

INDIND — Index vector of length *IIND* containing the column numbers of *X* that are the independent (explanatory) variables. (Input, if *IIND* is positive)
If *IIND* is negative, *INDIND* is not referenced and can be a vector of length one.

IRSP — Column number *IRSP* of *X* contains the data for the response (dependent) variable. (Input)

P — The p in the L_p norm. (Input)
 p must be greater than or equal to 1.0. A common choice for p is between 1.0 and 2.0, inclusively.

B — Vector of length *NCOEF* containing an L_p solution for the regression coefficients. (Output)
If $INTCEP = 1$, *B*(1) contains the intercept estimate. *B*($INTCEP + I$) contains the coefficient estimate for the I -th independent variable.

Optional Arguments

NOBS — Number of rows in *X*. (Input)
Default: $NOBS = \text{size}(X,1)$.

NCOL — Number of columns in X . (Input)
Default: $NCOL = \text{size}(X,2)$.

LDX — Leading dimension of X exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDX = \text{size}(X,1)$.

INTCEP — Intercept option. (Input)
Default: $INTCEP = 1$.

INTCEP Action

0 An intercept is not in the model.

1 An intercept is in the model.

IFRQ — Frequency option. (Input)
 $IFRQ = 0$ means that all frequencies are 1.0. For positive $IFRQ$, column number $IFRQ$ of X contains the frequencies.
Default: $IFRQ = 0$.

IWT — Weighting option. (Input)
 $IWT = 0$ means that all weights are 1.0. For positive IWT , column number IWT of X contains the weights.
Default: $IWT = 0$.

TOL — Tolerance used in determining linear dependence. (Input)
 $TOL = 100 * \text{AMACH}(4)$ is a common choice. See documentation for IMSL routine **AMACH** (Reference Material).
Default: $TOL = 1.e-5$ for single precision and $2.d-14$ for double precision.

MAXIT — Maximum number of iterations permitted. (Input)
A common choice is $MAXIT = 100$.
Default: $MAXIT = 100$.

EPS — Convergence criterion. (Input)
If the maximum relative difference in residuals from the k -th to $(k + 1)$ -st iterations is less than EPS , convergence is declared. $EPS = 100 * \text{AMACH}(4)$ is a common choice.
Default: $EPS = 1.e-5$ for single precision and $2.d-14$ for double precision.

R — $NCOEF$ by $NCOEF$ upper triangular matrix containing the R matrix from a QR decomposition of the matrix of regressors. (Output)

LDR — Leading dimension of R exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDR = \text{size}(R,1)$.

IRANK — Rank of the matrix of regressors. (Output)

If **IRANK** is less than **NCOEF**, linear dependence of the regressors is declared.

DFE — Sum of the frequencies minus **IRANK**. (Output) In a least squares fit ($p = 2$), **DFE** is called the degrees of freedom for error.

E — Vector of length **NOBS** containing the residuals. (Output)

SCALE2 — Square of the scale constant used in an L_p analysis. (Output)

An estimated asymptotic variance-covariance matrix of the regression coefficients is $SCALE2 * (R^T R)^{-1}$.

ELP — L_p norm of the residuals. (Output)

ITER — Number of iterations performed. (Output)

NRMISS — Number of rows of data that contain any missing values for the independent, dependent, weight, or frequency variables. (Output)

NaN (not a number) is used as the missing value code. Any row of **x** containing NaN as a value of the independent, dependent, weight, or frequency variables is omitted from the analysis.

FORTRAN 90 Interface

Generic: CALL RLLP (X, IIND, INDIND, IRSP, P, B [,...])

Specific: The specific interface names are S_RLLP and D_RLLP.

FORTRAN 77 Interface

Single: CALL RLLP (NOBS, NCOL, X, LDX, INTCEP, IIND, INDIND, IRSP,
 IFRQ, IWT, P, TOL, MAXIT, EPS, B, R, LDR, IRANK, DFE, E,
 SCALE2, ELP, ITER, NRMISS)

Double: The double precision name is DRLLP.

Example

Different straight line fits to a data set are computed under the criterion of minimizing the L_p norm by using p equal to 1, 1.5, 2.0 and 2.5.

```
USE RLLP_INT
USE UMACH_INT
USE WRRRL_INT
USE WRRRN_INT
INTEGER    INTCEP, LDR, LDX, NCOEF, NCOL, NIND, NOBS
PARAMETER (INTCEP=1, NCOL=2, NIND=1, NOBS=8, LDX=NOBS, &
          NCOEF=INTCEP+NIND, LDR=NCOEF)
```

!

```

INTEGER      IIND, INDIND(NIND), IRANK, IRSP, ITER, NOUT, NRMISS
REAL         B(NCOEF), DFE, E(NOBS), ELP, EPS, P, &
             R(LDR,NCOEF), SCALE2, X(LDX,NCOL)
CHARACTER    CLABEL(1)*4, RLABEL(1)*12
!
DATA (X(1,J),J=1,NCOL)/1.0, 1.0/
DATA (X(2,J),J=1,NCOL)/4.0, 5.0/
DATA (X(3,J),J=1,NCOL)/2.0, 0.0/
DATA (X(4,J),J=1,NCOL)/2.0, 2.0/
DATA (X(5,J),J=1,NCOL)/3.0, 1.5/
DATA (X(6,J),J=1,NCOL)/3.0, 2.5/
DATA (X(7,J),J=1,NCOL)/4.0, 2.0/
DATA (X(8,J),J=1,NCOL)/5.0, 3.0/
!
CALL UMACH (2, NOUT)
IIND       = NIND
INDIND(1) = 1
IRSP       = 2
EPS        = 0.001
!
DO 10 P=1.0, 2.5, 0.5
  CALL RLLP (X, IIND, INDIND, IRSP, P, B, EPS=EPS, R=R, &
            IRANK=IRANK, DFE=DFE, E=E, SCALE2=SCALE2, ELP=ELP, &
            ITER=ITER, NRMISS=NRMISS)
!
WRITE (NOUT,99997)
RLABEL(1) = 'Coefficients'
CLABEL(1) = 'NONE'
CALL WRRRL ('%/', B, RLABEL, CLABEL, 1, NCOEF, 1, FMT='(F6.2)')
RLABEL(1) = 'Residuals'
CLABEL(1) = 'NONE'
CALL WRRRL ('%/', E, RLABEL, CLABEL, 1, NOBS, 1, FMT='(F6.2)')
WRITE (NOUT,*)
WRITE (NOUT,99998) 'p', P
WRITE (NOUT,99998) 'Lp norm of the residuals', ELP
WRITE (NOUT,99999) 'Rank of the matrix of regressors', IRANK
WRITE (NOUT,99998) 'Degrees of freedom error', DFE
WRITE (NOUT,99999) 'Number of iterations', ITER
WRITE (NOUT,99999) 'Number of missing values', NRMISS
WRITE (NOUT,99998) 'Square of the scale constant', SCALE2
CALL WRRRN ('R matrix', R)
10 CONTINUE
99997 FORMAT (/1X, 72('-'))
99998 FORMAT (1X, A, T34, F5.2)
99999 FORMAT (1X, A, T34, I5)
END

```

Output

```

-----
Coefficients      0.50      0.50
Residuals        0.00      2.50     -1.50      0.50     -0.50      0.50     -0.50      0.00

p                                     1.00
Lp norm of the residuals                6.00
Rank of the matrix of regressors        2

```

Degrees of freedom error 6.00
Number of iterations 8
Number of missing values 0
Square of the scale constant 6.25

R matrix
1 2
1 2.828 8.485
2 0.000 3.464

Coefficients 0.39 0.55

Residuals 0.06 2.39 -1.50 0.50 -0.55 0.45 -0.61 -0.16

p 1.50
Lp norm of the residuals 3.71
Rank of the matrix of regressors 2
Degrees of freedom error 6.00
Number of iterations 6
Number of missing values 0
Square of the scale constant 1.06

R matrix
1 2
1 2.828 8.485
2 0.000 3.464

Coefficients -0.12 0.75

Residuals 0.38 2.12 -1.38 0.62 -0.62 0.38 -0.88 -0.62

p 2.00
Lp norm of the residuals 2.94
Rank of the matrix of regressors 2
Degrees of freedom error 6.00
Number of iterations 1
Number of missing values 0
Square of the scale constant 1.44

R matrix
1 2
1 2.828 8.485
2 0.000 3.464

Coefficients -0.44 0.87

Residuals 0.57 1.96 -1.30 0.70 -0.67 0.33 -1.04 -0.91

p 2.50
Lp norm of the residuals 2.54
Rank of the matrix of regressors 2
Degrees of freedom error 6.00

```

Number of iterations          4
Number of missing values     0
Square of the scale constant  0.79

```

```

R matrix
      1      2
1  2.828  8.485
2  0.000  3.464

```

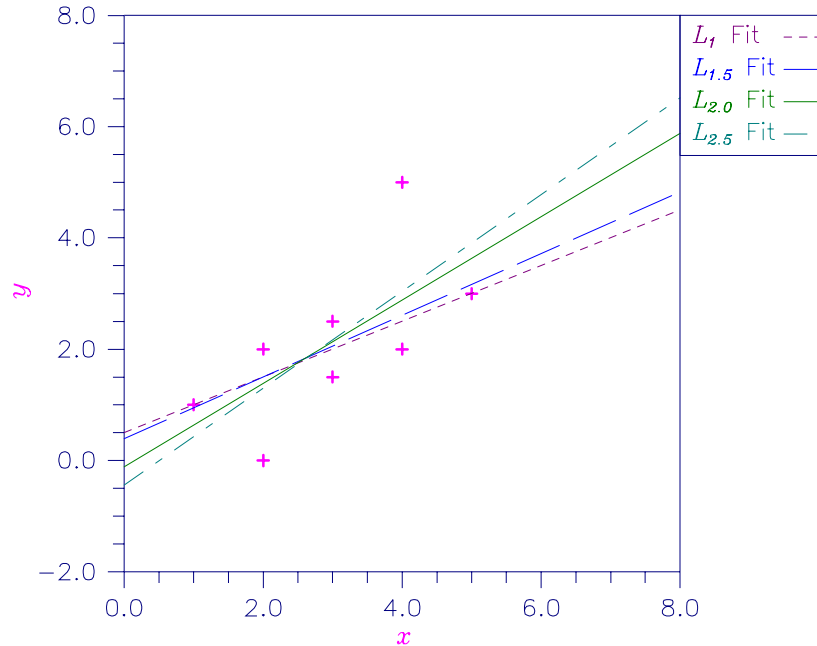


Figure 2-12 Various L_p Fitted Lines

Comments

1. Workspace may be explicitly provided, if desired, by use of R2LP/DR2LP. The reference is:

```

CALL R2LP (NROW, NCOL, X, LDX, INTCEP, IIND, INDIND, IRSP, IFRQ,
IWT, P, TOL, MAXIT, EPS, B, R, LDR, IRANK, DFE, E, SCALE2, ELP,
ITER, NRMISS, IWK, WK)

```

The additional arguments are as follows:

IWK — Work array of length $\text{NOBS} + |\text{IIND}| + 3$.

WK — Work array of length $2 * \text{NOBS} + 8 * \text{NCOEF} + 4$.

2. Informational errors
Type Code

- | | | |
|---|---|---|
| 4 | 1 | A negative weight was encountered. |
| 4 | 2 | A negative frequency was encountered. |
| 4 | 3 | The p -th power of the absolute value of one or more of the current residuals will result in overflow or underflow in subsequent computations. A solution cannot be computed because of a serious loss of accuracy. For large p , consider the use of IMSL routine RLMV, which uses the L_∞ (minimax) criterion. |
| 3 | 4 | Convergence has not been achieved after MAXIT iterations. MAXIT or EPS may be too small. Try increasing MAXIT or EPS. |
| 3 | 5 | Convergence is not declared. The line-search procedure failed to find an acceptable solution after 10 successive attempts. EPS may be too small. Try increasing its value. |

Description

Routine RLLP computes estimates of the regression coefficients in a multiple linear regression model $y = X\beta + \varepsilon$ under the criterion of minimizing the L_p norm of the deviations for $i = 1, \dots, n$ of the observed response y_i from the fitted response

$$\hat{y}_i$$

for a set on n observations and for $p \geq 1$. For the case IWT = 0 and IFRQ = 0 the estimated regression coefficient vector,

$$\hat{\beta}$$

(output in B) minimizes the L_p norm

$$\left(\sum_{i=1}^n |y_i - \hat{y}_i|^p \right)^{1/p}$$

The choice $p = 1$ yields the maximum likelihood estimate for β when the errors have a Laplace distribution. The choice $p = 2$ is best for errors that are normally distributed. Sposito (1989, pages 36–40) discusses other reasonable alternatives for p based on the sample kurtosis of the errors.

Weights are useful if the errors in the model have known unequal variances

$$\sigma_i^2$$

In this case, the weights should be taken as

$$w_i = 1/\sigma_i^2$$

Frequencies are useful if there are repetitions of some observations in the data set. If a single row of data corresponds to n_i observations, set the frequency $f_i = n_i$. In general, RLLP minimizes the L_p norm

$$\left(\sum_{i=1}^n f_i \left| \sqrt{w_i} (y_i - \hat{y}_i) \right|^p \right)^{1/p}$$

The asymptotic variance-covariance matrix of the estimated regression coefficients is given by

$$\text{asy. var}(\hat{\beta}) = \lambda^2 (R^T R)^{-1}$$

where R is from the QR decomposition of the matrix of regressors (output in R) and where an estimate of λ^2 is output in `SCALE2`. An estimated asymptotic variance-covariance matrix of the estimated regression coefficients can be computed following the call to `RLLP` by invoking routine `RCOVB` (page 164) with R and `SCALE2`.

In the discussion that follows, we will first present the algorithm with frequencies and weights all taken to be one. Later, we will present the modifications to handle frequencies and weights different from one.

Routine `RLLP` uses Newton's method with a line search for $p > 1.25$ and, for $p \leq 1.25$, uses a modification due to Ekblom (1973, 1987) in which a series of perturbed problems are solved in order to guarantee convergence and increase the convergence rate. The cutoff value of 1.25 as well as some of the other implementation details given in the remaining discussion were investigated by Sallas (1990) for their effect on `CPU` times.

In each case, for the first iteration a least-squares solution for the regression coefficients is computed using routine `RGIVN` (page 117). If $p = 2$, the computations are finished. Otherwise, the residuals from the k -th iteration,

$$e_i^{(k)} = y_i - \hat{y}_i^{(k)}$$

are used to compute the gradient and Hessian for the Newton step for the $(k + 1)$ -st iteration for minimizing the p -th power of the L_p norm. (The exponent $1/p$ in the L_p norm can be omitted during the iterations.)

For subsequent iterations, we first discuss the $p > 1.25$ case. For $p > 1.25$, the gradient and Hessian at the $(k + 1)$ -st iteration depend upon

$$z_i^{(k+1)} = |e_i^{(k)}|^{p-1} \text{sign}(e_i^{(k)})$$

and

$$v_i^{(k+1)} = |e_i^{(k)}|^{p-2}$$

In the case $1.25 < p < 2$ and

$$e_i^{(k)} = 0, v_i^{(k+1)}$$

and the Hessian are undefined; and we follow the recommendation of Merle and Spath (1974). Specifically, we modify the definition of

$$v_i^{(k+1)}$$

to the following:

$$v_i^{(k+1)} = \begin{cases} \tau^{p-2} & \text{if } p < 2 \text{ and } |e_i^{(k)}| < \tau \\ |e_i^{(k)}|^{p-2} & \text{otherwise} \end{cases}$$

where τ equals $100 * \text{AMACH}(4)$ times the square root of the residual mean square from the least-squares fit. (See routine `AMACH`, which is documented in the section “Machine-Dependent Constants” in Reference Material.)

Let $V^{(k+1)}$ be a diagonal matrix with diagonal entries

$$v_i^{(k+1)}$$

and let $z^{(k+1)}$ be a vector with elements

$$z_i^{(k+1)}$$

In order to compute the step on the $(k + 1)$ -st iteration, the R from the QR decomposition of $[V^{(k+1)}]^{1/2}X$ is computed using fast Givens transformations. Let $R^{(k+1)}$ denote the upper triangular matrix from the QR decomposition. Routine `GIRTS` (see Chapter 20, Mathematical Support) is used to solve the linear system $[R^{(k+1)}]^T R^{(k+1)} d^{(k+1)} = X^T z^{(k+1)}$ is solved for $d^{(k+1)}$ where $R^{(k+1)}$ is from the QR decomposition of $[V^{(k+1)}]^{1/2}X$. The step taken on the $(k + 1)$ -st iteration is

$$\hat{\beta}^{(k+1)} = \hat{\beta}^{(k)} + \alpha^{(k+1)} \frac{1}{p-1} d^{(k+1)}$$

The first attempted step on the $(k + 1)$ -st iteration is with $\alpha^{(k+1)} = 1$. If all of the

$$e_i^{(k)}$$

are nonzero, this is exactly the Newton step. See Kennedy and Gentle (1980, pages 528–529) for further discussion.

If the first attempted step does not lead to a decrease of at least one-tenth of the predicted decrease in the p -th power of the L_p norm of the residuals, a backtracking linesearch procedure is used. The backtracking procedure uses a one-dimensional quadratic model to estimate the backtrack constant p . The value of p is constrained to be no less than 0.1. An approximate upper bound for p is 0.5. If after 10 successive backtrack attempts, $\alpha^{(k)} = \rho_1 \rho_2 \dots \rho_{10}$ does not produce a step with a sufficient decrease, then `RLLP` issues a message with error code 5. For further details on the backtrack line-search procedure, see Dennis and Schnabel (1983, pages 126–127).

Convergence is declared when the maximum relative change in the residuals from one iteration to the next is less than or equal to `EPS`. The relative change

$$\delta_i^{(k+1)}$$

in the i -th residual from iteration k to iteration $k + 1$ is computed as follows:

$$\delta_i^{(k+1)} = \begin{cases} 0 & \text{if } e_i^{(k+1)} = e_i^{(k)} = 0 \\ \left| \frac{e_i^{(k+1)} - e_i^{(k)}}{\max(|e_i^{(k)}|, |e_i^{(k+1)}|, s)} \right| & \text{otherwise} \end{cases}$$

where s is the square root of the residual mean square from the least-squares fit on the first iteration.

For the case $1 \leq p \leq 1.25$, we describe the modifications to the previous procedure that incorporate Ekblom's (1973) results. A sequence of perturbed problems are solved with a successively smaller perturbation constant c . On the first iteration, the least-squares problem is solved. This corresponds to an infinite c . For the second problem, c is taken equal to s , the square root of the residual mean square from the least-squares fit. Then, for the $(j + 1)$ -st problem, the value of c is computed from the previous value of c according to

$$c_{j+1} = c_j / 10^{5p-4}$$

Each problem is stated as

$$\text{Minimize } \sum_{i=1}^n (e_i^2 + c^2)^{p/2}$$

For each problem, the gradient and Hessian on the $(k + 1)$ -st iteration depend upon

$$z_i^{(k+1)} = e_i^{(k)} r_i^{(k)}$$

and

$$v_i^{(k+1)} = \left[1 + \frac{(p-2)(e_i^{(k)})^2}{(e_i^{(k)})^2 + c^2} \right] r_i^{(k)}$$

where

$$r_i^{(k)} = [(e_i^{(k)})^2 + c^2]^{(p-2)/2}$$

The linear system $[R^{(k+1)}]^T R^{(k+1)} d^{(k+1)} = X^T z^{(k+1)}$ is solved for $d^{(k+1)}$ where $R^{(k+1)}$ is from the QR decomposition of $[V^{(k+1)}]^{1/2} X$. The step taken on the $(k + 1)$ -st iteration is

$$\hat{\beta}^{(k+1)} = \hat{\beta}^{(k)} + \alpha^{(k+1)} d^{(k+1)}$$

where the first attempted step is with $\alpha^{(k+1)} = 1$. If necessary, the backtracking line-search procedure discussed earlier is used.

Convergence for each problem is relaxed somewhat by using a convergence epsilon equal to $\max(\text{EPS}, 10^{-j})$ where $j = 1, 2, 3, \dots$ indexes the problems ($j = 0$ corresponds to the least-squares problem).

After the convergence of a problem for a particular c , Ekblom's (1987) extrapolation technique is used to compute the initial estimate of β for the new problem. Let $R^{(k)}$,

$$v_i^{(k)}, e_i^{(k)}$$

and c be from the last iteration of the last problem. Let

$$t_i = \frac{(p-2)v_i^{(k)}}{(e_i^{(k)})^2 + c^2}$$

and let t be the vector with elements t_i . The initial estimate of β for the new problem with perturbation constant $0.01c$ is

$$\hat{\beta}^{(0)} = \hat{\beta}^{(k)} + \Delta cd$$

where $\Delta c = (0.01c - c) = -0.99c$, and where d is the solution of the linear system $[R^{(k)}]^T R^{(k)} d = X^T t$.

Convergence of the sequence of problems is declared when the maximum relative difference in residuals from the solution of successive problems is less than `EPS`.

The preceding discussion was limited to the case for which `IWT` = 0 and `IFRQ` = 0, i.e., the weights and frequencies are all taken equal to one. The necessary modifications to the preceding algorithm to handle weights and frequencies not all equal to one are as follows:

1. Replace

$$e_i^{(k)} \text{ by } \sqrt{w_i} e_i^{(k)}$$

in the definitions of

$$z_i^{(k+1)}, v_i^{(k+1)}, \delta_i^{(k+1)}$$

and t_i .

2. Replace

$$z_i^{(k+1)} \text{ by } f_i \sqrt{w_i} z_i^{(k+1)}, v_i^{(k+1)} \text{ by } f_i w_i v_i^{(k+1)}, \text{ and } t_i^{(k+1)} \text{ by } f_i \sqrt{w_i} t_i^{(k+1)}$$

These replacements have the same effect as multiplying the i -th row of X and y by

$$\sqrt{w_i}$$

and repeating the row f_i times except for the fact that the residuals returned by `RLLP` are in terms of the original y and X .

Finally, R and an estimate of λ^2 are computed. Actually, R is recomputed because on output it corresponds to the R from the initial QR decomposition for least squares. The formula for the estimate of λ^- depends on p .

For $p = 1$, the estimator for λ^2 is given by (McKean and Schrader 1987)

$$\hat{\lambda}^2 = \left[\frac{\sqrt{\text{DFE}} (\tilde{e}_{(\text{DFE}-k+1)} - \tilde{e}_{(k)})}{2z_{0.975}} \right]^2$$

with

$$k = \frac{\text{DFE} + k}{2} - z_{0.975} \sqrt{\frac{\text{DFE}}{4}}$$

where $z_{0.975}$ is the 97.5 percentile of the standard normal distribution, and where

$$\tilde{e}_{(m)} (m = 1, 2, \dots, \text{DFE})$$

are the ordered residuals where `IRANK` zero residuals are excluded. (Note that

$$\text{DFE} = \sum_{i=1}^n f_i - \text{IRANK}$$

For $p = 2$, the estimator of λ^2 is the customary least-squares estimator given by

$$s^2 = \frac{\text{SSE}}{\text{DFE}} = \frac{\sum_{i=1}^n f_i w_i (y_i - \hat{y}_i)^2}{\sum_{i=1}^n f_i - \text{IRANK}}$$

For $1 < p < 2$ and for $p > 2$, the estimator for λ^2 is given by (Gonin and Money 1989)

$$\hat{\omega}_p^2 = \frac{m_{2p-2}}{[(p-1)m_{p-2}]^2}$$

with

$$m_r = \frac{\sum_{i=1}^n f_i |\sqrt{w_i} (y_i - \hat{y}_i)|^r}{\sum_{i=1}^n f_i}$$

RLMV

Fits a multiple linear regression model using the minimax criterion.

Required Arguments

X — NOBS by NCOL matrix containing the data. (Input)

IIND — Independent variable option. (Input)

The absolute value of IIND is the number of independent (explanatory) variables. The sign of IIND specifies the following options :

IIND Meaning

- < 0 The data for the -IIND independent variables are given in the first -IIND columns of X.
- > 0 The data for the IIND independent variables are in the columns of X whose column numbers are given by the elements of INDIND.
- = 0 There are no independent variables.

The regressors are the constant regressor (if INTCEP = 1) and the independent variables.

INDIND — Index vector of length IIND containing the column numbers of X that are the independent (explanatory) variables. (Input, if IIND is positive)
If IIND is negative, INDIND is not referenced and can be a vector of length one.

IRSP — Column number `IRSP` of `X` contains the data for the response (dependent) variable. (Input)

B — Vector of length `INTCEP + |IIND|` containing a minimax solution for the regression coefficients. (Output)

If `INTCEP = 1`, `B(1)` contains the intercept estimate. `B(INTCEP + I)` contains the coefficient estimate for the `I`-th independent variable.

Optional Arguments

NOBS — Number of observations. (Input)

Default: `NOBS = size(X,1)`.

NCOL — Number of columns in `X`. (Input)

Default: `NCOL = size(X,2)`.

LDX — Leading dimension of `X` exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDX = size(X,1)`.

INTCEP — Intercept option. (Input)

Default: `INTCEP = 1`.

INTCEP	Action
---------------	---------------

0	An intercept is not in the model.
---	-----------------------------------

1	An intercept is in the model.
---	-------------------------------

IRANK — Rank of the matrix of regressors. (Output)

If `IRANK` is less than `INTCEP + |IIND|`, linear dependence of the regressors was declared.

AEMAX — Magnitude of the largest residual. (Output)

ITER — Number of iterations performed. (Output)

NRMIS — Number of rows of data containing NaN (not a number) for the dependent or independent variables. (Output)

If a row of data contains NaN for any of these variables, that row is excluded from the computations.

FORTRAN 90 Interface

Generic: `CALL RLMV (X, IIND, INDIND, IRSP, B [,...])`

Specific: The specific interface names are `S_RLMV` and `D_RLMV`.

FORTRAN 77 Interface

Single: CALL RLMV (NOBS, NCOL, X, LDX, INTCEP, IIND, INDIND, IRSP,
 B, IRANK, AEMAX, ITER, NRMISS)

Double: The double precision name is DRLMV.

Example

A straight line fit to a data set is computed under the LMV criterion.

```
!                                     SPECIFICATIONS FOR PARAMETERS
      USE RLMV_INT
      USE UMACH_INT
      USE WRRRL_INT
      INTEGER    LDX, NCOEF, NCOL, NOBS
      PARAMETER (NCOEF=2, NCOL=2, NOBS=7, LDX=NOBS)
!
      INTEGER    IIND, INDIND(1), IRANK, IRSP, ITER, NOUT, &
                NRMISS
      REAL       B(NCOEF), AEMAX, X(LDX,NCOL)
      CHARACTER  CLABEL(1)*4, RLABEL(1)*4
!
      DATA (X(1,J),J=1,NCOL)/0.0, 0.0/
      DATA (X(2,J),J=1,NCOL)/1.0, 2.5/
      DATA (X(3,J),J=1,NCOL)/2.0, 2.5/
      DATA (X(4,J),J=1,NCOL)/3.0, 4.5/
      DATA (X(5,J),J=1,NCOL)/4.0, 4.5/
      DATA (X(6,J),J=1,NCOL)/4.0, 6.0/
      DATA (X(7,J),J=1,NCOL)/5.0, 5.0/
!
      IIND    = -1
      IRSP    = 2
!
      CALL RLMV (X, IIND, INDIND, IRSP, B, IRANK=IRANK, AEMAX=AEMAX, &
                ITER=ITER, NRMISS=NRMISS)
!
      CALL UMACH (2, NOUT)
      RLABEL(1) = 'B ='
      CLABEL(1) = 'NONE'
      CALL WRRRL (' ', B, RLABEL, CLABEL, 1, NCOEF, 1, FMT='(F6.2)')
      WRITE (NOUT,*) 'IRANK = ', IRANK
      WRITE (NOUT,*) 'AEMAX = ', AEMAX
      WRITE (NOUT,*) 'ITER = ', ITER
      WRITE (NOUT,*) 'NRMISS = ', NRMISS
      END
```

Output

```
B =       1.00       1.00
IRANK =    2
AEMAX =       1.00000
ITER =     3
NRMISS =    0
```

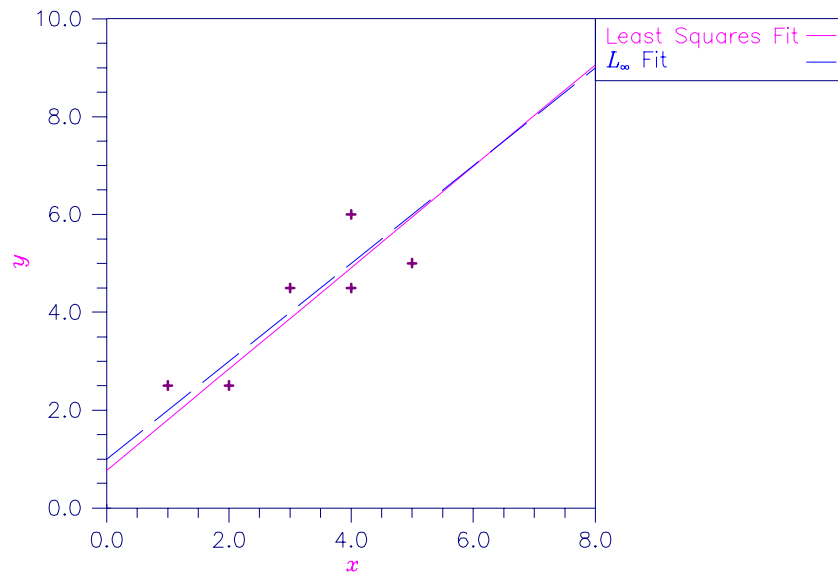


Figure 2-13 Least Squares and Least Maximum Value Fitted Lines

Comments

1. Workspace may be explicitly provided, if desired, by use of R2MV/DR2MV. The reference is:

```
CALL R2MV (NOBS, NCOL, X, LDX, INTCEP, IIND, INDIND, IRSP, B,
IRANK, AEMAX, ITER, WK)
```

The additional argument is:

WK — Workspace of length $\text{NOBS} * (|\text{IIND}| + 5) + 2 * |\text{IIND}| + 3$.

2. Informational errors
Type Code

3	1	The solution may not be unique.
4	1	Calculations terminated prematurely due to rounding.

3. If X is not needed, LDX=NOBS, and IIND < 0, then X and the first NOBS * (-IIND + 1) elements of WK may occupy the same storage locations. The reference would be:

```
CALL R2MV (NOBS, NCOL, WK, NOBS, INTCEP, IIND, INDIND, IRSP, B,
IRANK, AEMAX, ITER, WK)
```

Description

Routine `RLMV` computes estimates of the regression coefficients in a multiple linear regression model. The criterion satisfied is the minimization of the maximum deviation of the observed response y_i from the fitted response \hat{y}_i for a set of n observations. Under this criterion, known as the minimax or LMV (least maximum value) criterion, the regression coefficient estimates minimize $\max_{1 \leq i \leq n} |y_i - \hat{y}_i|$. The estimation problem can be posed as a linear programming problem.

A dual simplex algorithm is appropriate, however, the special nature of the problem allows for considerable gains in efficiency by modification of the dual simplex iterations so as to move more rapidly toward the optimal solution. The modifications are described in detail by Barrodale and Phillips (1975).

When multiple solutions exist for a given problem, `RLMV` may yield different estimates of the regression coefficients on different computers, however, the largest residual in absolute value should have the same absolute value (within rounding differences). The informational error indicating nonunique solutions may result from rounding accumulation. Conversely, because of rounding, the error may fail to result even when the problem does have multiple solutions.

Chapter 3: Correlation

Routines

3.1. The Correlation Matrix		
Correlation	CORVC	340
Pooled covariance matrix	COVPL	348
Partial correlations	PCORR	353
Robust estimate of correlation matrix	RBCOV	358
3.2. Correlation Measures for a Contingency Table		
The $r \times c$ contingency table.....	CTRHO	367
Tetrachoric correlation (2×2 tables).....	TETCC	370
3.3. A Dichotomous Variable with a Classification Variable		
Specified values for the classification variable	BSPBS	374
Computed values for the classification variable	BSCAT	377
3.4. Measures Based Upon Ranks		
Kendall coefficient of concordance	CNCRD	379
Kendall's τ	KENDL	383
Exact frequencies for Kendall's τ	KENDP	387

Usage Notes

This chapter is concerned with measures of correlation for bivariate data. The usual multivariate measures of correlation and covariance for continuous random variables are produced by routine [CORVC \(page 340\)](#). For data grouped by some auxiliary variable, routine [COVPL \(page 348\)](#) can be used to compute the pooled covariance matrix along with the means for each group. Partial correlations or covariances, given the correlation or covariance matrix computed from [CORVC](#) or [COVPL](#), are computed by [PCORR \(page 353\)](#). Routine [RBCOV \(page 358\)](#) computes robust estimates of the covariance matrix and mean vector. If data are grouped by some auxiliary variable, routine [RBCOV](#) can also be used to estimate the pooled covariance matrix and means for each group. The remaining routines are concerned with rank and/or discrete data. General references for these routines are Conover (1980) or Gibbons (1971).

[CTRHO \(page 367\)](#) and [TETCC \(page 370\)](#) produce measures of correlation for contingency tables. In [CTRHO](#), the inverse normal scores obtained from the row and column marginal distributions are assumed known, and the correlation coefficient is estimated by assuming bivariate normality. In

TETCC, a 2×2 table is produced from continuous input data using estimates for the sample medians. The correlation coefficient is estimated from the resulting 2×2 table.

If one of the variables is dichotomous while the second variable can be ranked, the routines `BSPBS` (page 374) or `BSCAT` (page 377) can be used. The difference between these routines is in whether the class values for the ranked variable are given by the user (`BSPBS`) or are estimated as inverse normal scores from the marginal cumulative distribution (`BSCAT`). Routine `CNCRD` (page 379) computes Kendall's coefficient of concordance, and routine `KENDL` (page 383) computes Kendall's rank correlation coefficient τ . Probabilities for τ are computed by routine `KENDP` (page 387).

Other Routines

Other IMSL routines compute measures of correlation or association and may be of interest. Routine `CTTWO` described in Chapter 5, "Categorical and Discrete Data Analysis," computes measures of association for the 2×2 contingency table. Routine `CTCHI` (page 491), in the same chapter, computes measures of association for the general $r \times c$ contingency table. Routine `CDIST` in Chapter 11, "Cluster Analysis," computes measures of similarity and dissimilarity, including the correlation coefficient. Measures of multivariate association or correlation are computed in Chapter 2, "Regression," and in "Independence of Sets of Variables and Canonical Correlation Analysis."

CORVC

Computes the variance-covariance or correlation matrix.

Required Arguments

NVAR — Number of variables. (Input)

The weight or frequency variables, if used, are not counted in `NVAR`.

X — $|\text{NROW}|$ by `NVAR` + m matrix containing the data, where m is 0, 1, or 2 depending on whether any column(s) of `X` correspond to weights and/or frequencies. (Input)

COV — `NVAR` by `NVAR` matrix containing either the correlation matrix (possibly with the standard deviations on the diagonal), the variance-covariance matrix, or the corrected sums of squares and crossproducts matrix, as controlled by the `COV` option, `ICOPT`. (Output, if `IDO` = 0 or 1; input/output, if `IDO` = 2 or 3)

The elements of `COV` correspond to the columns of `X`, except for the columns of `X` containing weights or frequencies (see `XMEAN`).

Optional Arguments

IDO — Processing option. (Input)

Default: `IDO` = 0.

IDO Action

- 0 This is the only invocation of `CORVC` for this data set, and all the data are input at once.
- 1 This is the first invocation, and additional calls to `CORVC` will be made. Initialization and updating for the `NROW` observations are performed. The means (in `XMEAN`) are output correctly, but the quantities output in `COV` are intermediate results.
- 2 This is an intermediate invocation of `CORVC`, and updating for the `NROW` observations is performed.
- 3 This is the final invocation of this routine. If `NROW` is not zero, updating is performed. The wrap-up computations for `COV` are performed.

It is possible to call `CORVC` twice in succession with `IDO = 3` in order to first compute covariances (`ICOPT = 1`) and then compute correlations (`ICOPT = 2` or `3`). This ability is most important when pairwise deletion of missing values is used (`MOPT = 3`). The workspace arrays (or the workspace) must not be altered in between calls.

NROW — The absolute value of `NROW` is the number of rows of data currently input in `X`. (Input)

Default: `NROW = size (X,1)`.

`NROW` may be positive, zero, or negative. Negative `NROW` means that the `-NROW` rows of data are to be deleted from (most aspects of) the analysis. This should be done only if `IDO` is 2 or 3 and the wrap-up computations for `COV` have not been performed. When a negative value is input for `NROW`, it is assumed that each of the `-NROW` rows of `X` has been input (with positive `NROW`) in previous invocations of `CORVC`. Use of negative values of `NROW` should be made with care since it is possible that a constant variable in the remaining data will not be recognized as such.

LDX — Leading dimension of `X` exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDX = size (X,1)`.

IFRQ — Frequency option. (Input)

`IFRQ = 0` means that all frequencies are 1.0. For positive `IFRQ`, column `IFRQ` of `X` contains the frequencies.*f*

Default: `IFRQ = 0`.

IWT — Weighting option. (Input)

`IWT = 0` means that all weights are 1.0. For positive `IWT`, column `IWT` of `X` contains the weights. Observations with zero weight are counted as observations in the frequencies, but do not contribute to the means, variances, covariances, or correlations.

Observations with negative weights are missing.

Default: `IWT = 0`.

MOPT — Missing value option. (Input)

NaN (not a number) is interpreted as the missing value code, and any value in x equal to NaN is excluded from the computations. If **MOPT** is positive, various pairwise exclusion methods are used. See routine *AMACH/DMACH* (Reference Material).

Default: **MOPT** = 0.

MOPT Action

- 0 The exclusion is listwise. (The entire row of x is excluded if any of the values of the row is equal to the missing value code.)
- 1 Raw crossproducts are computed from all valid pairs and means, and variances are computed from all valid data on the individual variables. Corrected crossproducts, covariances and correlations are computed using these quantities.
- 2 Raw crossproducts, means and variances are computed as in the case of **MOPT** = 1. However, corrected crossproducts and covariances are computed only from the valid pairs of data. Correlations are computed using these covariances and the variances from all valid data.
- 3 Raw crossproducts, means, variances, and covariances are computed as in the case of **MOPT** = 2. Correlations are computed using these covariances, but the variances used are computed only from the valid pairs of data.

ICOPT — COV option. (Input)

Default: **ICOPT** = 0.

ICOPT Action

- 0 COV contains the variance-covariance matrix.
- 1 COV contains the corrected sums of squares and crossproducts matrix.
- 2 COV contains the correlation matrix.
- 3 COV contains the correlation matrix, except for the diagonal elements, which are the standard deviations.

XMEAN — Vector of length *NVAR* containing the variable means. (Output, if **IDO** = 0 or 1; input/output, if **IDO** = 2 or 3)

The elements of **XMEAN** correspond to the columns of x , except that if weights and/or frequencies are used, the elements of **XMEAN** beyond the **IWT** or **IFRQ** element are shifted relative to the columns of x .

LDCOV — Leading dimension of COV exactly as specified in the dimension statement in the calling program. (Input)

Default: **LDCOV** = size (COV,1).

INCD — Incidence matrix. (Output, if `IDO = 0` or `1`; input/output, if `IDO = 2` or `3`)
If `MOPT` is zero, `INCD` is 1 by 1, and contains the number of valid observations. If `MOPT` is positive, `INCD` is `NVAR` by `NVAR` and contains the numbers of pairs of valid observations that are used in calculating the crossproducts for `COV`.

LDINCD — Leading dimension of `INCD` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDINCD = size(INCD,1)`.

NOBS — Total number of observations (that is, the total of the frequencies). (Output, if `IDO = 0` or `1`; input/output, if `IDO = 2` or `3`)
If `MOPT = 0`, observations with missing values are not included in `NOBS`. For other values of `MOPT`, all observations are included except for observations with missing values for the weight or the frequency.

NMISS — Total number of observations that contain any missing values. (Output, if `IDO = 0` or `1`; input/output, if `IDO = 2` or `3`)

SUMWT — Sum of the weights of all observations that are processed. (Output, if `IDO = 0`, or `1`; input/output, if `IDO = 2` or `3`)
If `MOPT = 0`, observations with missing values are not included in `SUMWT`. For other values of `MOPT`, all observations are included except for observations with missing values for the weight or the frequency.

FORTRAN 90 Interface

Generic: `CALL CORVC (NVAR, X, COV [,...])`

Specific: The specific interface names are `S_CORVC` and `D_CORVC`.

FORTRAN 77 Interface

Single: `CALL CORVC (IDO, NROW, NVAR, X, LDX, IFRQ, IWT, MOPT, ICOPT, XMEAN, COV, LDCOV, INCD, LDINCD, NOBS, NMISS, SUMWT)`

Double: The double precision name is `DCORVC`.

Example 1

The first example illustrates the use of `CORVC` when inputting all of the data at once. The first 50 observations in the Fisher iris data (see routine `GDATA`, Chapter 19, Utilities) are used. Note in this example that the first variable is constant over the first 50 observations.

```
USE GDATA_INT
USE UMACH_INT
USE CORVC_INT
USE WRRRN_INT
USE WRIRN_INT
```

```

INTEGER   LDICOV, LDINCD, LDX, NVAR
PARAMETER (LDICOV=5, LDINCD=1, LDX=150, NVAR=5)
!
INTEGER   INCD(LDINCD,1), NMISS, NOBS, NOUT, NROW, NV
REAL      COV(LDICOV,NVAR), SUMWT, X(LDX,NVAR), XMEAN(NVAR)
!
CALL GDATA (3, X, NOBS, NV)
!
CALL UMACH (2, NOUT)
NROW = 50
!
CALL CORVC (NVAR, X, COV, NROW=NROW, XMEAN=XMEAN, INCD=INCD, &
            NOBS=NOBS, NMISS=NMISS, SUMWT=SUMWT)
!
CALL WRRRN ('XMEAN', XMEAN, 1, NVAR, 1, 0)
CALL WRRRN ('COV', COV)
CALL WRIRN ('INCD', INCD)
WRITE (NOUT,*) ' NOBS = ', NOBS, ' NMISS = ', NMISS, ' SUMWT = ', &
            SUMWT
END

```

Output

		XMEAN				
		1	2	3	4	5
1.000		5.006	3.428	1.462	0.246	

		COV				
		1	2	3	4	5
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.1242	0.0992	0.0164	0.0103	
3	0.0000	0.0992	0.1437	0.0117	0.0093	
4	0.0000	0.0164	0.0117	0.0302	0.0061	
5	0.0000	0.0103	0.0093	0.0061	0.0111	

```

INCD
50
NOBS = 50 NMISS = 0 SUMWT = 50.0000

```

Comments

1. Workspace may be explicitly provided, if desired, by use of C2RVC/DC2RVC. The reference is:

```

CALL C2RVC (IDO, NROW, NVAR, X, LDX, IFRQ, IWT, MOPT, ICOPT,
XMEAN, COV, LDICOV, INCD, LDINCD, NOBS, NMISS, SUMWT, WK)

```

The additional argument is:

WK — Workspace of the length specified above. **WK** should not be changed between calls to C2RVC.

The workspace may contain statistics of interest. Let

$$m = \text{NVAR}$$

$$k = m(m + 1)/2$$

Statistics that are stored in the workspace that are part of symmetric matrices are stored in symmetric storage mode, i.e., only the lower triangular elements are stored. The workspace utilization is :

MOPT	IWT	Start	Length	Contents
All	All	1	m	Indicators of constant data
All	All	$m + 1$	m	First nonmissing data
0	All	$2m+1$	m	Deviation from temporary mean
0	Positive	$3m + 1$	1	Sum of weights
1, 2	All	$2m + 1$	m^2	Pairwise means
1, 2	Positive	$2m + m^2 + 1$	k	Pairwise sums of weights
3	All	$2m + 1$	m^2	Pairwise means
3	0	$2m + m^2 + 1$	m^2	Pairwise sums of products
3	Positive	$2m + m^2 + 1$	k	Pairwise sums of weights
3	Positive	$2m + k + m^2 + 1$	m^2	Pairwise sums of products

2. Informational errors

Type	Code	Description
3	12	The sum of the weights is zero. The means, variance and covariances are set to NaN.
3	13	The sum of the weights is zero. The means and correlations are set to NaN.
3	14	Correlations are requested but the observations on a variable are constant. The pertinent correlations are set to NaN.
3	15	Variances and covariances are requested but fewer than two valid observations are present for some variables. The corresponding variances or covariances are set to NaN.
3	16	Pairwise correlations are requested but the observations on a variable are constant. The pertinent correlations are set to NaN.
3	17	Correlations are requested but fewer than two valid observations are present for some variables. The corresponding variances or covariances are set to NaN.
4	10	More observations have been deleted than were originally entered.
4	11	More observations have been deleted from $\text{COV}(i, j)$ than were originally entered. $\text{INCD}(i, j)$ is less than zero.

- 4 18 Different observations have been deleted from $\text{COV}(i, j)$ than were originally entered. $\text{COV}(i, j)$ is less than zero.

Description

Routine `CORVC` computes estimates of correlations, covariances, or sums of squares and crossproducts for a data matrix X . Weights and frequencies are allowed but not required. Also allowed are listwise or pairwise deletion of missing values. Routine `CORVC` is an “IDO routine,” so it may be called with all of the data in one invocation, or it may be called in several invocations with some (or none) of the data input during each call. By setting `NROW` to a negative integer, observations that have previously been added to the covariance/correlation statistics may be deleted from the statistics. Exercise care with this option, however, since the program may not be able to detect constant variables when negative `NROW` is used.

The means, (corrected) sums of squares, and (corrected) sums of crossproducts are computed using the method of provisional means. Let

$$\bar{x}_{ki}$$

denote the mean based upon i observations for the k -th variable, f_i denote the frequency of the i -th observation, w_i denote the weight of the i -th observation, and let c_{jki} denote the sum of crossproducts (or sum of squares if $j = k$) based upon i observations. Then, the method of provisional means finds new means and sums of crossproducts as follows:

The means and crossproducts are initialized as:

$$\begin{aligned}\bar{x}_{k0} &= 0.0 & k &= 1, \dots, p \\ c_{jk0} &= 0.0 & j, k &= 1, \dots, p\end{aligned}$$

where p denotes the number of variables. Letting $x_{k(i+1)}$ denote the k -th variable on observation $i + 1$, each new observation leads to the following updates for

$$\bar{x}_{ki}$$

and c_{jki} using update constant r_{i+1} :

$$\begin{aligned}r_{i+1} &= \frac{f_{i+1}w_{i+1}}{\sum_{j=1}^{i+1} f_j w_j} \\ \bar{x}_{k(i+1)} &= \bar{x}_{ki} + (x_{k(i+1)} - \bar{x}_{ki})r_{i+1} \\ c_{jk(i+1)} &= c_{jki} + w_{i+1}f_{i+1}(x_{j(i+1)} - \bar{x}_{ji})(x_{k(i+1)} - \bar{x}_{ki})(1 - r_{i+1})\end{aligned}$$

If there is no weight variable, weights of 1.0 are used. If there is no frequency column, frequencies of 1.0 are used. Means and variances are computed based upon all of the valid data for each variable or, if required, based upon all of the valid data for each pair of variables.

Usage Notes

In `CORVC`, each observation x_{ki} with weight w_i is assumed to have mean μ_k and variance

$$\sigma_k^2 / w_i$$

With these assumptions, CORVC uses the following definition of a sample mean:

$$\bar{x}_k = \frac{\sum_{i=1}^{n_r} f_i w_i x_{ki}}{\sum_{i=1}^{n_r} f_i w_i}$$

where n_r is the number of cases. The following formula defines the sample covariance, s_{jk} , between variables j and k :

$$s_{jk} = \frac{\sum_{i=1}^n f_i w_i (x_{ji} - \bar{x}_j)(x_{ki} - \bar{x}_k)}{\left(\sum_{i=1}^n f_i\right) - 1}$$

The sample correlation between variables j and k , r_{jk} , is defined as:

$$r_{jk} = \frac{s_{jk}}{\sqrt{s_{jj}s_{kk}}}$$

Example 2

In the second example, the IDO option is used. After the initialization step in which IDO = 1, the first 53 observations in the Fisher iris data are input, one observation at a time. The last three observations input are then deleted from the covariances by setting NROW = -1. Finally, the wrap-up step is accomplished by calling CORVC with IDO = 3. The output is identical to the output above.

```

USE IMSL_LIBRARIES
INTEGER   LDCOV, LDINCD, LDX, LDY, NVAR
PARAMETER (LDCOV=5, LDINCD=1, LDX=150, LDY=1, NVAR=5)
!
INTEGER   I, IDO, INCD(LDINCD,1), NMISS, NOBS, NOUT, NROW, NV
REAL      COV(LDCOV,NVAR), SUMWT, X(LDX,NVAR), XMEAN(NVAR), &
          Y(LDY,NVAR)
!
CALL GDATA (3, X, NOBS, NV)
!
CALL UMACH (2, NOUT)
!
!
IDO = 1
NROW = 0
!
          Initialization
CALL CORVC (NVAR, Y, COV, IDO=IDO, NROW=NROW, XMEAN=XMEAN, &
          INCD=INCD, NOBS=NOBS, NMISS=NMISS, SUMWT=SUMWT)
!
IDO = 2
NROW = 1
!
          Add the observations
DO 10 I=1, 53
  CALL SCOPY (NVAR, X(I:,1), LDX, Y(1:,1), 1)
  CALL CORVC (NVAR, Y, COV, IDO=IDO, NROW=NROW, XMEAN=XMEAN, &
          INCD=INCD, NOBS=NOBS, NMISS=NMISS, SUMWT=SUMWT)
10 CONTINUE
!
          Delete the last 3 added

```

```

NROW = -1
DO 20 I=51, 53
  CALL SCOPY (NVAR, X(I:,1), LDX, Y(1:,1), 1)
  CALL CORVC (NVAR, Y, COV, IDO=IDO, NROW=NROW, XMEAN=XMEAN, &
    INCD=INCD, NOBS=NOBS, NMISS=NMISS, SUMWT=SUMWT)
20 CONTINUE
!
!                               Wrap-up
  IDO = 3
  NROW = 0
  CALL CORVC (NVAR, Y, COV, IDO=IDO, NROW=NROW, XMEAN=XMEAN, &
    INCD=INCD, NOBS=NOBS, NMISS=NMISS, SUMWT=SUMWT)
  CALL WRRRN ('XMEAN', XMEAN, 1, NVAR, 1)
  CALL WRRRN ('COV', COV)
  CALL WRIRN ('INCD', INCD)
  WRITE (NOUT,*) ' NOBS = ', NOBS, ' NMISS = ', NMISS, ' SUMWT = ', &
    SUMWT
END

```

Output

		XMEAN				
	1	2	3	4	5	
	1.000	5.006	3.428	1.462	0.246	

		COV				
	1	2	3	4	5	
1	0.0000	0.0000	0.0000	0.0000	0.0000	
2	0.0000	0.1242	0.0992	0.0164	0.0103	
3	0.0000	0.0992	0.1437	0.0117	0.0093	
4	0.0000	0.0164	0.0117	0.0302	0.0061	
5	0.0000	0.0103	0.0093	0.0061	0.0111	


```

INCD
  50
NOBS =   50 NMISS =   0 SUMWT =   50.0000

```

COVPL

Computes a pooled variance-covariance matrix from the observations.

Required Arguments

NROW — The absolute value of *NROW* is the number of rows of *X* that contain an observation. (Input)

NROW may be positive, zero, or negative. Negative *NROW* means that the $-NROW$ rows of data are to be deleted from (most aspects of) the analysis. This should be done only if *IDO* is 2 or 3 and the wrap-up computations for *COV* have not been performed. When a negative value is input for *NROW*; it is assumed that each of the $-NROW$ rows of *X* has been input (with positive *NROW*) in previous invocations of *CORVC*. Use of negative values of *NROW* should be made with care since it is possible that a constant variable in the remaining data will not be recognized as such.

NVAR — Number of variables to be used in computing the covariance matrix. (Input)
The weight, frequency or group variables, if used, are not counted in **NVAR**.

X — $|\text{NROW}|$ by $\text{NVAR} + m$ matrix containing the data. (Input)
The number of columns of **X** that are used is $\text{NVAR} + m$, where m is 0, 1, 2, or 3 depending upon whether any columns in **X** contain frequencies, weights or group numbers.

NGROUP — Number of groups in the data. (Input)

COV — NVAR by NVAR matrix of covariances. (Output, if **IDO** = 0 or 1; input/output, if **IDO** = 2 or 3)

Optional Arguments

IDO — Processing option. (Input)
Default: **IDO** = 0.

IDO Action

- | | |
|---|--|
| 0 | This is the only invocation of COVPL and all the data are input at once. |
| 1 | This is the first invocation of COVPL with this data, and additional calls will be made. Initialization of program variables and updating for the NROW observations are performed. |
| 2 | This is an intermediate invocation of COVPL , and updating for the NROW observations is performed. |
| 3 | All statistics are updated for the NROW observations. The covariance matrix is computed. |

NCOL — Number of columns in matrix **X**.
Default: **NCOL** = size (**X**,2).

LDX — Leading dimension of **X** exactly as specified in the dimension statement in the calling program. (Input)
Default: **LDX** = size (**X**,1).

IND — Vector of length **NVAR** containing the column numbers in **X** to be used in computing the covariance matrices. (Input)
By default: **IND**(**I**) = **I**.

IFRQ — Frequency option. (Input)
IFRQ = 0 means that all frequencies are 1.0. Positive **IFRQ** indicates that column number **IFRQ** of **X** contains the frequencies. All frequencies should be integer values. The **NINT** (nearest integer) function is used to obtain integer frequencies if this is not

the case.
Default: IFRQ = 0.

IWT — Weighting option. (Input)

IWT = 0 means that all weights are 1.0. Positive IWT means that column IWT of X contains the weights. Negative weights are not allowed.
Default: IWT = 0.

IGRP — Column of X giving the group numbers. (Input)

If IGRP = 0, one group is assumed. If IGRP > 0, then column number IGRP of X contains the group number for the observation. Group numbers must be numbered 1, 2, ..., NGROUP. The NINT function is used to get integer values for the group numbers.
Default: IGRP = NVAR + 1.

NI — Vector of length NGROUP containing the numbers of observations in the groups.

(Output, if IDO = 0 or 1; input/output, if IDO = 2 or 3)
The *i*-th element of NI contains the number of observations in group *i*.

SWT — Vector of length NGROUP containing the sum of the weights times the frequencies in the groups. (Output, if IDO = 0 or 1; input/output, if IDO = 2 or 3)

XMEAN — NGROUP by NVAR matrix. (Output, if IDO = 0 or 1; input/output, if IDO = 2 or 3)

The *i*-th row of XMEAN contains the group *i* variable means.

LDXMEA — Leading dimension of XMEAN exactly as specified in the dimension statement in the calling program. (Input)

Default: LDXMEAN = size (XMEAN ,1).

LDCOV — Leading dimension of COV exactly as specified in the dimension statement of the calling program. (Input)

Default: LDCOV = size (COV,1).

NRMISS — Number of rows of data encountered in calls to COVPL containing missing values (NaN) for any of the variables used. (Output, if IDO = 0 or 1; input/output, if IDO = 2 or 3)

FORTRAN 90 Interface

Generic: CALL COVPL (NROW, NVAR, X, NGROUP, COV [,...])

Specific: The specific interface names are S_COVPL and D_COVPL.

FORTRAN 77 Interface

Single: CALL COVPL (IDO, NROW, NVAR, NCOL, X, LDX, IND, IFRQ, IWT, NGROUP, IGRP, NI, SWT, XMEAN, LDXMEA, COV, LDCOV, NRMISS)

Double: The double precision name is DCOVPL.

Example

The following example computes a pooled variance-covariance matrix for the Fisher iris data (see routine GDATA, see Chapter 19, Utilities). The first column in this data set is the group indicator. To illustrate the use of the IDO argument, multiple calls to COVPL are made.

```
!                                     Specifications
USE GDATA_INT
USE COVPL_INT
USE UMACH_INT
USE WRRRN_INT

INTEGER   IFRQ, IGRP, IWT, LDCOV, LDX, LDXMEA, NCOL, NGROUP, &
          NROW, NVAR
PARAMETER (IFRQ=0, IGRP=1, IWT=0, LDX=150, NCOL=5, NGROUP=3, &
          NROW=1, NVAR=4, LDCOV=NVAR, LDXMEA=NGROUP)

!
INTEGER   I, IDO, IND(4), NI(NGROUP), NOBS, NOUT, NRMISS, NV
REAL      COV(LDCOV,LDCOV), SWT(NGROUP), X(LDX,5), &
          XMEAN(LDXMEA,NVAR)

!
DATA IND/2, 3, 4, 5/

!
CALL GDATA (3, X, NOBS, NV)

!
IDO = 1
CALL COVPL (0, NVAR, X, NGROUP, COV, IDO=IDO, IND=IND, IGRP=IGRP, &
          NI=NI, SWT=SWT, XMEAN=XMEAN, NRMISS=NRMISS)
!                                     Add the observations
IDO = 2
DO 10 I=1, NOBS
    CALL COVPL (NROW, NVAR, X(I:, 1:NCOL), NGROUP, COV, &
              IDO=IDO, IND=IND, IGRP+IGRP, NI=NI, SWT=SWT, &
              XMEAN=XMEAN, NRMISS=NRMISS)
10 CONTINUE

!                                     Summarize the statistics
IDO = 3
CALL COVPL (0, NVAR, X, NGROUP, COV, IDO=IDO, IND=IND, IGRP=IGRP, &
          NI=NI, SWT=SWT, XMEAN=XMEAN, NRMISS=NRMISS)

!
CALL UMACH (2, NOUT)
WRITE (NOUT,*) ' NRMISS = ', NRMISS
CALL WRRRN ('XMEAN', XMEAN)
CALL WRRRN ('COV', COV)
END
```

Output

```
NRMISS = 0
          XMEAN
          1      2      3      4
1  5.006  3.428  1.462  0.246
```

```

2  5.936  2.770  4.260  1.326
3  6.588  2.974  5.552  2.026

```

```

          COV
          1      2      3      4
1  0.2650  0.0927  0.1675  0.0384
2  0.0927  0.1154  0.0552  0.0327
3  0.1675  0.0552  0.1852  0.0427
4  0.0384  0.0327  0.0427  0.0419

```

Comments

1. Workspace may be explicitly provided, if desired, by use of `C2VPL/DC2VPL`. The reference is:

```
CALL C2VPL (IDO, NROW, NVAR, NCOL, X, LDX, IND, IFRQ, IWT,
NGROUP, IGRP, NI, SWT, XMEAN, LDXMEA, COV, LDCOV, NRMISS, D, OB,
XVAL, DIF)
```

The additional arguments are as follows:

D — Real work vector of length `NVAR`.

OB — Real work vector of length `NVAR`.

XVAL — Real work vector of length `NVAR * NGROUP`.

DIF — Real work vector of length `NVAR`.

2. Informational error

Type Code

```

3      1  The group number is not between 1 and NGROUP. The observation is
         ignored.

```

Description

Routine `COVPL` computes the pooled variance-covariance matrix from a matrix of observations. The within-groups means are also computed. Listwise deletion of missing values is assumed so that all observations used are “complete”; in any row of `X`, if an element in the “list” `IND`, `IGRP`, `IFRQ` or `IWT` is missing, then the row is not used. Routine `COVPL` should be used whenever one suspects that the data has been sampled from populations with different means but identical variance-covariance matrices. If these assumptions cannot be made, a different variance-covariance matrix should be estimated within each group.

When `IDO = 0`, the same computations occur as if `COVPL` were consecutively called with `IDO` equal to 1, 2, and 3. For brevity, the following discusses the computations with `IDO > 0`.

When `IDO = 1` variables are initialized, workspace is allocated, and input variables are checked for errors.

If `NROW` \neq 0 (for any value of `IDO`), the group observation totals, T_i , for $i = 1, \dots, g$, where g is the number of groups, are updated for the $|\text{NROW}|$ observations in X . The group totals are computed as: X

$$T_i = \sum_j \omega_{ij} f_{ij} x_{ij}$$

where ω_{ij} is the observation weight, x_{ij} is the j -th observation in the i -th group, and f_{ij} is the observation frequency.

Modified Givens rotations (see routines `SROTM` and `SROTMG` in the IMSL MATH/LIBRARY) are used in computing the Cholesky decomposition of the pooled sums of squares and crossproducts matrix. The interested reader is referred to Golub and Van Loan (1983) for details.

The group means and the pooled sample covariance matrix S are computed from the intermediate results when `IDO` = 3. These quantities are defined by

$$\bar{x}_{i\bullet} = \frac{T_i}{\sum_j \omega_{ij} f_{ij}}$$

$$S = \frac{1}{\sum_{i,j} f_{ij} - g} \sum_{i,j} \omega_{ij} f_{ij} (x_{ij} - \bar{x}_{i\bullet})(x_{ij} - \bar{x}_{i\bullet})^T$$

Occasionally, the Cholesky factorization, such that $S = U^T U$ where U is lower triangular of the pooled sample cross-products matrix, may be desired. U may be computed from the output array `COV`, and the workspace array `D` returned in calls to `C2VPL`. The Cholesky factor U can be computed prior to calling `C2VPL` with `IDO` = 3 by multiplying the elements in the i -th row of `COV` by

$$\sqrt{D_i} / \sqrt{\sum_{ij} f_{ij} - g}.$$

If subsequent calls to `C2VPL` are to be made, `COV` must not be modified in computing U .

PCORR

Computes partial correlations or covariances from the covariance or correlation matrix.

Required Arguments

COR — `NVAR` by `NVAR` correlation or covariance matrix. (Input)

NIND — Number of “independent” variables to be used in the partial correlations. (Input)

If `NIND` is -1 , the independent variables are taken to be the `NVAR` - `NDEP` variables not in `INDDEP`. If `NIND` is zero, no independent variables are used, and p -values for the input dependent (see `INDDEP`) correlations (or covariances) are computed. The partial correlations (covariances) are the correlations (covariances) between the dependent variables after removing the linear effect of the independent variables. `NIND` and `NDEP` cannot simultaneously be -1 .

IND — Vector of length $NIND$ containing the column (or row) numbers in COR of the independent variables. (Input, if $NIND > 0$; not referenced otherwise)
If $NIND$ is negative or zero, IND is not used and can be dimensioned of length 1 in the calling program.

NDEP — Number of variables for which partial correlations (covariances) are desired (the number of “dependent” variables). (Input)
If $NDEP$ is -1 , the dependent variables are taken as the $NVAR - NIND$ variables not in IND . $NIND$ and $NDEP$ cannot simultaneously be -1 .

INDDEP — Vector of length $NDEP$ containing the indices of the dependent variables. (Input, if $NDEP > 0$; not referenced otherwise)
If $NDEP$ is 1, $INDDEP$ is not used and can be dimensioned of length 1 in the calling program.

PCOR — Matrix of size m by m containing the partial correlations or partial covariances. (Output)
 $m = NDEP$ if $NDEP > 0$, and $m = NVAR - NIND$ otherwise. If $NIND = 0$, then COR and $PCOR$ can share the same memory location.

Optional Arguments

NVAR — Number of variables in COR . (Input)
Default: $NVAR = \text{size}(COR, 1)$.

LDCOR — Leading dimension of COR exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDCOR = \text{size}(COR, 1)$.

NDF — Number of degrees of freedom in COR . (Input)
If the number of degrees of freedom in COR varies from element to element, then a conservative choice for NDF is the minimum degrees of freedom for all elements in COR . If NDF is not known, then $NDF \leq 0$ defaults to $NDF = 100$.
Default: $NDF = 0$.

ICOR — Partial correlations/covariances option. (Input)
Default: $ICOR = 0$.

ICOR Action

1 Partial correlations are desired.

0 Partial covariances are desired.

Partial correlations can be computed when either a correlation or a covariance matrix in input in COR . To compute partial covariances, COR must contain a covariance matrix.

LDPCOR — Leading dimension of PCOR exactly as specified in the dimension statement of the calling program. (Input)

Default: LDPCOR = size (PCOR,1).

NDFP — Number of degrees of freedom in the test that the partial correlation (covariance) is zero. (Output)

This will usually be NDF – NIND but will be greater than this value if the variables in IND are computationally linearly related.

PVAL — Matrix of size m by m (see PCOR) containing the p -values for testing the null hypothesis that the associated partial correlation (covariance) is zero. (Output)

The p -values reported in PVAL assume that the observations from which COR was computed follow a multivariate normal distribution and that each element in COR has NDF degrees of freedom.

LDPVAL — Leading dimension of PVAL exactly as specified in the dimension statement in the calling program. (Input)

Default: LDPVAL = size(PVAL, 1).

FORTRAN 90 Interface

Generic: CALL PCORR (COR, NIND, IND, NDEP, INDDEP, PCOR [,...])

Specific: The specific interface names are S_PCORR and D_PCORR.

FORTRAN 77 Interface

Single: CALL PCORR (NVAR, COR, LDCOR, NDF, ICOR, NIND, IND, NDEP, INDDEP, PCOR, LDPCOR, NDFP, PVAL, LDPVAL)

Double: The double precision name is DPCORR.

Example

The following example computes partial correlations from a 9 variable correlation matrix originally given by Emmett (1949). The partial correlations between the remaining variables, after adjusting for variables 1, 3, and 9, are computed. Note in the output that the row and column labels are column numbers, not variable numbers. The corresponding variable numbers would be 2, 4, 5, 6, 7, and 8, respectively.

```
!                                     SPECIFICATIONS FOR PARAMETERS
USE PCORR_INT
USE UMACH_INT
USE WRRRN_INT

INTEGER    ICOR, LDCOR, LDP, LDPCOR, NDEP, NDF, NIND, NVAR
PARAMETER (ICOR=1, LDCOR=9, LDP=6, LDPCOR=6, NDEP=-1, NDF=30, &
           NIND=3, NVAR=9)
!
```

```

INTEGER      IND(NIND), INDDEP(1), NDFP, NOUT
REAL         COR(LDCOR,NVAR), P(LDP,LDP), PCOR(LDPCOR,LDPCOR)
!
DATA IND/1, 3, 9/
!
DATA COR/1.000, 0.523, 0.395, 0.471, 0.346, 0.426, 0.576, 0.434, &
      0.639, 0.523, 1.000, 0.479, 0.506, 0.418, 0.462, 0.547, &
      0.283, 0.645, 0.395, 0.479, 1.000, 0.355, 0.270, 0.254, &
      0.452, 0.219, 0.504, 0.471, 0.506, 0.355, 1.000, 0.691, &
      0.791, 0.443, 0.285, 0.505, 0.346, 0.418, 0.270, 0.691, &
      1.000, 0.679, 0.383, 0.149, 0.409, 0.426, 0.462, 0.254, &
      0.791, 0.679, 1.000, 0.372, 0.314, 0.472, 0.576, 0.547, &
      0.452, 0.443, 0.383, 0.372, 1.000, 0.385, 0.680, 0.434, &
      0.283, 0.219, 0.285, 0.149, 0.314, 0.385, 1.000, 0.470, &
      0.639, 0.645, 0.504, 0.505, 0.409, 0.472, 0.680, 0.470, &
      1.000/
!
CALL PCORR (COR, NIND, IND, NDEP, INDDEP, PCOR, NDF=NDF, &
           ICOR=ICOR, NDFP=NDFP, PVAL=P)
!
CALL UMACH (2, NOUT)
WRITE (NOUT,*) 'The degrees of freedom are ', NDFP
CALL WRRRN ('PCOR', PCOR)
CALL WRRRN ('P', P)
!
END

```

Output

The degrees of freedom are 27

		PCOR					
		1	2	3	4	5	6
1	1.000	0.224	0.194	0.211	0.125	-0.061	
2	0.224	1.000	0.605	0.720	0.092	0.025	
3	0.194	0.605	1.000	0.598	0.123	-0.077	
4	0.211	0.720	0.598	1.000	0.035	0.086	
5	0.125	0.092	0.123	0.035	1.000	0.062	
6	-0.061	0.025	-0.077	0.086	0.062	1.000	

		P					
		1	2	3	4	5	6
1	0.0000	0.2525	0.3232	0.2801	0.5249	0.7576	
2	0.2525	0.0000	0.0006	0.0000	0.6417	0.9000	
3	0.3232	0.0006	0.0000	0.0007	0.5328	0.6982	
4	0.2801	0.0000	0.0007	0.0000	0.8602	0.6650	
5	0.5249	0.6417	0.5328	0.8602	0.0000	0.7532	
6	0.7576	0.9000	0.6982	0.6650	0.7532	0.0000	

Comments

1. Workspace may be explicitly provided, if desired, by use of P2ORR/DP2ORR. The reference is:

CALL P2ORR (NVAR, COR, LDCOR, NDF, ICOR, NIND, IND, NDEP, INDDP, PCOR, LDPCOR, NDFP, PVAL, LDPVAL, SXY, SXX, LDSXX, IY, IX)

The additional arguments are as follows:

SXY — Work vector of length $m * n$.

SXX — Work vector of length n^2 .

LDSXX — The value of n .

IY — Work vector of length NVAR.

IX — Work vector of length NVAR.

2. Informational errors

Type Code

4	1	COR is incorrectly specified for two independent variables.
4	2	COR is incorrectly specified for an independent variable and a dependent variable.
4	3	COR is incorrectly specified for two dependent variables.
4	4	A computed partial correlation is greater than one.

Description

Routine PCORR computes partial correlations or partial covariances from an input correlation or covariance matrix. If the “independent” variables (the linear “effect” of the independent variables is removed in computing the partial correlations/covariances) are linearly related to one another, PCORR detects the linearity and eliminates one or more of the independent variables from the list of independent variables. The number of variables eliminated, if any, can be determined from argument NDFP.

Given a correlation or covariance matrix Σ partitioned as

$$\begin{pmatrix} \sum_{11} & \sum_{12} \\ \sum_{21} & \sum_{22} \end{pmatrix}$$

Routine PCORR computes the partial covariances (of the standardized variables if Σ is a correlation matrix) as

$$\sum_{22|1} = \sum_{22} - \sum_{21} \sum_{11}^{-1} \sum_{12}$$

If partial correlations are desired, these are computed as

$$P_{22|1} = \left[\text{diag} \left(\sum_{22|1} \right) \right]^{-\frac{1}{2}} \sum_{22|1} \left[\text{diag} \left(\sum_{22|1} \right) \right]^{-\frac{1}{2}}$$

where “diag” denotes the matrix containing the diagonal of its argument along its diagonal with zeros off the diagonal. If \sum_{11} is singular, then as many variables as required are deleted from

Σ_{11} (and Σ_{12}) in order to eliminate the linear dependency(ies). The computations then proceed as above.

The p -value for a partial correlation (covariance) tests the null hypothesis $H_0 : \rho_{ij|1} = 0$ ($H_0 : \sigma_{ij|1} = 0$), where $\rho_{ij|1}(\sigma_{ij|1})$ is the (i, j) element in matrix $P_{22|1}(\Sigma_{22|1})$. The p -values are returned in `PVAL`. If `NDF` is not known, the p -values are computed as if each element in `COR` had 100 degrees of freedom. When `NDF` is not known, the resulting p -values may be useful for comparison, but they should not be used as an approximation to the actual probabilities.

RBCOV

Computes a robust estimate of a covariance matrix and mean vector.

Required Arguments

WGHTS — User-supplied SUBROUTINE to compute observation weights. The form is `CALL WGHTS (R, NVAR, PERCNT, UU, WW, UP)`, where

R — Distance of observation from the mean vector at which weights are to be computed. (Input)

UU, WW, and UP are to be computed at distance **R**.

NVAR — Number of variables. (Input)

PERCNT — Percentage of outliers expected. (Input)

UU — Value of covariance matrix weighting function at distance **R**. (Output)

WW — Value of mean vector weighting function at distance **R**. (Output)

UP — Value of first derivative of **UU** with respect to **R**. (Output)

WGHTS must be declared `EXTERNAL` in the calling program. A standard weighting subroutine is provided as routine `R5COV/DR5COV`. See the “Algorithm” section for further description of the subroutine `WGHTS`.

X — `NOBS` by `NVAR + m` matrix containing the data. (Input)
m is 0, 1, 2, or 3 depending upon whether any columns in **X** contain frequencies, weights or group numbers.

IND — Vector of length `NVAR` containing the column numbers in **X** for which covariances are desired. (Input)

XMEAN — `NGROUP` by `NVAR` matrix containing the estimates of the location parameters in each group. (Output, if `INIT` \neq 2; input/output, otherwise)

Row *i* of **XMEAN** contains the location estimates for the variables in group *i*. The columns of **XMEAN** are in the order specified by **IND**.

COV — `NVAR` by `NVAR` matrix of estimated covariances. (Output, if `INIT` \neq 2; input/output, otherwise)

Optional Arguments

NOBS — Number of observations. (Input)

Default: NOBS = size (X,1).

NVAR — Number of variables in the covariance matrix. (Input)

Default: NVAR = size (IND,1).

NCOL — Number of columns in matrix X. (Input)

Default: NCOL = size (X,2).

LDX — Leading dimension of X exactly as specified in the dimension statement in the calling program. (Input)

Default: LDX = size (X,1).

IFRQ — Frequency option. (Input)

IFRQ = 0 means that all frequencies are 1.0. Positive IFRQ indicates that column number IFRQ of X contains the frequencies. All frequencies should be positive integer values. The NINT (nearest integer) function is used to obtain integer frequencies from X.

Default: IFRQ = 0.

IWT — Weighting option. (Input)

IWT = 0 means that all weights are 1.0. Positive IWT means that column IWT of X contains the positive weights. Negative weights are not allowed. Note that weights in column IWT are the proportionality constants used in computing a covariance matrix from observations with proportional covariance matrices. The weights used for robust estimation are computed in the estimation procedure.

Default: IWT = 0.

NGROUP — Number of groups (populations) in the data. (Input)

If the data comes from a single population, NGROUP = 1.

Default: NGROUP = 1.

IGRP — Column of X giving the group numbers. (Input)

If IGRP = 0, one group is assumed. If IGRP > 0, then column number IGRP of X contains the group number for the observation. Group numbers must be 1, 2, ..., NGROUP. The NINT intrinsic function is used to obtain integer group numbers

Default: IGRP = 0.

INIT — Estimate initialization option. (Input)

Default: INIT = 0.

INIT Method

0 Initial estimates are obtained as the usual estimate of a mean vector and of a covariance matrix.

- 1 Initial estimates based upon the median and interquartile range are used.
- 2 User input initial estimates are used.

IMTH— Option parameter giving the algorithm to be used in computing the estimates. (Input)
 Default: `IMTH = 0`.

IMTH Method

- 0 Huber’s conjugate-gradient algorithm is used.
- 1 Stahel’s algorithm is used.

PERCNT— Percentage of gross errors expected in the data. (Input)
`PERCNT` is in the range from zero to 100 and contains the percentage of outliers expected in the data. `PERCNT` is usually only used if IMSL supplied weighting subroutine `R5COV/DR5COV` is used as the subroutine `WGHTS`.
 Default: `PERCNT = 0.e0`.

MAXIT— Maximum number of iterations. (Input)
`MAXIT = 30` is typical.
 Default: `MAXIT = 30`.

EPS— Convergence criterion. (Input)
 When the maximum absolute change in a location or covariance estimate is less than `EPS`, convergence is assumed.
 Default: `EPS = 1.e-4`.

NI— Vector of length `NGROUP` containing the number of observations in each group. (Output)

SWT— Vector of length `NGROUP` containing the sum of the weights times the frequencies for the observations in each group. (Output)

LDXMEA— Leading dimension of `XMEAN` exactly as specified in the dimension statement in the calling program. (Input)
 Default: `LDXMEA = size (XMEAN,1)`.

LDCOV— Leading dimension of `COV` exactly as specified in the dimension statement of the calling program. (Input)
 Default: `LDCOV = size (COV,1)`.

CNST— Vector of length 4 containing some constants computed by `RBCOV`. (Output)
`CNST(1)` contains the constant β (see the “Algorithm” section) used to ensure that the estimated covariance matrix has unbiased expectation (for given mean vector) for a multivariate normal density. `CNST(2)`, `CNST(3)`, and `CNST(4)` are the parameters a , b ,

and c , respectively, in IMSL-supplied subroutine R5COV/DR5COV. They are set to NaN (not a number) if R5COV is not used.

NRMIS — Number of rows of data in X containing any missing values (NaN, not a number) in the columns IND, IWT, IFRQ, or IGRP. (Output)
Rows of X contributing to NRMIS are ignored in all other computations.

FORTRAN 90 Interface

Generic: CALL RBCOV (WGHTS, X, IND, XMEAN, COV [,...])

Specific: The specific interface names are S_RBCOV and D_RBCOV.

FORTRAN 77 Interface

Single: CALL RBCOV (WGHTS, NOBS, NVAR, NCOL, X, LDX, IND, IFRQ, IWT, NGROUP, IGRP, INIT, IMTH, PERCNT, MAXIT, EPS, NI, SWT, XMEAN, LDXMEA, COV, LDCOV, CNST, NRMIS)

Double: The double precision name is DRBCOV.

Example

The following example computes estimates of the mean vectors and the pooled covariance matrix for the Fisher iris data (routine GDATA, (page 1474), provides these data with the group indicator in the first column.). For comparison, these estimates are first computed via routine COVPL (page 348). Routine RBCOV with PERCNT = 0.02 is then used to compute the robust estimates. As can be seen from the output, the resulting estimates are quite similar.

To study the behavior of RBCOV, three observations are made into outliers, and, again, both COVPL and RBCOV are used to compute estimates. When outliers are present, COVPL gives estimates that have clearly been adversely affected, while the estimates produced by RBCOV are close to the estimates produced when no outliers are present.

In both calls to RBCOV, the usual pooled estimates were used for the initial estimates, and IMSL supplied routine R5COV with argument PERCNT = 0.02 was used. Because neither NOBS or PERCNT changed in the two calls, the values returned in CNST are identical. If the percentage of gross errors expected in the data, PERCNT, is not known, a reasonable strategy is to use a value of PERCNT that is such that larger values do not result in significant changes in the estimates.

```
USE IMSL_LIBRARIES
INTEGER IGRP, LDCOV, LDX, LDXMEA, MAXIT, NCOL, NGROUP, &
        NOBS, NV, NVAR
REAL PERCNT
PARAMETER (IGRP=1, NCOL=5, NGROUP=3, NOBS=150, NV=5, NVAR=4, &
        PERCNT=2.0, LDCOV=NVAR, LDX=NOBS, LDXMEA=NGROUP)
!
INTEGER IND(NVAR), NI(NGROUP), NOB1, NOUT, NRMIS, NV1
REAL CNST(4), COV(LDCOV,NVAR), SWT(NGROUP), X(LDX,NCOL), &
        XMEAN(NGROUP,NVAR)
EXTERNAL S_R5COV
```

```

!
DATA IND/2, 3, 4, 5/
!
CALL GDATA (3, X, NOB1, NV1)
!
CALL COVPL (NOBS, NVAR, X, NGROUP, COV, IND=IND, IGRP=IGRP, &
           XMEAN=XMEAN)
!
CALL UMACH (2, NOUT)
WRITE (NOUT,*) 'COVPL estimates with no outliers'
CALL WRRRN ('XMEAN', XMEAN)
CALL WRRRN ('COV', COV, ITRING=1)
!
CALL RBCOV (S_R5COV, X, IND, XMEAN, COV, NGROUP=NGROUP, IGRP=IGRP, &
           PERCNT=PERCNT, NI=NI, SWT=SWT, CNST=CNST)
!
WRITE (NOUT,*) 'RBCOV estimates with no outliers'
CALL WRRRN ('XMEAN', XMEAN)
CALL WRRRN ('COV', COV, ITRING=1)
CALL WRRRN ('SWT', SWT, 1, NGROUP, 1)
CALL WRIRN ('NI', NI, 1, NGROUP, 1)
CALL WRRRN ('CNST', CNST, 1, 4, 1)
!
X(1,2)   = 100.0
X(5,5)   = 100.0
X(100,3) = -100.0
!
CALL COVPL (NOBS, NVAR, X, NGROUP, COV, IND=IND, IGRP=IGRP, &
           XMEAN=XMEAN)
!
CALL UMACH (2, NOUT)
WRITE (NOUT,*) 'COVPL estimates with three outliers'
CALL WRRRN ('XMEAN', XMEAN)
CALL WRRRN ('COV', COV, ITRING=1)
!
CALL RBCOV (S_R5COV, X, IND, XMEAN, COV, NGROUP=NGROUP, &
           IGRP=IGRP, PERCNT=PERCNT, NI=NI, SWT=SWT, CNST=CNST)
!
WRITE (NOUT,*) 'RBCOV estimates with three outliers'
CALL WRRRN ('XMEAN', XMEAN)
CALL WRRRN ('COV', COV, ITRING=1)
CALL WRRRN ('SWT', SWT, 1, NGROUP, 1)
CALL WRIRN ('NI', NI, 1, NGROUP, 1)
CALL WRRRN ('CNST', CNST, 1, 4, 1)
!
END

```

Output

COVPL estimates with no outliers

	XMEAN			
	1	2	3	4
1	5.006	3.428	1.462	0.246
2	5.936	2.770	4.260	1.326
3	6.588	2.974	5.552	2.026

	COV			
	1	2	3	4
1	0.2650	0.0927	0.1675	0.0384
2		0.1154	0.0552	0.0327
3			0.1852	0.0427
4				0.0419

RBCOV estimates with no outliers

	XMEAN			
	1	2	3	4
1	4.989	3.411	1.465	0.244
2	5.951	2.784	4.265	1.324
3	6.529	2.970	5.489	2.026

	COV			
	1	2	3	4
1	0.2474	0.0872	0.1535	0.0360
2		0.1073	0.0538	0.0322
3			0.1705	0.0412
4				0.0401

	SWT		
	1	2	3
50.00	50.00	50.00	50.00

	NI		
	1	2	3
50	50	50	50

	CNST			
	1	2	3	4
0.972	0.000	3.093	1.717	

COVPL estimates with three outliers

	XMEAN			
	1	2	3	4
1	6.904	3.428	1.462	2.242
2	5.936	0.714	4.260	1.326
3	6.588	2.974	5.552	2.026

	COV			
	1	2	3	4
1	60.43	0.30	0.13	-1.28
2		70.53	0.17	0.17
3			0.19	0.00
4				66.38

RBCOV estimates with three outliers

	XMEAN			
	1	2	3	4
1	4.999	3.405	1.468	0.253

```

2  5.959  2.772  4.271  1.324
3  6.528  2.970  5.489  2.026

```

```

          COV
          1      2      3      4
1  0.2567  0.0885  0.1553  0.0361
2          0.1133  0.0546  0.0324
3          0.1723  0.0412
4          0.0424

```

```

          SWT
          1      2      3
50.00  50.00  50.00

```

```

          NI
          1      2      3
50    50    50

```

```

          CNST
          1      2      3      4
0.972  0.000  3.093  1.717

```

Comments

1. Workspace may be explicitly provided, if desired, by use of `R2COV/DR2COV`. The reference is:

```

CALL R2COV (WGHTS, NOBS, NVAR, NCOL, X, LDX, IND, IFRQ, IWT,
NGROUP, IGRP, INIT, IMTH, PERCNT, MAXIT, EPS, NI, SWT, XMEAN,
LDXMEA, COV, LDICOV, CNST, NRMISS, D, U, GXB, OB, OB1, OB2, SWW,
WK, IRN, ISF)

```

The additional arguments are as follows:

D — Work vector of length `NVAR`.

U — Work vector of length $\max(m * \text{NVAR}, \text{NGROUP}) * \text{NVAR}$; where $m = 2$ if `IMTH = 0`, and $m = 1$ otherwise.

GXB — Work vector of length `NVAR * NGROUP`.

OB — Work vector of length `NVAR`.

OB1 — Work vector of length `NVAR`.

OB2 — Work vector of length `NVAR`.

SWW — Work vector of length `NGROUP`.

WK — Work vector of length `NOBS`.

IRN — Work vector of length `NOBS`.

ISF — Work vector of length `NGROUP`.

2. Informational errors

Type Code

4 1 The derivative of `UU` with respect `R` is not correctly specified.

Description

Routine `RBCOV` computes robust M-estimates of the mean and covariance matrix from a matrix of observations. A pooled estimate of the covariance matrix is computed when multiple groups are present in the input data. M-estimate weights are obtained from a user specified weighting subroutine. In addition, user specified observation weights and frequencies may be given for each row in `X`. Listwise deletion of missing values is assumed so that all observations used are “complete.” In any row of `X`, if any column in the list determined by `IND`, `IFRQ`, `IWT`, or `IGRP` is missing, the row is not used.

Let $f(x; \mu_i, \Sigma)$ denote the density of an observation p -vector x in population (group) i with mean vector μ_i , for groups $i = 1, \dots, \tau$. Let the covariance matrix Σ be such that $\Sigma = R^T R$. If

$$y = R^{-T}(x - \mu_i)$$

then

$$g(y) = |\Sigma|^{1/2} f(R^T y + \mu_i; \mu_i, \Sigma)$$

It is assumed that $g(y)$ is a spherically symmetric density in p -dimensions.

In `RBCOV`, Σ and μ_i are estimated as the solutions

$$(\hat{\Sigma}, \hat{\mu}_i)$$

of the estimation equations

$$\frac{1}{n} \sum_{j=1}^{n_i} f_{ij} \omega_{ij} w(r_{ij}) y_{ij} = 0$$

and

$$\frac{1}{n} \sum_{i=1}^{\tau} \sum_{j=1}^{n_i} f_{ij} \omega_{ij} \left[(u(r_{ij}) y_{ij} y_{ij}^T - \beta I_p) \right] = 0$$

where i indexes the τ groups, n_i is the number of observations in group i , f_{ij} is the frequency for the j -th observation in group i , ω_{ij} is the observation weight specified in column `IWT` of `X`, I_p is a $p \times p$ identity matrix,

$$r_{ij} = \sqrt{y_{ij}^T y_{ij}}$$

$w(r)$ and $u(r)$ are weighting functions specified by the user through subroutine `WGHTS`, and where β is a constant computed by the program to make the expected weighted Mahalanobis

distance $(y^T y)$ equal the expected Mahalanobis distance from a multivariate normal distribution (see Marazzi 1985). The constant β is described more fully below.

Routine `RBCOV` uses one of two algorithms for solving the estimation equations. The first algorithm is discussed in detail in Huber (1981) and is a variant of the conjugate gradient method. The second algorithm is due to Stahel (1981) and is discussed in detail by Marazzi (1985). In both algorithms, correction vectors T_{ki} for the group i means and correction matrix $W_k = I_p + U_k$ for the Cholesky factorization of Σ are found such that the updated mean vectors are given by

$$\hat{\mu}_{i,k+1} = \hat{\mu}_{i,k} + T_{ki}$$

and the updated matrix R is given

$$\hat{R}_{k+1} = W_k \hat{R}_k$$

where k is the iteration number and

$$\hat{\Sigma}_k = R_k^T R_k$$

When all elements of U_k and T_{ki} are less than $\varepsilon = \text{EPS}$, convergence is assumed.

Three methods for obtaining initial estimates are allowed. In the first method, the sample weighted estimate of Σ is computed (using routine `COVPL`, page 348). In the second method, estimates based upon the median and the interquartile range are used. Finally, in the last method, the user inputs initial estimates.

Routine `RBCOV` computes estimates for any weighting functions u and w . The constant β is chosen such that $E(u(r)r^2) = p\beta$ where the expectation is with respect to a standard p -variate multivariate normal distribution. This yields estimates with the correct expectation for the multivariate normal distribution (for given mean vector). The expectation is computed via integration of estimated spline functions. 200 knots are used on an equally spaced grid from 0.0 to the 99.999 percentile of a

$$\chi_p^2$$

distribution. An error estimate is computed based upon 100 of these knots. If the estimated relative error is greater than 0.001, a warning message is issued. If β is not computed accurately (*i.e.*, if the warning message is issued), the computed estimates are still optimal, but the scale of the estimated covariance matrix may need to be multiplied by a constant in order for

$$\hat{\Sigma}$$

to have the correct multivariate normal covariance expectation.

The Weighting Subroutine

The name of the weighting subroutine (`WGHTS`) is input into `RBCOV`. User-supplied weights may be used. Alternatively, the user may input the name of the IMSL-supplied subroutine, `S_R5COV` in single precision, or `D_R5COV` in double precision. The weights computed by this subroutine are the “minimax” weights of Huber (1981, pages 231–235), with `PERCNT` expected gross errors. Huber’s (1981) weighting equations are given by:

$$u(r) = \begin{cases} \frac{a^2}{r^2} & r < a \\ 1 & a \leq r \leq b \\ \frac{b^2}{r^2} & r > b \end{cases}$$

$$w(r) = \min\left(1, \frac{c}{r}\right)$$

The constants a , b , and c depend upon the number of variables p and upon the expected percentage of gross errors. They are computed by `R5COV` as the zeroes of equations given by Huber and are returned in the array `CNST` from `RBCOV`.

CTRHO

Estimates the bivariate normal correlation coefficient using a contingency table.

Required Arguments

TABLE — `NROW` by `NCOL` contingency table containing the observed counts. (Input)

RHO — Maximum likelihood estimate of the correlation coefficient. (Output)

VAR — Estimated asymptotic variance of `RHO`. (Output)

PLTMY — Vector of length `NROW + NCOL - 2` containing the points of polytomy of the marginal rows and columns of `TABLE`. (Output)
The first `NROW - 1` elements of `PLTMY` are the points of polytomy for the rows while the last `NCOL - 1` elements are the points of polytomy for the columns.

PROB — `NROW` by `NCOL` matrix containing the bivariate normal probabilities corresponding to `RHO` and `PLTMY`. (Output)

DRIV — `NROW` by `NCOL` matrix containing the partial derivatives of the bivariate normal probability with respect to `RHO`. (Output)

Optional Arguments

NROW — Number of rows in the table. (Input)
Default: `NROW = size(TABLE,1)`.

NCOL — Number of columns in the table. (Input)
Default: `NCOL = size(TABLE,2)`.

LDTABL — Leading dimension of `TABLE` exactly as specified in the dimension statement of the calling program. (Input)
Default: `LDTABL = size(TABLE,1)`.

EPS — Convergence criterion in the iterative estimation. (Input)
 RHO will be within EPS of the maximum likelihood estimate unless roundoff errors prevent this precision. EPS must be less than 2. EPS less than or equal to zero defaults to 0.00001.
 Default: EPS = .00001.

LDPROB — Leading dimension of PROB exactly as specified in the dimension statement in the calling program. (Input)
 Default: LDPROB = size (PROB,1).

LDDRIV — Leading dimension of DRIV exactly as specified in the dimension statement in the calling program. (Input)
 Default: LDDRIV = size (DRIV,1).

FORTRAN 90 Interface

Generic: CALL CTRHO (TABLE, RHO, VAR, PLTMY, PROB, DRIV [,...])

Specific: The specific interface names are S_CTRHO and D_CTRHO.

FORTRAN 77 Interface

Single: CALL CTRHO (NROW, NCOL, TABLE, LDTABL, EPS, RHO, VAR, PLTMY, PROB, LDPROB, DRIV, LDDRIV)

Double: The double precision name is DCTRHO.

Example

The data are taken from Martinson and Hamdan (1972), who attribute it to Karl Pearson. The row variable is head breadth (in millimeters) for a human male while the column variable is the head breadth of his sister. Head breadth increases across the columns and decreases down the row. The row and column variables have been categorized into one of three intervals. The original table is as follows:

1.0	36.5	77.5
52.5	340.5	143.5
40.5	58.0	9.0

Note that routine CTRHO can accept other than integer counts. It is not clear from Martinson and Hamdan (1972) how the non-integral counts arise in the table here. The correlation is estimated to be 0.5502.

```

USE CTRHO_INT
USE UMACH_INT
USE WRRRN_INT

INTEGER LDDRIV, LDPROB, LDTABL, NCOL, NROW
PARAMETER (LDDRIV=3, LDPROB=3, LDTABL=3, NCOL=3, NROW=3)
!
INTEGER NOUT

```

```

REAL      DRIV(LDDRIV, NCOL), PLTMY(NROW+NCOL-2), &
          PROB(LDPROB, NCOL), RHO, TABLE(LDTABL, NCOL), VAR
!
DATA TABLE/1.0, 52.5, 40.5, 36.5, 340.5, 58.0, 77.5, 143.5, 9.0/
!
!
CALL CTRHO (TABLE, RHO, VAR, PLTMY, PROB, DRIV)
!
CALL UMACH (2, NOUT)
WRITE (NOUT,*) 'RHO =', RHO, '      VAR =', VAR
CALL WRRRN ('PLTMY', PLTMY, 1, NROW+NCOL-2, 1, 0)
CALL WRRRN ('PROB', PROB)
CALL WRRRN ('DRIV', DRIV)
END

```

Output

```
RHO = 0.549125      VAR = 1.33199E-03
```

```

          PLTMY
      1      2      3      4
-1.073  1.030 -1.156  0.516

```

```

          PROB
      1      2      3
1  0.0015  0.0517  0.0983
2  0.0700  0.4398  0.1970
3  0.0523  0.0816  0.0077

```

```

          DRIV
      1      2      3
1 -0.0134 -0.0984  0.1118
2 -0.0717  0.1388 -0.0672
3  0.0851 -0.0404 -0.0447

```

Description

Routine CTRHO computes the maximum likelihood estimate and the asymptotic variance for the correlation coefficient of a bivariate normal population from a two-way contingency table. The maximum likelihood estimates are conditional upon the points of polytomy in the marginal distribution. The resulting estimate for the correlation coefficient should be very close to the unconditional estimate (see Martinson and Hamdan 1972).

The points of polytomy for the row and column marginal probabilities are first computed. If the i -th cumulative column marginal is denoted by p_{ci} , then the point of polytomy x_i is given as $\Phi^{-1}(p_{ci})$, where Φ denotes the cumulative normal distribution. Let $\alpha_i, i = 0, \dots, r$ denote these points for the row marginal cumulative probabilities where $r = \text{NROW}$, $\alpha_0 = -\infty$, and $\alpha_r = \infty$. Similarly, let $\beta_j, j = 0, \dots, c$ denote the points of polytomy for the columns where $c = \text{NCOL}$. Then, the probability of the (i, j) cell in the table, p_{ij} , is defined as

$$p_{ij} = \Pr(\alpha_{i-1} < X < \alpha_i, \beta_{j-1} < Y < \beta_j)$$

where X and Y are the bivariate random variables. Maximum likelihood estimates for the correlation coefficient are computed based upon the bivariate normal density. The likelihood is specified by the multinomial distribution of the table using probabilities p_{ij} .

Routine `CTRHO` assumes that the row random variable decreases with increasing row number while the column variable increases with the column number. If this is not the case, the sign of the estimated correlation coefficient may need to be changed.

TETCC

Categorizes bivariate data and compute the tetrachoric correlation coefficient.

Required Arguments

- NROW*** — The absolute value of `NROW` is the number of observations currently in `X` and `Y`. (Input)
`NROW` may be positive, zero, or negative. Negative `NROW` means delete the $-NROW$ observations in `X` and `Y` from the analysis. In the usual case, in which all of the data have already been categorized into counts in `ICOUNT`, `NROW` should be set to 0 and `IDO` set to 3.
- X*** — Vector of length $|NROW|$ containing the observations on one variable. (Input)
- Y*** — Vector of length $|NROW|$ containing the observations on the second variable. (Input)
- HX*** — Constant used to categorize values of `X`. (Input)
See description of `ICOUNT`.
- HY*** — Constant used to categorize values of `Y`. (Input)
See description of `ICOUNT`.
- ICOUNT*** — 2 by 2 matrix containing counts. (Output, if `IDO` = 0 or 1; input/output, if `IDO` = 2 or 3.)
The elements of `ICOUNT` are the numbers of observations satisfying the following relations:
`ICOUNT(1, 1)` : $X(i) < HX$ and $Y(i) < HY$
`ICOUNT(1, 2)` : $X(i) < HX$ and $Y(i) \geq HY$
`ICOUNT(2, 1)` : $X(i) \geq HX$ and $Y(i) < HY$
`ICOUNT(2, 2)` : $X(i) \geq HX$ and $Y(i) \geq HY$
- NR*** — Number of real roots in the interval $(-1.0, 1.0)$ of the seventh-degree polynomial used to estimate the correlation coefficient. (Output)
- R*** — Vector of length 7 containing in the first `NR` positions estimates of the correlation coefficient. (Output)

RS — Estimate of the standard error of the estimates of the correlation coefficient(s).
(Output)

Optional Arguments

IDO — Processing option. (Input)
Default: `IDO = 0`.

IDO Action

- | | |
|---|--|
| 0 | This is the only invocation of <code>TETCC</code> , and all the data are input at once in <code>X</code> and <code>Y</code> . |
| 1 | This is the first invocation of <code>TETCC</code> with this data, and additional calls will be made. Initialization and updating for the data in <code>X</code> and <code>Y</code> are performed. |
| 2 | This is an intermediate invocation of <code>TETCC</code> , and updating for the observations in <code>X</code> and <code>Y</code> is performed. |
| 3 | Updating for the observations in <code>X</code> and <code>Y</code> is performed, and the tetrachoric correlation coefficient is computed using the values in <code>ICOUNT</code> . |

LDICOU — Leading dimension of `ICOUNT` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDICOU = size (ICOUNT,1)`.

FORTRAN 90 Interface

Generic: `CALL TETCC (NROW, X, Y, HX, HY, ICOUNT, NR, R, RS [,...])`

Specific: The specific interface names are `S_TETCC` and `D_TETCC`.

FORTRAN 77 Interface

Single: `CALL TETCC (IDO, NROW, X, Y, HX, HY, ICOUNT, LDICOU, NR, R, RS)`

Double: The double precision name is `DTETCC`.

Example 1

In the first example, the data are counts. The 374 in `ICOUNT(1, 1)` indicates that in the raw data there were 374 pairs having both values less than some cutoff point. The 186 in `ICOUNT(1, 2)` indicates that there were 186 pairs in the raw data for which the first value was less than its cutoff value and the second value was greater than or equal to its cutoff value.

```
USE UMACH_INT
USE TETCC_INT
```

```

INTEGER      I, ICOUNT(2,2), IDO, NOUT, NR, NROW
REAL        HX, HY, R(7), RS, X(1), Y(1)
!
CALL UMACH (2, NOUT)
ICOUNT(1,1) = 374
ICOUNT(1,2) = 186
ICOUNT(2,1) = 167
ICOUNT(2,2) = 203
IDO         = 3
NROW        = 0
CALL TETCC (NROW, X, Y, HX, HY, ICOUNT, NR, R, RS, IDO=IDO)
WRITE (NOUT,99998) NR, (R(I),I=1,NR)
99998 FORMAT (' Number of roots (estimates) is ', I1, '/', ' ', &
             'Estimate(s) = ', 7F10.5)
WRITE (NOUT,99999) RS
99999 FORMAT (' The estimated standard error is ', F10.5)
END

```

Output

```

Number of roots (estimates) is 1
Estimate(s) =      0.33511
The estimated standard error is      0.05255

```

Comments

- Informational errors

Type	Code	
3	1	Fewer than 200 observations are used.
3	2	The polynomial used to estimate the correlation coefficient has more than one root in the interval $(-1.0, 1.0)$. It is probable that the numerical precision is not good enough to obtain an estimate.
4	4	The proportion of counts in a row or column is so close to one that the inverse normal cdf cannot be computed.
4	6	The polynomial used to estimate the correlation coefficient has no roots in the interval $(-1.0, 1.0)$. It is probable that the numerical precision is not good enough to obtain an estimate.
- If data for X and Y are available, it is better to use the Pearson product moment correlation coefficient (as computed by routine `CORVC`, [page 340](#), for example) than to use the tetrachoric correlation coefficient.
- The tetrachoric correlation coefficient should be considered somewhat questionable if the sample size is less than 200, if the cutpoints `HX` and `HY` are not close to the medians, or if there are multiple roots of the estimating equation in the interval $(-1.0, 1.0)$. Also, the tetrachoric correlation coefficient is a better estimate of the true correlation coefficient if the true coefficient is large in absolute value.

Description

Routine `TETCC` computes the tetrachoric correlation coefficient for a bivariate sample, using either the sample itself or a two by two table of counts of the data. The tetrachoric correlation

coefficient is taken as the solution to the seventh-degree polynomial obtained from the first seven terms of the expansion given by Kendall and Stuart (1979, page 326).

The standard error estimate results from an approximate, asymptotic expression derived under the assumption of bivariate normality with zero correlation. The zero correlation assumption is not overly restrictive since most uses of this standard error would be in tests of zero correlation.

If all of the data is available, the Pearson product-moment correlation coefficient (which can be computed using routine `CORVC`, (page 340)) is a much better estimate for the population correlation coefficient than is the tetrachoric correlation coefficient. If the counts in `ICOUNT` are all that is available, call `TETCC` with `IDO = 3` and `NROW = 0`.

Example 2

In this example, some artificial bivariate normal data are generated using IMSL routine `RNMVN` (page 1379), and then, the tetrachoric correlation coefficient is computed. Since the mean (and median) of each variable is 0.0, the cutpoints `HX` and `HY` are set to 0.0.

```

USE IMSL_LIBRARIES
INTEGER I, ICOUNT(2,2), IRANK, NOUT, NR, NROW
REAL COV(2,2), HX, HY, R(7), RS, RSIG(2,2), X(1000), &
      XY(1000,2), Y(1000)
!
EQUIVALENCE (X, XY), (Y, XY(1,2))
!
CALL UMACH (2, NOUT)
!
!           Generate random sample from
!           bivariate normal with correlation
!           of 0.5.
COV(1,1) = 1.0
COV(1,2) = 0.5
COV(2,1) = 0.5
COV(2,2) = 1.0
!
!           Obtain the Cholesky factorization.
CALL CHFAC (COV, IRANK, RSIG)
!
!           Initialize seed of random number
!           generator.
CALL RNSET (123457)
CALL RNMVN (RSIG, XY)
!
NROW = 1000
HX = 0.0
HY = 0.0
CALL TETCC (NROW, X, Y, HX, HY, ICOUNT, NR, R, RS)
WRITE (NOUT,99997) ICOUNT
99997 FORMAT (' ICOUNT = ', 4I4)
WRITE (NOUT,99998) NR, (R(I),I=1,NR)
99998 FORMAT (' Number of roots (estimates) is ', I1, '/', ' ', &
             ' Estimate(s) = ', 7F10.5)
WRITE (NOUT,99999) RS
99999 FORMAT (' The estimated standard error is ', F10.5)
END

```

Output

```
ICOUNT = 326 163 171 340
Number of roots (estimates) is 1
Estimate(s) = 0.49824
The estimated standard error is 0.04968
```

BSPBS

Computes the biserial and point-biserial correlation coefficients for a dichotomous variable and a numerically measurable classification variable.

Required Arguments

A — 3 by κ matrix containing the frequencies and the class marks of the measured classification variable. (Input)
The first row of *A* contains frequencies for the classification variable when the dichotomous variable takes on one of its values, and the second row of *A* contains the frequencies when the dichotomous variable takes on its other value. The third row of *A* contains the values (class marks) of the classification variable. The elements of the first two rows of *A* must be nonnegative.

STAT — Vector of length 11 containing various statistics. (Output)

I	STAT(I)
1	Total count of the first value of the dichotomous variable (the sum of the first row of <i>A</i>)
2	Total count for the second value
3	Total count (sum of <i>STAT</i> (1) and <i>STAT</i> (2))
4	Mean of the measured variable
5	Mean of the measured variable in the first class of the dichotomy
6	Mean of the measured variable in the second class of the dichotomy
7	Standard deviation of the measured variable
8	Biserial correlation coefficient estimate
9	Standard deviation estimate for the biserial correlation coefficient estimate
10	Asymptotic significance level of the biserial correlation coefficient, that is, the probability of a more extreme value
11	Point-biserial correlation coefficient estimate

Optional Arguments

K — Number of classes for the measured classification variable. (Input)

Default: $K = \text{size}(A,2)$.

LDA — Leading dimension of *A* exactly as specified in the dimension statement in the calling program. (Input)

Default: $LDA = \text{size}(A,1)$.

FORTRAN 90 Interface

Generic: `CALL BSPBS (A, STAT[,...])`

Specific: The specific interface names are `S_BSPBS` and `D_BSPBS`.

FORTRAN 77 Interface

Single: `CALL BSPBS (K, A, LDA, STAT)`

Double: The double precision name is `DBSPBS`.

Example

The example is taken from Kendall and Stuart (1979, page 327). The data involve the classification of criminals as alcoholic (first row) or nonalcoholic for each level of a crimetype classification. The severity of the crime decreases with increasing column number. In the example, the column number is used for the column score. The biserial correlation of -0.17 indicates that more criminals responsible for the most serious crimes tend to be alcoholic.

```
USE IMSL_LIBRARIES
INTEGER    K, LDA
PARAMETER  (K=6, LDA=3)
!
REAL       A(LDA,K), STAT(11)
CHARACTER  CLABEL(2)*10, RLABEL(11)*10
!
DATA A/50, 43, 1, 88, 62, 2, 155, 110, 3, 379, 300, 4, &
      18, 14, 5, 63, 144, 6/
DATA RLABEL/'Count-1', 'Count-2', 'Count', 'Mean(X)', &
      'Mean(X-1)', 'Mean(X-2)', 'S-X', 'r-b', 'std(r-b)', &
      'p-value', 'r-p'/
DATA CLABEL/'Statistic', '    '/
!
CALL WRRRN('A', A)
!
CALL BSPBS (A, STAT)
!
CALL WRRRL ('    ', STAT, RLABEL, CLABEL, FMT='(W12.8)')
END
```

Output

	A					
	1	2	3	4	5	6
1	50.0	88.0	155.0	379.0	18.0	63.0
2	43.0	62.0	110.0	300.0	14.0	144.0
3	1.0	2.0	3.0	4.0	5.0	6.0

Statistic

Count-1	753.00
Count-2	673.00
Count	1426.00
Mean(X)	3.72
Mean(X-1)	3.55
Mean(X-2)	3.91
S-X	1.31
r-b	-0.17
std(r-b)	0.03
p-value	0.00
r-p	-0.14

Description

Routine `BSPBS` computes the biserial and point-biserial correlation coefficient for a dichotomous variable and a numerically measurable (classification) variable. Input to `BSPBS` is a $3 \times K$ array, `A`. The first two rows of `A` contain the frequencies for the dichotomous variable as measured at each level of the classification variable. The third row contains the values (class marks) to be used for the classification variable.

The biserial correlation coefficient should be used in situations where the dichotomous variable and the classification variable are assumed to come from a bivariate normal distribution. If this is not the case (i.e., if the bivariate normal assumption cannot be made), then the point-biserial correlation should be used (see Kendall and Stuart 1979, page 331).

Let $a_{\bullet 1}$ and $a_{\bullet 2}$ denote the total count in rows one and two of `A`, respectively, and let $n = a_{\bullet 1} + a_{\bullet 2}$. Let Φ denote the cumulative normal distribution; let a_{ij} , $i = 1, 2, j = 1, \dots, K$, denote the counts in rows 1 and 2 of `A`, and let x_j denote the values in row 3 of `A`. The biserial correlation coefficient r_b is computed as follows:

$$\begin{aligned}
p &= \frac{a_{*1}}{n} \\
z_p &= \Phi^{-1}(p) \\
\bar{x}_1 &= \frac{\sum_{j=1}^k a_{1j} x_j}{\sum_{j=1}^k a_{1j}} \\
\bar{x}_2 &= \frac{\sum_{j=1}^k (a_{1j} + a_{2j}) x_j}{n} \\
s_x^2 &= \frac{\sum_{j=1}^k (a_{1j} + a_{2j} - \bar{x})^2}{n-1} \\
r_b &= \frac{\bar{x}_1 - \bar{x}_2}{s_x} \frac{p(1-p)}{z_p} \\
\text{var}(r_b) &\approx \left(\frac{\sqrt{p(1-p)}}{z_p} - r_b^2 \right) \frac{1}{n}
\end{aligned}$$

Let

$$z = |r_b| / \sqrt{\text{var}(r_b)}$$

If the underlying distributions are normal with zero correlation, then z is asymptotically a standard normal deviate that may be used to test that the correlation is zero. The p -value for z is reported in `STAT(10)`.

The point-biserial correlation coefficient is computed as

$$r_p = \frac{z_p r_b}{\sqrt{p(1-p)}}$$

BSCAT

Computes the biserial correlation coefficient for a dichotomous variable and a classification variable.

Required Arguments

A — 2 by κ matrix containing the frequencies. (Input)

The first row of **A** contains frequencies for the classification variable when the dichotomous variable takes on one of its values, and the second row of **A** contains the frequencies when the dichotomous variable takes on its other value. No ordering is assumed for the values of the classification variable. The elements of **A** must be nonnegative.

STAT — Vector of length 5 containing various statistics. (Output)

I	STAT(I)
1	Total count of the first value of the dichotomous variable (the sum of the first row of <i>A</i>)
2	Total count for the second value
3	Total count (sum of <i>STAT(1)</i> and <i>STAT(2)</i>)
4	Absolute value of the biserial correlation coefficient
5	Square of the biserial correlation coefficient

Optional Arguments

K — Number of classes for the classification variable. (Input)
 Default: *K* = size (*A*,2).

LDA — Leading dimension of *A* exactly as specified in the dimension statement in the calling program. (Input)
 Default: *LDA* = size (*A*,1).

FORTRAN 90 Interface

Generic: CALL BSCAT (A, STAT [,...])

Specific: The specific interface names are S_BSCAT and D_BSCAT.

FORTRAN 77 Interface

Single: CALL BSCAT (K, A, LDA, STAT)

Double: The double precision name is DBSCAT.

Example

The example is taken from Kendall and Stuart (1979, page 327). The data involve the classification of criminals as alcoholic (first row) or nonalcoholic for each level of a crimetype classification. The severity of the crime decreases with increasing column number. The absolute value of the biserial correlation is 0.23.

```

USE WRRRN_INT
USE BSCAT_INT
USE WRRRL_INT

INTEGER      K, LDA
PARAMETER   (K=6, LDA=2)
!
REAL        A(LDA,K), STAT(5)
CHARACTER   CLABEL(2)*10, RLABEL(5)*10

```



```

!
DATA A/50, 43, 88, 62, 155, 110, 379, 300, 18, 14, 63, 144/
DATA RLABEL/'Count-1', 'Count-2', 'Count', 'r-b', '(r-b)**2'/
DATA CLABEL/'Statistic', ' ' /
!
CALL WRRRN ('A', A)
!
CALL BSCAT (A, STAT)
!
CALL WRRRL (' ', STAT, RLABEL, CLABEL, FMT='(W12.6)')
END

```

Output

	A					
	1	2	3	4	5	6
1	50.0	88.0	155.0	379.0	18.0	63.0
2	43.0	62.0	110.0	300.0	14.0	144.0
Statistic						
Count-1		753.00				
Count-2		673.00				
Count		1426.00				
r-b		0.23				
(r-b)**2		0.05				

Description

Routine BSCAT computes the biserial correlation coefficient for a dichotomous variable and a classification variable. The data are input in a $2 \times k$ array, A, where the row indicates the value of the dichotomous variable, and the column indicates the value of the classification variable. In BSCAT, column scores are computed as $x_i = \Phi^{-1}(a_{1i}/(a_{1i} + a_{2i}))$, and the row score is computed as $y = \Phi^{-1}(a_{\bullet 1}/(a_{\bullet 1} + a_{\bullet 2}))$, where $a_{\bullet 1}$ is the sum of the counts in row 1, $a_{\bullet 2}$ is the sum of the counts for row 2, and Φ denotes the cumulative normal distribution. Let N denote the total number of observations (the sum of the elements of A). Then, the biserial correlation is computed as

$$r_b^2 = \frac{\sum_{i=1}^k (a_{1i} + a_{2i}) x_i^2 - Ny^2}{N + \sum_{i=1}^k (a_{1i} + a_{2i}) x_i^2}$$

An underlying bivariate normal distribution is assumed. The validity of the estimate depends heavily upon this assumption.

CNCRD

Calculates and test the significance of the Kendall coefficient of concordance.

Required Arguments

X — NOBS by K matrix containing the data. (Input)
Each column of X is a set of observations (which can be converted to ranks) or a set of ranks.

FUZZ — Value to be used for determining ties. (Input)
If within a column of X , the difference between two elements is less than or equal to **FUZZ** in absolute value, then the elements are said to be tied.

SUMS — Vector of length NOBS containing the sums of the K ranks in the corresponding row of X . (Output)

STAT — Vector of length 4 containing the output statistics. (Output)

<i>i</i>	STAT(i)
1	W , the coefficient of concordance
2	Chi-squared statistic corresponding to W with $\text{NOBS} - 1$ degrees of freedom
3	Asymptotic probability of exceeding STAT(2) under the null hypothesis of independence
4	Kendall S statistic. This is the sum of the squared deviations from the expected sum of the ranks

Optional Arguments

NOBS — Number of observations per set of rankings. (Input)
Default: $\text{NOBS} = \text{size}(\text{SUMS}, 1)$.

K — Number of sets of rankings. (Input)
 K must be greater than or equal to two.
Default: $K = \text{size}(X, 2)$.

LDX — Leading dimension of X exactly as specified in the dimension statement in the calling program. (Input)
Default: $\text{LDX} = \text{size}(X, 1)$.

FORTRAN 90 Interface

Generic: CALL CNCRD (X, FUZZ, SUMS, STAT [, ...])

Specific: The specific interface names are S_CNCRD and D_CNCRD.

FORTRAN 77 Interface

Single: CALL CNCRD (NOBS, K, X, LDX, FUZZ, SUMS, STAT)

Double: The double precision name is DCNCRD.

Example

The example is taken from Kendall (1962, pages 97–98). It involves ten observations in three sets. The resulting coefficient of concordance, 0.828, is quite large, indicating a strong relationship.

```
USE WRRRN_INT
USE CNCRD_INT
USE WRRRL_INT

INTEGER      K, LDX, NOBS
REAL         FUZZ
PARAMETER   (FUZZ=0.0001, K=3, LDX=10, NOBS=10)
!
REAL         STAT(4), SUMS(NOBS), X(LDX,K)
CHARACTER    CLABEL(2)*11, RLABEL(4)*11
!
DATA RLABEL/'W', 'Chi-squared', 'p-value', 'S'/
DATA CLABEL/'Statistic', ' ' /
DATA X/1, 4.5, 2, 4.5, 3, 7.5, 6, 9, 7.5, 10, 2.5, 1, 2.5, 4.5, &
      4.5, 8, 9, 6.5, 10, 6.5, 2, 1, 4.5, 4.5, 4.5, 4.5, 8, 8, 8, &
      10/
!
CALL WRRRN ('X', X)
!
CALL CNCRD (X, FUZZ, SUMS, STAT)
!
CALL WRRRN ('SUMS', SUMS, 1, NOBS, 1, 0)
CALL WRRRL (' %/%', STAT, RLABEL, CLABEL, FMT='(W10.6)')
END
```

Output

	X		
	1	2	3
1	1.00	2.50	2.00
2	4.50	1.00	1.00
3	2.00	2.50	4.50
4	4.50	4.50	4.50
5	3.00	4.50	4.50
6	7.50	8.00	4.50
7	6.00	9.00	8.00
8	9.00	6.50	8.00
9	7.50	10.00	8.00
10	10.00	6.50	10.00

	SUMS									
	1	2	3	4	5	6	7	8	9	10
	5.50	6.50	9.00	13.50	12.00	20.00	23.00	23.50	25.50	26.50

Statistic	
W	0.828
Chi-squared	22.349
p-value	0.008
S	591.000

Comments

1. Workspace may be explicitly provided, if desired, by use of `C2CRD/DC2CRD`. The reference is:

```
CALL C2CRD (NOBS, K, X, LDX, FUZZ, SUMS, STAT, IWK, XWK)
```

The additional arguments are as follows:

IWK — Work vector of length `NOBS`.

XWK — Work vector of length `NOBS * K`.

2. Informational errors

Type	Code	
------	------	--

- | | | |
|---|---|--|
| 3 | 6 | Within each of the <code>K</code> sets of rankings all observations are tied. <code>STAT(1) - STAT(3)</code> cannot be computed and are set to NaN (not a number). |
| 3 | 7 | The chi-squared degrees of freedom is less than 7. <code>STAT(3)</code> should be regarded with suspicion. |

Description

Routine `CNCRD` computes and tests the significance of the Kendall coefficient of concordance.

The coefficient of concordance is computed as follows: Within each of the k sets the $n = \text{NOBS}$ observations are ranked. Tied ranks are used for tied observations where two observations are tied if they are within `FUZZ` of each other. Let x_i denote the sum of the ranks for the i -th observation over the k sets. The mean of the x_i is

$$\bar{x} = k(n+1)/2$$

Using this mean, compute the sums of squares of the x_i about their mean as

$$S = \sum_{i=1}^N (x_i - \bar{x})^2.$$

This is the Kendall S statistic (`STAT(4)`). If there are tied ranks within a set i , compute the adjustment

$$T_i^* = \frac{\sum_j (t_j^3 - t_j)}{12}$$

where t_j is the number of ties in the j -th group of ties, and the summation is over all tie groups for the set. Kendall's coefficient of concordance, W , is computed as

$$W = \frac{12S}{k^2(n^3 - n) - k \sum_{i=1}^k T_i^*}$$

Kendall's coefficient of concordance is related to the Friedman one-way analysis of variance on ranks chi-squared test statistic T (see IMSL routine `FRDMN`, page 620), as

$$W = \frac{T}{n(k-1)}$$

When n or k is small, tables of the exact distribution of W exist. See Owen (1962, pages 396–397). The probability reported in `STAT(3)` is asymptotic. It is only approximate when k and n are small.

KENDL

Computes and test Kendall's rank correlation coefficient.

Required Arguments

- X*** — Vector of length `NOBS` containing the observations for the first variable. (Input)
- Y*** — Vector of length `NOBS` containing the observations for the second variable. (Input)
- FUZZ*** — Value used to determine ties in X or Y . (Input)
Two observations are said to be tied if the absolute value of their difference is less than or equal to `FUZZ`.
- STAT*** — Vector of length 9 containing some output statistics. (Output)
See the "Description" section for full definitions. The output statistics are

<i>i</i>	STAT(<i>i</i>)
1	Kendall τ_a (assumes no ties)
2	Kendall τ_b (corrects for ties)
3	Ties statistic for variable X
4	Ties statistic for variable Y
5	Statistic S corresponding to Kendall's τ
6	Exact probability of achieving a score at least as large as S . S is not calculated if <code>NOBS</code> is too large (34 on many computers) or there are ties. In either case, <code>STAT(6)</code> is set to NaN (not a number).

- 7 The same probability as `STAT(6)` but using a normal approximation. (Set to NaN if `NOBS` is less than 8.)
- 8 The same probability as `STAT(6)` but using a continuity correction with a normal approximation. (Set to NaN if `NOBS` is less than 8.)
- 9 Index in `FRQ` corresponding to the frequency of the observed S statistic. `STAT(9)` is not computed when there are ties.

Optional Arguments

NOBS — Number of observations. (Input)

`NOBS` must be 3 or more.

Default: `NOBS = size (X,1)`.

FRQ — Vector of length $\text{NOBS} * (\text{NOBS} - 1) / 2 + 1$ containing the frequencies of occurrence of the possible values of the statistic S , `STAT(5)`, under the null hypothesis of no relationship. (Output)

`FRQ` is not calculated if there are ties or if `NOBS` is too large (34 on many computers).

FORTRAN 90 Interface

Generic: `CALL KENDL (X, Y, FUZZ, STAT [,...])`

Specific: The specific interface names are `S_KENDL` and `D_KENDL`.

FORTRAN 77 Interface

Single: `CALL KENDL (NOBS, X, Y, FUZZ, STAT, FRQ)`

Double: The double precision name is `DKENDL`.

Example

In this example, the Kendall test is performed on a sample of size 8. The test fails to reject the null hypothesis of no correlation.

```
!                                     SPECIFICATIONS FOR PARAMETERS
      USE KENDL_INT
      USE WRRRL_INT
      USE WRRRN_INT

      REAL      FUZZ
      PARAMETER (FUZZ=0.0001)

!
      REAL      FRQ(29), STAT(9), X(8), Y(8)
      CHARACTER CLABEL(2)*10, RLABEL(9)*10
!
      DATA RLABEL/'tau(a)', 'tau(b)', 'ties(X)', 'ties(Y)', &
```

```

      'S', 'Pr(S)', 'Pr(S)-n', 'Pr(S)-na', 'IFRQ' /
!
DATA CLABEL/'Statistic', '  ' /
!
DATA X/6, 4, 7, 3, 8, 1, 5, 2/
DATA Y/7, 1, 5, 8, 6, 4, 2, 3/
!
CALL KENDL (X, Y, FUZZ, STAT, FRQ=FRQ)
!
CALL WRRRL ('STAT', STAT, RLABEL, CLABEL, FMT='(W10.6)')
CALL WRRRN ('FRQ', FRQ, 1, 29, 1, 0)
END

```

Output

```

      STAT
Statistic
tau(a)      0.1429
tau(b)      0.1429
ties(X)     0.0000
ties(Y)     0.0000
S           4.0000
Pr(S)       0.3598
Pr(S)-n    0.3103
Pr(S)-na    0.3553
IFREQ      17.0000

      FRQ
      1      2      3      4      5      6      7      8
      1.0    7.0   27.0   76.0  174.0  343.0  602.0  961.0

      9     10     11     12     13     14     15     16
1415.0  1940.0  2493.0  3017.0  3450.0  3736.0  3836.0  3736.0

      17     18     19     20     21     22     23     24
3450.0  3017.0  2493.0  1940.0  1415.0  961.0   602.0   343.0

      25     26     27     28     29
174.0   76.0   27.0   7.0    1.0

```

Comments

1. Workspace may be explicitly provided, if desired, by use of `K2NDL/DK2NDL`. The reference is:

```
CALL K2NDL (NOBS, X, Y, FUZZ, STAT, FRQ, IWK, WK, XRNK, YRNK)
```

The additional arguments are as follows:

IWK — Work vector of length `NOBS`.

WK — Work vector of length $(NOBS - 1) * (NOBS - 2) / 2 + 1$.

XRNK — Work vector of length `NOBS`.

YRNK — Work vector of length `NOBS`.

2. Informational errors

Type Code

3	4	Ties are detected in the two samples. <code>STAT(6)</code> is set to NaN (not a number) and <code>FREQ</code> is not calculated.
3	5	<code>NOBS</code> is less than 8 so the asymptotic normal probabilities are not determined. <code>STAT(7)</code> and <code>STAT(8)</code> are set to NaN (not a number).
3	6	<code>NOBS</code> is too large (34 on many computers). <code>STAT(6)</code> is set to NaN (not a number) and <code>FREQ</code> is not calculated.
4	2	All the elements of <code>X</code> are tied. The output statistics are not defined.
4	3	All the elements of <code>Y</code> are tied. The output statistics are not defined.

Description

Routine `KENDL` performs Kendall's test of the hypothesis of no correlation (independence) by calculating τ_a and τ_b (τ_b handles ties), the Kendall sum S , and associated probabilities. The frequencies of occurrence of S are also computed if the sample size (`NOBS`) is not too large.

Kendall's (1962) method is used in computing the τ statistics. Each pair (x_i, y_i) is compared with every other pair (x_j, y_j) . The Kendall S statistic is incremented if the two pairs are concordant ($(x_i > x_j \text{ and } y_i > y_j)$ or $(x_i < x_j \text{ and } y_i < y_j)$) and decremented if the pairs are discordant ($(x_i > x_j \text{ and } y_i < y_j)$ or $(x_i < x_j \text{ and } y_i > y_j)$). Ties ($x_i = x_j$ or $y_i = y_j$) are not counted. Generally, when ties exist, τ_b is a better measure of correlation than is τ_a . The untied form of the denominator is used to calculate τ_a . That is,

$$\tau_a = \frac{S}{n(n-1)/2}$$

where $n = \text{NOBS}$. Ties enter into the denominator of τ_b as follows:

$$\tau_b = \frac{S}{\sqrt{(D - T_x)(D - T_y)}}$$

where $D = n(n-1)/2$ and

$$T_x = \sum t_i(t_i - 1)/2$$

where t_i is the number of ties in the x variable with the i -th tie value. T_y is calculated in a similar manner.

For `NOBS` less than 34 (on many machines other values on machines with a different value for the largest real number that can be represented), the array `FRQ` is computed. `FRQ` contains the frequency distribution of S under the null hypothesis of independence. The probability distribution of S can be obtained directly from these frequencies by dividing each frequency by the sum of the frequencies. See routine `KENDP` ([page 387](#)) for further discussion on the use of the `FRQ` array.

For a two-sided test, if the appropriate probability p of achieving or exceeding S is small (less than $\alpha/2$, where α is the significance level of the test) or if $1 - p$ is small (less than $\alpha/2$), then the two-sided hypothesis of no correlation can be rejected. Alternatively, for small p or $1 - p$, the appropriate one-sided hypothesis can be rejected.

For $n > 7$, asymptotic normal probabilities are determined using the fact that

$$z = \frac{S}{\sqrt{\text{var}(S)}}$$

is approximately standard normal for large n . Here,

$$\begin{aligned} \text{var}(S) = & \frac{n(n-1)(2n+5) - \sum_x t_i(t_i-1)(2t_i+5) - \sum_y t_i(t_i-1)(2t_i+5)}{18} + \\ & \frac{[\sum_x t_i(t_i-1)(t_i-2)][\sum_y t_i(t_i-1)(t_i-2)]}{9n(n-1)(n-2)} + \\ & \frac{[\sum_x t_i(t_i-1)][\sum_y t_i(t_i-1)]}{2n(n-1)} \end{aligned}$$

where t_i is the number of observations in the i -th tie group for the x (or y) summation variable.

STAT(7) contains the probability associated with the z statistic while STAT(8) contains the same probability but with the value of S reduced by 1. This reduction is for “continuity correction.”

For n less than 25, these probabilities are conservative at the 1% level of significance.

KENDP

Computes the frequency distribution of the total score in Kendall’s rank correlation coefficient.

Required Arguments

NOBS — Sample size. (Input)

Must be greater than 1 and less than 34 (56 on some computers).

K — Score for which the probability is to be calculated. (Input)

K must be in the range from minus to plus $\text{NOBS} * (\text{NOBS} - 1)/2$, inclusive.

FRQ — Vector of length $\text{NOBS} * (\text{NOBS} - 1)/2 + 1$ containing the frequency distribution of possible values of K . (Output)

K will range from minus to plus $\text{NOBS} * (\text{NOBS} - 1)/2$, inclusive, in increments of 2, with frequency $\text{FRQ}(i)$, for a possible $K = 2 * (i - 1) - \text{NOBS} * (\text{NOBS} - 1)/2$, where $i = 1, 2, \dots, \text{NOBS} * (\text{NOBS} - 1)/2 + 1$.

PROB — Probability of equaling or exceeding K if the samples on which K is based are uncorrelated. (Output)

FORTRAN 90 Interface

Generic: CALL KENDP (NOBS, K, FRQ, PROB)

Specific: The specific interface names are S_KENDP and D_KENDP.

FORTRAN 77 Interface

Single: CALL KENDP (NOBS, K, FRQ, PROB)

Double: The double precision name is DKENDP.

Example

The frequency distribution S for NOBS of 4 is computed. The probability is computed for $S = 4$.

```
USE IMSL_LIBRARIES
INTEGER K, NOBS
PARAMETER (K=4, NOBS=4)
!
INTEGER I, M, NOUT
REAL FRQ(NOBS*(NOBS-1)/2+1,3), PROB, SUM
CHARACTER CLABEL(4)*10, RLABEL(1)*10
!
DATA RLABEL/'NONE'/
DATA CLABEL/' ', 'S', 'FRQ', 'pf'/
!
M = NOBS*(NOBS-1)/2 + 1
DO 10 I=1, M
  FRQ(I,1) = 2*(I-1) - NOBS*(NOBS-1)/2
10 CONTINUE
!
CALL KENDP (NOBS, K, FRQ(1:,2), PROB)
!
                                Compute the probabilities
SUM = SSUM(M,FRQ(1:,2),1)
CALL SCOPY (M, FRQ(1:,2), 1, FRQ(1:,3), 1)
CALL SSCAL (M, 1.0/SUM, FRQ(1:,3), 1)
!
                                Print results
CALL UMACH (2, NOUT)
CALL WRRRL (' ', FRQ, RLABEL, CLABEL, FMT='(W10.4)')
WRITE (NOUT,*) 'PROB = ', PROB
END
```

Output

S	FRQ	pf
-6.000	1.000	0.042
-4.000	3.000	0.125
-2.000	5.000	0.208
0.000	6.000	0.250
2.000	5.000	0.208
4.000	3.000	0.125
6.000	1.000	0.042

PROB = 0.16666667

Comments

Workspace may be explicitly provided, if desired, by use of `K2NDP/DK2NDP`.

The reference is:

```
CALL K2NDP (NOBS, K, FRQ, PROB, FWK)
```

The additional argument is:

FWK — Work vector of length $(\text{NOBS} - 1) * (\text{NOBS} - 2) / 2 + 1$.

Description

Routine `KENDP` computes the frequency distribution of the Kendall S statistic and the probability that S equals or exceeds a given value K . Routine `KENDP` requires the sample size, $n = \text{NOBS}$, on input. The frequencies reported in position i of `FRQ` correspond to

$$S = 2(i - 1) - n(n - 1)/2$$

To obtain the probability distribution of S , divide each frequency by the sum of the frequencies in `FRQ`.

The upper bound on `NOBS` that can be handled by `KENDP` depends upon the largest real number that can be represented in the computer being used (`AMACH(2)`). If this value is $1.0\text{E}+46$ or less, `NOBS` cannot be greater than 33.

Chapter 4: Analysis of Variance

Routines

4.1. General Analysis		
One-way.....	AONEW	392
One-way analysis of covariance.....	AONEC	395
Randomized block or two-way balanced design.....	ATWOB	409
Balanced incomplete block design.....	ABIBD	415
Latin square design.....	ALATN	422
Factorial.....	ANWAY	427
Balanced complete design for mixed models.....	ABALD	434
Completely random nested design.....	ANEST	449
4.2. Inference on Means and Variance Components		
Contrast estimates and sums of squares.....	CTRST	458
Simultaneous confidence intervals on differences of means.....	SCIPM	460
Student-Newman-Keuls multiple comparisons.....	SNKMC	467
CI on a difference of expected mean squares.....	CIDMS	469
4.3. Service Routine		
Reorder data for a balanced experimental design.....	ROREX	472

Usage Notes

The routines described in this chapter are for commonly-used experimental designs. Typically, responses are stored in the input vector Y in a pattern that takes advantage of the balanced design structure. Consequently, the full set of model subscripts is not needed to identify each response. The routines assume the usual pattern, which requires that the last model subscript change most rapidly, the next to last model subscript change next most rapidly, and so forth, with the first subscript changing the slowest. This pattern is referred to as lexicographical ordering.

Routines [AONEW \(page 392\)](#), [AONEC \(page 395\)](#), and [ANEST \(page 449\)](#) allow missing responses. NaN (not a number) is the missing value code used by these routines. Use routine `AMACH` to retrieve NaN. Any element of Y that is missing must be set to `AMACH(6)`. For a description of `AMACH`, see the section “Machine-Dependent Constants” in the Reference Material. Other routines described in this chapter do not allow missing responses because they generally deal with balanced designs.

As a diagnostic tool for determination of the validity of a model, routines in this chapter typically perform a test for lack of fit when n ($n > 1$) responses are available in each cell of the experimental design. Routines in Chapter 2, "Regression," are useful for analysis of generalizations of many of the models treated in this chapter. In particular, Chapter 2 provides routines for the general linear model.

AONEW

Analyzes a one-way classification model.

Required Arguments

NI— Vector of length `NGROUP` containing the number of responses for each group. (Input)

Y— Vector of length $NI(1) + NI(2) + \dots + NI(NGROUP)$ containing the responses for each group. (Input)

AOV— Vector of length 15 containing statistics relating to the analysis of variance. (Output)

I **AOV(I)**

1 Degrees of freedom for among groups

2 Degrees of freedom for within groups

3 Total (corrected) degrees of freedom

4 Sum of squares for among groups

5 Sum of squares for within groups

6 Total (corrected) sum of squares

7 Among-groups mean square

8 Within-groups mean square

9 F -statistic

10 p -value

11 R^2 (in percent)

12 Adjusted R^2 (in percent)

13 Estimated standard deviation of the error within groups

- I** **AOV(I)**
- 14 Overall mean of Y
- 15 Coefficient of variation (in percent)

Optional Arguments

NGROUP — Number of groups. (Input)

Default: `NGROUP = size (NI,1)`.

IPRINT — Printing option. (Input)

Default: `IPRINT = 0`.

IPRINT	Action
0	No printing is performed.
1	AOV is printed only.
2	STAT is printed only.
3	All printing is performed.

STAT — `NGROUP` by 4 matrix containing information concerning the groups. (Output)

Row `I` contains information pertaining to the `I`-th group. The information in the columns is as follows:

Col.	Description
1	Group number
2	Number of nonmissing observations
3	Group mean
4	Group standard deviation

LDSTAT — Leading dimension of `STAT` exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDSTAT = size (STAT , 1)`

NMISS — Number of missing values. (Output)

Elements of Y containing NaN (not a number) are omitted from the computations.

FORTRAN 90 Interface

Generic: `CALL AONEW (NI , Y , AOV [,...])`

Specific: The specific interface names are S_AONEW and D_AONEW.

FORTRAN 77 Interface

Single: CALL AONEW (NGROUP, NI, Y, IPRINT, AOV, STAT, LDSTAT, NMISS)

Double: The double precision name is DAONEW.

Example

This example computes a one-way analysis of variance for data discussed by Searle (1971, Table 5.1, pages 165–179). The responses are plant weights for 6 plants of 3 different types—3 normal, 2 off-types, and 1 aberrant. The responses are given by type of plant in the following table:

Type of Plant		
Normal	Off-Type	Aberrant
101	84	32
105	88	
94		

Note that for the group with only one response, the standard deviation is undefined and is set to NaN (not a number).

```

USE AONEW_INT
INTEGER   NGROUP, NOBS
PARAMETER (NGROUP=3, NOBS=6)
!
INTEGER   IPRINT, NI (NGROUP)
REAL     AOV (15), Y (NOBS)
!
DATA NI/3, 2, 1/
DATA Y/101.0, 105.0, 94.0, 84.0, 88.0, 32.0/
!
IPRINT = 3
CALL AONEW (NI, Y, AOV, IPRINT=IPRINT)
END

```

Output

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	98.028	96.714	4.83	84	5.751

```

* * * Analysis of Variance * * *
Source          DF      Sum of Squares      Mean Square      Overall F      Prob. of Larger F
Among Groups    2          3480          1740.0          74.571          0.0028
Within Groups   3           70           23.3
Corrected Total 5          3550

```


Group Statistics			
Group	N	Mean	Standard Deviation
1	3	100	5.568
2	2	86	2.828
3	1	32	NaN

Description

Routine `AONEW` performs an analysis of variance of responses from a one-way classification design. The model is

$$y_{ij} = \mu_i + \varepsilon_{ij} \quad i = 1, 2, \dots, k; j = 1, 2, \dots, n_i$$

where the observed value of y_{ij} constitutes the j -th response in the i -th group, μ_i denotes the population mean for the i -th group, and the ε_{ij} 's are errors that are identically and independently distributed normal with mean zero and variance σ^2 . `AONEW` requires the y_{ij} 's as input into a single vector `Y` with responses in each group occupying contiguous locations. The analysis of variance table is computed along with the group sample means and standard deviations. A discussion of formulas and interpretations for the one-way analysis of variance problem appears in most elementary statistics texts, e.g., Snedecor and Cochran (1967, Chapter 10).

AONEC

Analyzes a one-way classification model with covariates.

Required Arguments

NI — Vector of length `NGROUP` containing the number of responses for each group. (Input)

XY — (`NI(1) + NI(2) + ... + NI(NGROUP)`) by (`NCOV + 1`) matrix containing the data for each covariate and the response variable. (Input)

Data for each group must appear in contiguous rows of `XY`, and the responses must appear in the last column.

AOV — Vector of length 15 that contains statistics relating to the analysis of variance for the model assuming parallelism. (Output)

I **AOV(I)**

1 Degrees of freedom for model (groups + covariates)

2 Degrees of freedom for error

3 Total (corrected) degrees of freedom

4 Sum of squares for model

5 Sum of squares for error

I	AOV(I)
6	Total (corrected) sum of squares
7	Model mean square
8	Error mean square
9	<i>F</i> -statistic
10	<i>p</i> -value
11	R^2 (in percent)
12	Adjusted R^2 (in percent)
13	Estimate of the error standard deviation
14	Overall response mean
15	Coefficient of variation (in percent)

Optional Arguments

NGROUP — Number of groups. (Input)
Default: `NGROUP = size (NI,1)`.

NCOV — Number of covariates. (Input)
Default: `NCOV = size (XY,2) - 1`.

LDXY — Leading dimension of `XY` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDXY = size (XY,1)`.

ITEST — Indicator for test for parallelism (equal covariate coefficients across groups). (Input)
Default: `ITEST = 0`.

ITEST	Action
--------------	---------------

0	Test for parallelism is not performed.
---	--

1	Test for parallelism is performed.
---	------------------------------------

IPRINT — Printing option. (Input)
Default: `IPRINT = 0`.

I	PRINT	Action
0		No printing is performed.
1		Printing for model assuming parallelism is performed.
2		Printing for separate regression models for each group is performed as well as for the model assuming parallelism.

COEF — $NGROUP + NCOV$ by 4 matrix containing statistics relating to the regression coefficients for the model assuming parallelism. (Output)
 Each row corresponds to a coefficient in the model. For $I = 1, 2, \dots, NGROUP$, row I is for the Y intercept for the I -th group. The remaining $NCOV$ rows are for the covariate coefficients. The statistics in the columns are as follows:

Col.	Description
1	Coefficient estimate
2	Estimated standard error of the estimate
3	t -statistic
4	p -value

LDCOEF — Leading dimension of **COEF** exactly as specified in the dimension statement in the calling program. (Input)
 Default: $LDCOEF = \text{size}(\text{COEF}, 1)$.

R — $NGROUP + NCOV$ by $NGROUP + NCOV$ upper triangular matrix containing the R matrix from the QR decomposition. (Output)
 The R matrix is from the regression assuming parallelism.

LDR — Leading dimension of R exactly as specified in the dimension statement in the calling program. (Input)
 Default: $LDR = \text{size}(R, 1)$.

PTSS — Vector of length 8 containing statistics relating to the partial sums of squares for groups and for covariates in the model assuming parallelism. (Output)

I	PTSS(I)
1	Degrees of freedom for groups after covariates
2	Degrees of freedom for covariates after groups
3	Sum of squares for groups after covariates

- I** **PTSS(I)**
- 4 Sum of squares for covariates after groups
- 5 *F* -statistic for groups
- 6 *F* -statistic for covariates
- 7 *p*-value for groups
- 8 *p*-value for covariates

TESTPL — Vector of length 10 containing statistics relating to the test for parallelism. (Output if **ITEST** = 1)
 If **ITEST** = 0, **TESTPL** is not referenced and can be a vector of length one.

- I** **TESTPL(I)**
- 1 Extra degrees of freedom for model not assuming parallelism
- 2 Degrees of freedom for error for model not assuming parallelism
- 3 Degrees of freedom for error for model assuming parallelism
- 4 Extra sum of squares for model not assuming parallelism
- 5 Sum of squares for error for model not assuming parallelism
- 6 Sum of squares for error for model assuming parallelism
- 7 Mean square for **TESTPL(1)**
- 8 Mean square for **TESTPL(2)**
- 9 *F* -statistic
- 10 *p*-value

XYMEAN — **NGROUP** + 1 by **NCOV** + 3 matrix containing means. (Output)
 Each row for **I** = 1, 2, ..., **NGROUP** corresponds to a group. Row **NGROUP** + 1 contains overall statistics. The statistics in the columns are as follows:

Column	Description
1	Number of nonmissing cases
2 thru NCOV + 1	Covariate means

Column	Description
NCOV + 2	Response mean
NCOV + 3	Adjusted mean assuming parallelism

LDXYME — Leading dimension of XYMEAN exactly as specified in the dimension statement in the calling program. (Input)
 Default: LDXYME = size (XYMEAN,1).

COVM — NGROUP by NGROUP matrix containing the estimated variance-covariance matrix of the adjusted group means in the model assuming parallelism. (Output)

LDCOVM — Leading dimension of COVM exactly as specified in the dimension statement in the calling program. (Input)
 Default: LDCOVM = size (COVM,1).

COVB — NGROUP + NCOV by NGROUP + NCOV matrix containing the estimated variance-covariance matrix of the estimated coefficients in the model assuming parallelism. (Output)
 If R is not needed, R and COVB can occupy the same storage locations.

LDCOVB — Leading dimension of COVB exactly as specified in the dimension statement in the calling program. (Input)
 Default: LDCOVB = size (COVB,1).

NRMISS — Number of rows of XY that contain any missing values. (Output)
 Rows of XY containing NaN (not a number) are omitted from computations.

FORTRAN 90 Interface

Generic: CALL AONEC (NI, XY, AOV [, ...])

Specific: The specific interface names are S_AONEC and D_AONEC.

FORTRAN 77 Interface

Single: CALL AONEC (NGROUP, NI, NCOV, XY, LDXY, ITEST, IPRINT, COEF, LDCOEF, R, LDR, AOV, PTSS, TESTPL, XYMEAN, LDXYME, COVM, LDCOVM, COVB, LDCOVB, NRMISS)

Double: The double precision name is DAONEC.

Example 1

This example fits a one-way analysis of covariance model assuming parallelism using data discussed by Snedecor and Cochran (Table 14.6.1, pages 432–436). The responses are

concentrations of cholesterol (in mg/100 ml) in the blood of two groups of women: women from Iowa and women from Nebraska. Age of a woman is the single covariate. The cholesterol concentrations and ages of the women according to state are shown in the following table. (There are 11 Iowa women and 19 Nebraska women in the study. Only the first 5 women from each state are shown here.)

Iowa		Nebraska	
Age	Cholesterol	Age	Cholesterol
46	181	18	137
52	228	44	173
39	182	33	177
65	249	78	241
54	259	51	225

There is no evidence from the data to indicate that the regression lines for cholesterol concentration as a function of age are not parallel for Iowa and Nebraska women (p -value is 0.5425). The parallel line model suggests that Nebraska women may have higher cholesterol concentrations than Iowa women. The cholesterol concentrations (adjusted for age) are 195.5 for Iowa women versus 224.2 for Nebraska women. The difference is 28.7 with an estimated standard error of

$$\sqrt{170.4 + 97.4 - 2(2.9)} = 16.1$$

```

USE AONEC_INT
INTEGER      LDXY, NCOV,  NGROUP, NOBS
PARAMETER   (NCOV=1, NGROUP=2, NOBS=30, LDXY=NOBS)
!
INTEGER      IPRINT, ITEST, NI (NGROUP)
REAL         AOV(15), XY (LDXY, NCOV+1)
!
DATA NI/11, 19/
DATA XY/46.0, 52.0, 39.0, 65.0, 54.0, 33.0, 49.0, 76.0, 71.0, &
  41.0, 58.0, 18.0, 44.0, 33.0, 78.0, 51.0, 43.0, 44.0, 58.0, &
  63.0, 19.0, 42.0, 30.0, 47.0, 58.0, 70.0, 67.0, 31.0, 21.0, &
  56.0, 181.0, 228.0, 182.0, 249.0, 259.0, 201.0, 121.0, &
  339.0, 224.0, 112.0, 189.0, 137.0, 173.0, 177.0, 241.0, &
  225.0, 223.0, 190.0, 257.0, 337.0, 189.0, 214.0, 140.0, &
  196.0, 262.0, 261.0, 356.0, 159.0, 191.0, 197.0/
!
ITEST = 1
IPRINT = 2
CALL AONEC (NI, XY, AOV, ITEST=ITEST, IPRINT=IPRINT)
!
END

```

Output

SEPARATE REGRESSION FOR GROUP 1

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	47.120	41.245	48.9	207.7	23.54

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Model	1	19177.2	19177.2	8.020	0.0197
Error	9	21521.0	2391.2		
Corrected Total	10	40698.2			

Inference on Coefficients

Coef.	Estimate	Standard Error	t-statistic	Prob. of Larger t
1	35.81	62.47	0.573	0.5805
2	3.24	1.14	2.832	0.0197

SEPARATE REGRESSION FOR GROUP 2

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	56.812	54.272	39.76	217.1	18.31

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Model	1	35351.9	35351.9	22.363	0.0002
Error	17	26873.9	1580.8		
Corrected Total	18	62225.8			

Inference on Coefficients

Coef.	Estimate	Standard Error	t-statistic	Prob. of Larger t
1	101.3	26.13	3.876	0.0012
2	2.5	0.53	4.729	0.0002

SAME REGRESSION FOR ALL GROUPS

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	47.303	45.421	44.14	213.7	20.66

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Model	1	48976.3	48976.3	25.134	0.0000
Error	28	54560.4	1948.6		
Corrected Total	29	103536.7			

Inference on Coefficients

Coef.	Estimate	Standard Error	t-statistic	Prob. of Larger t
1	91.57	25.65	3.570	0.0013
2	2.51	0.50	5.013	0.0000

REGRESSION ASSUMING PARALLELISM

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	52.573	49.060	42.65	213.7	19.96

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Model	2	54432.8	27216.4	14.965	0.0000
Error	27	49103.9	1818.7		
Corrected Total	29	103536.7			

Partial Sums of Squares

Source	DF	Sum of Squares	F	Prob. of Larger F
Groups after Covariates	1	5456.5	3.000	0.0947
Covariates after Groups	1	53820.1	29.593	0.0000

R Matrix

	1	2	3
1	3.3	0.0	176.1
2		4.4	200.3
3			86.0

Inference on Coefficients

Coef.	Estimate	Standard Error	t-statistic	Prob. of Larger t
1	64.49	29.3	2.201	0.0365
2	93.14	24.8	3.756	0.0008
3	2.70	0.5	5.440	0.0000

Test for Parallelism

Source	DF	Sum of Squares	Mean Square	F	Prob. of Larger F
Extra due to nonparallelism	1	709.0	709.0	0.381	0.5425
Error assuming nonparallelism	26	48394.9	1861.3		
Error assuming parallelism	27	49103.9			

XYMEAN

	1	2	3	4
1	11	53.09	207.7	195.5
2	19	45.95	217.1	224.2
3	30	48.57	213.7	213.7

Variance-Covariance Matrix of the Adjusted Group Means

	1	2
1	170.4	-2.9
2		97.4

Variance-Covariance Matrix of the Estimated Coefficients

	1	2	3
1	858.6	600.0	-13.1
2		615.0	-11.3
3			0.2

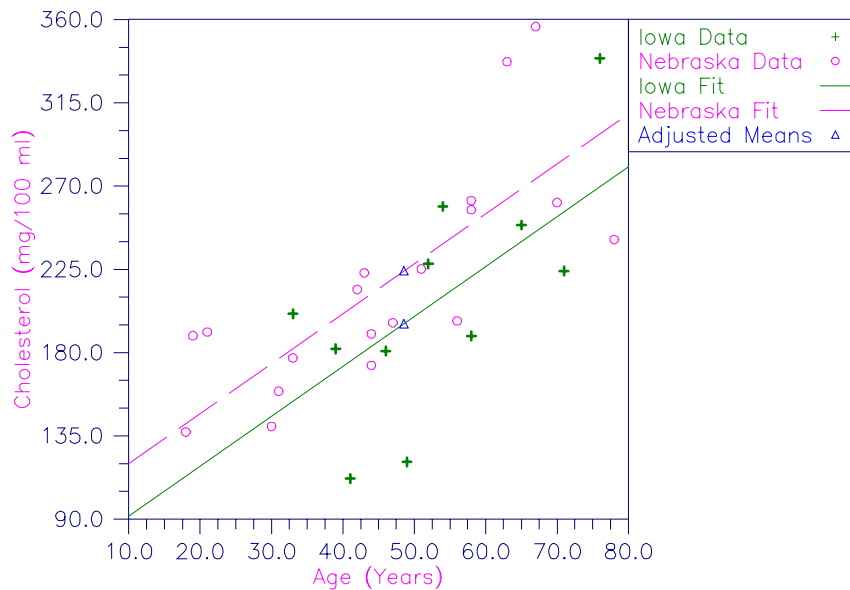


Figure 4-1 Plot of Cholesterol Concentrations and Fitted Parallel Lines by State

Comments

Workspace may be explicitly provided, if desired, by use of A2NEC/DA2NEC. The reference is:

```
CALL A2NEC (NGROUP, NI, NCOV, XY, LDXY, ITEST, IPRINT, COEF, LDCOEF,
R, LDR, AOV, PTSS, TESTPL, XYMEAN, LDXYME, COVM, LDCOVM, COVB,
LDCOVB, NRMISS, WK)
```

The additional argument is:

WK — Work vector of length $4 * (NGROUP + NCOV + 1)$.

Description

Routine AONEC performs analyses for models that combine the features of a one-way analysis of variance model with that of a multiple linear regression model. The basic one-way analysis of covariance model is

$$y_{ij} = \beta_{0i} + \beta_1 x_{ij1} + \beta_2 x_{ij2} + \dots + \beta_m x_{ijm} + \varepsilon_{ij} \quad i = 1, 2, \dots, k; \quad j = 1, 2, \dots, n_i$$

where the observed value of y_{ij} constitutes the j -th response in the i -th group, β_{0i} denotes the y intercept for the regression function for the i -th group, $\beta_1, \beta_2, \dots, \beta_m$ are the regression coefficients for the covariates, and the ε_{ij} 's are independently distributed normal errors with mean zero and variance σ^2 . This model allows the regression function for each group to have different intercepts. However, the remaining m regression coefficients are the same for each group, i.e., the regression functions are parallel. Often in practice, the regression functions are not parallel. In addition to estimates for the model assuming parallelism, AONEC computes estimates and summary statistics for the separate regressions for each group. With `IPRINT = 2`, the estimates and summary statistics for each group are printed. If `ITEST = 1`, a test for parallelism is performed.

AONEC requires $(x_{ij1}, x_{ij2}, \dots, x_{ijk}, y_{ij})$ as input into a single data matrix `XY` with the data for each group occupying contiguous rows of `XY`.

Estimates for the β_{0i} 's and $\beta_1, \beta_2, \dots, \beta_m$ in the model assuming parallelism are computed and stored in `COEF`. Summary statistics are also computed for this model. The adjusted group means (stored in column $m + 3$ of `XYMEAN`) are given by

$$\hat{\beta}_{0i} + \hat{\beta}_1 \bar{x}_1 + \hat{\beta}_2 \bar{x}_2 + \dots + \hat{\beta}_m \bar{x}_m$$

The estimated covariance between the i_1 -th and i_2 -th adjusted group mean is given by

$$v_{i_1 i_2} + \sum_{r=1}^m \sum_{s=1}^m \bar{x}_r v_{k+r, k+s} \bar{x}_s + \sum_{r=1}^m \bar{x}_r v_{i_1, k+r} + \sum_{r=1}^m \bar{x}_r v_{i_2, k+r}$$

where v_{pq} is the pq -th entry in `COVB` and is the estimated covariance between the p -th and q -th estimated coefficients in the regression function.

The design of AONEC can be used with routines described in Chapter 2, "Regression." For example, confidence intervals and diagnostics for the individual cases can be computed by using the output matrices `R` and `COEF` as input into regression routines for case analysis.

A discussion of formulas and interpretations for the one-way analysis of covariance problem appears in most elementary statistics texts, e.g., Snedecor and Cochran (1967, Chapter 14).

Additional Example

Example 2

This example fits a one-way analysis of covariance model and performs a test for parallelism using data discussed by Snedecor and Cochran (1967, Table 14.8.1, pages 438–443). The responses are weight gains (in pounds per day) of 40 pigs for 4 groups of pigs under varying

treatments. Two covariates—initial age (in days) and initial weight (in pounds)—are used. For each treatment, there are 10 pigs. Only the first 5 pigs from each treatment are shown here.

Treatment 1			Treatment 2			Treatment 3			Treatment 4		
Age	Wt.	Gain	Age	Wt.	Gain	Age	Wt.	Gain	Age	Wt.	Gain
78	61	1.40	78	74	1.61	78	80	1.67	77	62	1.40
90	59	1.79	99	75	1.31	83	61	1.41	71	55	1.47
94	76	1.72	80	64	1.12	79	62	1.73	78	62	1.37
71	50	1.47	75	48	1.35	70	47	1.23	70	43	1.15
99	61	1.26	94	62	1.29	85	59	1.49	95	57	1.22

```

USE AONEC_INT
INTEGER   LDXY, NCOV, NGROUP, NOBS
PARAMETER (NCOV=2, NGROUP=4, NOBS=40, LDXY=NOBS)
!
INTEGER   IPRINT, ITEST, NI (NGROUP)
REAL      AOV(15), XY (LDXY,NCOV+1)
!
DATA NI/10, 10, 10, 10/
DATA XY/78.0, 90.0, 94.0, 71.0, 99.0, 80.0, 83.0, 75.0, 62.0, &
      67.0, 78.0, 99.0, 80.0, 75.0, 94.0, 91.0, 75.0, 63.0, 62.0, &
      67.0, 78.0, 83.0, 79.0, 70.0, 85.0, 83.0, 71.0, 66.0, 67.0, &
      67.0, 77.0, 71.0, 78.0, 70.0, 95.0, 96.0, 71.0, 63.0, 62.0, &
      67.0, 61.0, 59.0, 76.0, 50.0, 61.0, 54.0, 57.0, 45.0, 41.0, &
      40.0, 74.0, 75.0, 64.0, 48.0, 62.0, 42.0, 52.0, 43.0, 50.0, &
      40.0, 80.0, 61.0, 62.0, 47.0, 59.0, 42.0, 47.0, 42.0, 40.0, &
      40.0, 62.0, 55.0, 62.0, 43.0, 57.0, 51.0, 41.0, 40.0, 45.0, &
      39.0, 1.40, 1.79, 1.72, 1.47, 1.26, 1.28, 1.34, 1.55, 1.57, &
      1.26, 1.61, 1.31, 1.12, 1.35, 1.29, 1.24, 1.29, 1.43, 1.29, &
      1.26, 1.67, 1.41, 1.73, 1.23, 1.49, 1.22, 1.39, 1.39, 1.56, &
      1.36, 1.40, 1.47, 1.37, 1.15, 1.22, 1.48, 1.31, 1.27, 1.22, &
      1.36/
!
      ITEST = 1
      IPRINT = 2
      CALL AONEC (NI, XY, AOV, ITEST=ITEST, IPRINT=IPRINT)
!
END

```

Output

SEPARATE REGRESSION FOR GROUP 1

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Coefficient of Mean Var. (percent)
Y	13.271	0.000	0.2013	1.464 13.75

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Model	2	0.0434	0.02170	0.536	0.6075

Error	7	0.2836	0.04052
Corrected Total	9	0.3270	

Inference on Coefficients

Coef.	Estimate	Standard Error	t-statistic	Prob. of Larger t
1	1.357	0.4639	2.925	0.0222
2	-0.006	0.0105	-0.572	0.5849
3	0.011	0.0114	0.948	0.3749

SEPARATE REGRESSION FOR GROUP 2

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	21.989	0.000	0.1292	1.319	9.799

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Model	2	0.0330	0.01648	0.987	0.4193
Error	7	0.1169	0.01670		
Corrected Total	9	0.1499			

Inference on Coefficients

Coef.	Estimate	Standard Error	t-statistic	Prob. of Larger t
1	1.401	0.2694	5.199	0.0013
2	-0.005	0.0040	-1.164	0.2825
3	0.005	0.0040	1.301	0.2343

SEPARATE REGRESSION FOR GROUP 3

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	49.246	34.745	0.1369	1.445	9.473

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Model	2	0.1273	0.06364	3.396	0.0931
Error	7	0.1312	0.01874		
Corrected Total	9	0.2584			

Inference on Coefficients

Coef.	Estimate	Standard Error	t-statistic	Prob. of Larger t
1	1.452	0.4709	3.082	0.0178
2	-0.008	0.0075	-1.017	0.3429
3	0.011	0.0043	2.544	0.0384

SEPARATE REGRESSION FOR GROUP 4

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	17.076	0.000	0.1141	1.325	8.609

* * * Analysis of Variance * * *						
Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F	
Model	2	0.0188	0.00938	0.721	0.5193	
Error	7	0.0911	0.01301			
Corrected Total	9	0.1098				

Inference on Coefficients					
Coef.	Estimate	Standard Error	t-statistic	Prob. of Larger t	
1	1.044	0.2574	4.055	0.0048	
2	0.001	0.0038	0.251	0.8094	
3	0.004	0.0051	0.833	0.4324	

SAME REGRESSION FOR ALL GROUPS

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	17.724	13.277	0.1508	1.388	10.86

* * * Analysis of Variance * * *						
Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F	
Model	2	0.181	0.09064	3.985	0.0271	
Error	37	0.842	0.02274			
Corrected Total	39	1.023				

Inference on Coefficients					
Coef.	Estimate	Standard Error	t-statistic	Prob. of Larger t	
1	1.251	0.1708	7.327	0.0000	
2	-0.003	0.0028	-1.178	0.2464	
3	0.007	0.0027	2.743	0.0093	

REGRESSION ASSUMING PARALLELISM

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	34.467	24.829	0.1404	1.388	10.11

* * * Analysis of Variance * * *						
Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F	
Model	5	0.353	0.07050	3.576	0.0105	
Error	34	0.670	0.01971			
Corrected Total	39	1.023				

Partial Sums of Squares					
Source	DF	Sum of Squares	F	Prob. of Larger F	
Groups after Covariates	3	0.1712	2.895	0.0493	
Covariates after Groups	2	0.1750	4.438	0.0194	

R Matrix						
	1	2	3	4	5	6
1	3.2	0.0	0.0	0.0	252.7	172.0
2		3.2	0.0	0.0	247.9	173.9
3			3.2	0.0	236.9	164.4
4				3.2	237.2	156.5
5					67.4	42.7
6						55.3

Inference on Coefficients					
Coef.	Estimate	Standard Error	t-statistic	Prob. of Larger t	
1	1.337	0.1724	7.751	0.0000	
2	1.182	0.1697	6.965	0.0000	
3	1.318	0.1626	8.109	0.0000	
4	1.217	0.1624	7.493	0.0000	
5	-0.003	0.0026	-1.314	0.1978	
6	0.007	0.0025	2.919	0.0062	

Test for Parallelism						
Source	DF	Sum of Squares	Mean Square	F	Prob. of Larger F	
Extra due to nonparallelism	6	0.0474	0.00790	0.355	0.9007	
Error assuming nonparallelism	28	0.6228	0.02224			
Error assuming parallelism	34	0.6703				

XYMEAN					
	1	2	3	4	5
1	10	79.90	54.40	1.464	1.461
2	10	78.40	55.00	1.319	1.307
3	10	74.90	52.00	1.445	1.443
4	10	75.00	49.50	1.325	1.342
5	40	77.05	52.72	1.388	1.388

Variance-Covariance Matrix of the Adjusted Group Means					
	1	2	3	4	
1	0.002007	0.000016	-0.000027	-0.000024	
2		0.001992	-0.000007	-0.000030	
3			0.001994	0.000011	
4				0.002014	

Variance-Covariance Matrix of the Estimated Coefficients						
	1	2	3	4	5	6
1	0.02974	0.02729	0.02605	0.02602	-0.00033	-0.00002
2		0.02880	0.02561	0.02556	-0.00032	-0.00003
3			0.02642	0.02441	-0.00031	-0.00003
4				0.02638	-0.00032	-0.00001
5					0.00001	0.00000
6						0.00001

ATWOB

Analyzes a randomized block design or a two-way balanced design.

Required Arguments

NBLK — Number of blocks. (Input)

NTRT — Number of treatments. (Input)

NRESP — Number of repeated responses within each block-treatment combination. (Input)

Y — Vector of length $NBLK * NTRT * NRESP$ containing the responses. (Input)

The first *NRESP* elements of *Y* contain the responses for block one, treatment one, the second *NRESP* elements of *Y* contain the responses for block one, treatment two; ...; the last *NRESP* elements of *Y* contain the responses for block *NBLK*, treatment *NTRT*.

AOV — Vector of length 15 containing statistics relating to the analysis of variance. (Output)

I	AOV(I)
1	Degrees of freedom for the model (blocks and treatments)
2	Degrees of freedom for error (interaction is pooled with the within-cell error)
3	Total (corrected) degrees of freedom
4	Sum of squares for the model (blocks and treatments)
5	Sum of squares for error (interaction is pooled with the within-cell error)
6	Total (corrected) sum of squares
7	Model mean square
8	Error mean square
9	<i>F</i> -statistic
10	<i>p</i> -value
11	R^2 (in percent)
12	Adjusted R^2 (in percent)
13	Estimated standard deviation of the model error

- I** **AOV(I)**
- 14 Overall mean of Y
- 15 Coefficient of variation (in percent)

Optional Arguments

IPRINT — Printing option. (Input)

Default: `IPRINT = 0`.

IPRINT Action

- 0 No printing is performed.
- 1 Print `AOV`, `EFSS`, and `TESTLF` (if `NRESP > 1`).
- 2 Print `YMEANS` only.
- 3 All printing is performed.

EFSS — Vector of length 8 containing statistics relating to the sums of squares for the effects in the model. (Output)

Elements of `EFSS` are described as follows:

Elem. Description

- 1, 2 Degrees of freedom for blocks and treatments, respectively
- 3, 4 Sum of squares for blocks and treatments, respectively
- 5, 6 F -statistics for blocks and treatments, respectively. F -statistics are computed using `AOV(8)` as the estimated error variance.
- 7, 8 p -values associated with the F -statistics

TESTLF — Vector of length 10 containing statistics relating to the test for lack of fit of the two-way model without interaction. (Output if `NRESP > 1`)

If `NRESP = 1`, `TESTLF` is not referenced and can be a vector of length one. Elements of `TESTLF` are described as follows:

Elem. Description

- 1 Degrees of freedom for interaction
- 2 Degrees of freedom for within-cell error
- 3 Degrees of freedom for error (`TESTLF(1) + TESTLF(2)`)

- 4 Sum of squares for interaction
- 5 Sum of squares for within-cell error
- 6 Sum of squares for error
- 7 Mean square for interaction
- 8 Mean square for within-cell error
- 9 *F*-statistic
- 10 *p*-value

YMEANS — Vector of length $NBLK + NTRT + NBLK * NTRT$ containing the block means, treatment means and block-by-treatment means, respectively. (Output)

FORTRAN 90 Interface

Generic: `CALL ATWOB (NBLK, NTRT, NRESP, Y, AOV [,...])`

Specific: The specific interface names are `S_ATWOB` and `D_ATWOB`.

FORTRAN 77 Interface

Single: `CALL ATWOB (NBLK, NTRT, NRESP, Y, IPRINT, AOV, EFSS, TESTLF, YMEANS)`

Double: The double precision name is `DATWOB`.

Example 1

This example performs an analysis for a randomized block design using data discussed by Neter and Wasserman (1974, Table 23.2, pages 725–730). Fifteen businessmen were shown one of three methods for quantifying the maximum risk premium they would be willing to pay to avoid uncertainty. The responses are a stated degree of confidence, on a scale of 0 (no confidence) to 20 (highest confidence). The fifteen businessmen were grouped into five blocks by age. The three businessmen in each block were randomly assigned to a rating method. The data are given in the following table:

Block	Confidence Rating		
	Method 1	Method 2	Method 3
1	1	5	8
2	2	8	14
3	7	9	16

	Confidence Rating		
Block	Method 1	Method 2	Method 3
4	6	13	18
5	12	14	17

```

USE ATWOB_INT
INTEGER    NBLK, NRESP, NTRT
PARAMETER (NBLK=5, NRESP=1, NTRT=3)
!
INTEGER    IPRINT
REAL      AOV(15), Y(NBLK*NTRT*NRESP)
!
DATA Y/1.0, 5.0, 8.0, 2.0, 8.0, 14.0, 7.0, 9.0, 16.0, 6.0, 13.0, &
      18.0, 12.0, 14.0, 17.0/
!
IPRINT = 3
CALL ATWOB (NBLK, NTRT, NRESP, Y, AOV, IPRINT=IPRINT)
END

```

Output

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	94.003	89.506	1.727	10	17.27

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Model	6	374.1	62.36	20.901	0.0002
Error	8	23.9	2.98		
Corrected Total	14	398.0			

* * * Decomposition of Variation Attributable to the Model * * *

Source	DF	Sum of Squares	F	Prob. of Larger F
Blocks	4	171.3	14.358	0.0010
Treatment	2	202.8	33.989	0.0001

* * * Block Means * * *

Block	Mean (N=3)
1	4.6667
2	8.0000
3	10.6667
4	12.3333
5	14.3333

* * * Treatment Means * * *

Treatment	Mean (N=5)
1	5.6000

```

      2      9.8000
      3     14.6000

```

```

* * * Cell Means * * *
Block Treatment Mean (N=1)
  1      1      1.0000
  1      2      5.0000
  1      3      8.0000
  2      1      2.0000
  2      2      8.0000
  2      3     14.0000
  3      1      7.0000
  3      2      9.0000
  3      3     16.0000
  4      1      6.0000
  4      2     13.0000
  4      3     18.0000
  5      1     12.0000
  5      2     14.0000
  5      3     17.0000

```

Description

Routine `ATWOB` performs an analysis for a two-way classification design with balanced data. For balanced data, there must be an equal number of responses in each cell of the two-way layout. The basic model is the same as for the randomized block design. The block and treatment effects are additive, i.e., there are no interactions. The model is

$$y_{ijk} = \mu + \alpha_i + \beta_j + \varepsilon_{ij} \quad i = 1, 2, \dots, n_1; j = 1, 2, \dots, n_2; k = 1, 2, \dots, n_3$$

where the observed value of y_{ijk} constitutes the k -th response in the ij -th cell of the two-way layout, $\mu + \alpha_i + \beta_j$ is the population mean for the ij -th cell, and the ε_{ijk} 's are identically and independently distributed normal errors with mean zero and variance σ^2 . This model assumes that the effects for the two factors are additive. Often in practice, there are interactions between the two factors. For this reason, in addition to summary statistics for the additive model, `ATWOB` computes a test for nonadditivity (lack of fit). The test used here requires at least two responses in each cell. Tests for nonadditivity with one response per cell are given by Tukey (1949) and Mandel (1961). Tukey's test is discussed by Snedecor and Cochran (1967, pages 331–334).

The routine `ATWOB` requires y_{ijk} 's as input into a single vector `Y` with the data for each cell occupying contiguous elements. The cells must be in standard order, i.e., (1, 1), (1, 2), ..., (1, n_2), (2, 1), (2, 2), ..., (2, n_2), ..., (n_1 , 1), (n_1 , 2), ..., (n_1 , n_2):

Additional Example

Example 2

This example fits an additive two-way analysis of variance model and performs a test for nonadditivity (lack of fit) using data discussed by Kirk (1982, Table 8.3-1, pages 354–359). The data for the two-way layout is given in the following table:

	BLOCK		
TREATMENT	1	2	3
1	24, 33, 37, 29, 42	44, 36, 25, 27, 43	38, 29, 28, 47, 48
2	30, 21, 39, 26, 34	35, 40, 27, 31, 22	26, 27, 36, 46, 45
3	21, 18, 10, 31, 20	41, 39, 50, 36, 34	42, 52, 53, 49, 64

```

USE ATWOB_INT
INTEGER    NBLK, NRESP, NTRT
PARAMETER  (NBLK=3, NRESP=5, NTRT=3)
!
INTEGER    IPRINT
REAL       AOV(15), Y(NBLK*NTRT*NRESP)
!
DATA Y/24.0, 33.0, 37.0, 29.0, 42.0, 30.0, 21.0, 39.0, 26.0, &
      34.0, 21.0, 18.0, 10.0, 31.0, 20.0, 44.0, 36.0, 25.0, 27.0, &
      43.0, 35.0, 40.0, 27.0, 31.0, 22.0, 41.0, 39.0, 50.0, 36.0, &
      34.0, 38.0, 29.0, 28.0, 47.0, 48.0, 26.0, 27.0, 36.0, 46.0, &
      45.0, 42.0, 52.0, 53.0, 49.0, 64.0/
!
IPRINT = 3
CALL ATWOB (NBLK, NTRT, NRESP, Y, AOV, IPRINT=IPRINT)
END

```

Output

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Coefficient of Mean Var. (percent)
Y	33.206	26.526	9.336	35 26.68

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Model	4	1733.3	433.3	4.971	0.0024
Error	40	3486.7	87.2		
Corrected Total	44	5220.0			

* * * Decomposition of Variation Attributable to the Model * * *

Source	DF	Sum of Squares	F	Prob. of Larger F
Blocks	2	1543.3	8.853	0.0007
Treatment	2	190.0	1.090	0.3460

* * * Test for Lack of Fit * * *

Source	DF	Sum of Squares	Mean Square	F	Prob. of Larger F
Interaction	4	1236.7	309.2	4.947	0.0028
Within cell	36	2250.0	62.5		
Error	40	3486.7			

* * * Block Means * * *

Block	Mean (N=3)
1	27.6667

```

      2      35.3333
      3      42.0000

* * * Treatment Means * * *
  Treatment  Mean (N=3)
      1      35.3333
      2      32.3333
      3      37.3333

* * * Cell Means * * *
Block  Treatment  Mean (N=5)
  1      1      33.0000
  1      2      30.0000
  1      3      20.0000
  2      1      35.0000
  2      2      31.0000
  2      3      40.0000
  3      1      38.0000
  3      2      36.0000
  3      3      52.0000

```

ABIBD

Analyzes a balanced incomplete block design or a balanced lattice design.

Required Arguments

NTRT — Number of treatments. (Input)

NREP — Number of replications. (Input)

NBLK — Number of blocks. (Input)

NTBLK — Number of treatments within each block. (Input)

NRESP — Number of responses within each treatment-block combination. (Input)

Y — Vector of length $NBLK * NTBLK * NRESP$ containing the responses. (Input)

The first *NRESP* elements of *Y* contain the responses for the first treatment in the first block in the first replicate. The second *NRESP* elements of *Y* contain the responses for the second treatment in the first block in the first replicate. ... The *NTBLK*-th *NRESP* elements of *Y* contain the responses for the *NTBLK*-th treatment in the first block in the first replicate. ... The last *NRESP* elements of *Y* contain the responses for the *NTBLK*-th treatment in the *NBLK*-th block in the *NREP*-th replicate.

ITRT — Vector of length $NBLK * NTBLK$ containing the treatment numbers for the responses in *Y*. (Input)

The treatment numbers must be from the set 1, 2, ..., *NTRT*. For $I = 1, 2, \dots, NBLK * NTBLK$, element numbers $(I - 1) * NRESP + 1$ thru $(I - 1) * NRESP + NRESP$ of *Y* correspond to treatment number *ITRT*(*I*).

AOV — Vector of length 15 containing statistics relating to the analysis of variance.
(Output)

I	AOV(I)
1	Degrees of freedom for the model (replicates, blocks, and treatments)
2	Degrees of freedom for error (experimental error pooled with the within-cell error)
3	Total (corrected) degrees of freedom
4	Sum of squares for the model (replicates, blocks, and treatments)
5	Sum of squares for error (experimental error pooled with the within-cell error)
6	Total (corrected) sum of squares
7	Model mean square
8	Error mean square
9	F -statistic
10	p -value
11	R^2 (in percent)
12	Adjusted R^2 (in percent)
13	Estimated standard deviation of the model error
14	Overall mean of Y
15	Coefficient of variation (in percent)

Optional Arguments

INTER — Interblock analysis option. (Input)
Default: `INTER = 0`.

INTER Means

- | | |
|---|--|
| 0 | Intrablock analysis is requested. (Blocks are fixed effects.) |
| 1 | Interblock analysis is requested. (Blocks are random effects.) |

IPRINT — Printing option. (Input)

Default: `IPRINT = 0`.

IPRINT Action

- 0 No printing is performed.
- 1 Print `AOV`, `SQSS`, and `TESTLF` (if `NRESP > 1`).
- 2 Print `YMEANS` only.
- 3 All printing is performed.

SQSS — Vector of length 12 containing statistics relating to the sequential sum of squares for the model. (Output)

Elem. Description

- 1, 2, 3 Degrees of freedom for replicates, blocks within replicates, and treatments (adjusted), respectively
- 4, 5, 6 Sum of squares for replicates, blocks within replicates, and treatments (adjusted), respectively
- 7, 8, 9 *F*-statistics for replicates, blocks, and treatments, respectively, computed using `AOV(8)` as the estimated error variance
- 10–12 *p*-values associated with the *F*-statistics

SSALT — Vector of length 2 containing an alternative partitioning of the model sum of squares. (Output)

`SSALT(1)` is the treatment sum of squares (unadjusted) and `SSALT(2)` is the block sum of squares (adjusted).

TESTLF — Vector of length 10 containing statistics relating to the test for lack of fit of the model. (Output, if `NRESP > 1`)

If `NRESP = 1`, `TESTLF` is not referenced and can be a vector of length one. Elements of `TESTLF` are described as follows:

Elem. Description

- 1 Degrees of freedom for experimental error
- 2 Degrees of freedom for within-cell error
- 3 Degrees of freedom for error (`TESTLF(1) + TESTLF(2)`)
- 4 Sum of squares for experimental error

Elem. Description

- 5 Sum of squares for within-cell error
- 6 Sum of squares for error
- 7 Mean square for experimental error
- 8 Mean square for within-cell error
- 9 F -statistic
- 10 p -value

YMEANS — Vector of length $NREP + NBLK + NTRT + NTBLK * NBLK$ containing the replicate means, block by replicate means, treatment means (adjusted), and treatment by block means, respectively. (Output)
The treatment means (adjusted) in **YMEANS** are used for estimating treatment differences.

SETRTD — Estimated standard error of a treatment difference. (Output)

EFNCY — Estimated efficiency of this design relative to a randomized complete block design. (Output)
The randomized complete block design has $NBLK * NTBLK / NTRT$ complete blocks.

FORTRAN 90 Interface

Generic: CALL ABIBD (NTRT, NREP, NBLK, NTBLK, NRESP, Y, ITRT, AOV [, ...])

Specific: The specific interface names are S_ABIBD and D_ABIBD.

FORTRAN 77 Interface

Single: CALL ABIBD (NTRT, NREP, NBLK, NTBLK, NRESP, Y, ITRT, INTER, IPRINT, AOV, SQSS, SSALT, TESTLF, YMEANS, SETRTD, EFNCY)

Double: The double precision name is DABIBD.

Example

This example performs an intrablock analysis for a balanced incomplete block design using data discussed by Anderson and Bancroft (1952, pages 254–256). The responses are weight gains of rats fed $p = 9$ different rations. There are four replications with $k = 3$ blocks within each replicate. (Since $p = k^2$, this balanced incomplete block design is a balanced lattice design.) The data with the treatment numbers in parentheses are given in the following table:

Replicate	Block	(Treatment): Weight Gain
1	1	(1): 20 (4): 15 (7): 11
1	2	(3): 8 (6): 18 (9): 26
1	3	(2): 18 (5): 16 (8): 2
2	1	(7): 8 (8): 12 (9): 16
2	2	(1): 20 (2): 2 (3): 2
2	3	(4): 20 (5): 6 (6): 2
3	1	(1): 13 (9): 19 (5): 14
3	2	(8): 14 (4): 34 (3): 2
3	3	(6): 14 (2): 20 (7): 14
4	1	(5): 19 (7): 23 (3): 6
4	2	(1): 22 (6): 12 (8): 2
4	3	(9): 27 (2): 7 (4): 20

```

USE ABIBD_INT
INTEGER  NBLK, NREP, NRESP, NTBLK, NTRT
PARAMETER (NBLK=12, NREP=4, NRESP=1, NTBLK=3, NTRT=9)
!
INTEGER  IPRINT, ITRT(NBLK*NTBLK)
REAL     AOV(15), Y(NBLK*NTBLK*NRESP)
!
DATA Y/20.0, 15.0, 11.0, 8.0, 18.0, 26.0, 18.0, 16.0, 2.0, 8.0, &
      12.0, 16.0, 20.0, 2.0, 2.0, 20.0, 6.0, 2.0, 13.0, 19.0, &
      14.0, 14.0, 34.0, 2.0, 14.0, 20.0, 14.0, 19.0, 23.0, 6.0, &
      22.0, 12.0, 2.0, 27.0, 7.0, 20.0/
DATA ITRT/1, 4, 7, 3, 6, 9, 2, 5, 8, 7, 8, 9, 1, 2, 3, 4, 5, 6, &
      1, 9, 5, 8, 4, 3, 6, 2, 7, 5, 7, 3, 1, 6, 8, 9, 2, 4/
!
IPRINT = 3
CALL ABIBD (NTRT, NREP, NBLK, NTBLK, NRESP, Y, ITRT, &
           AOV, IPRINT=IPRINT)
END

```

Output

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Coefficient of Mean Var. (percent)
Y	79.771	55.748	5.345	14 38.18

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Model	19	1802.8	94.88	3.321	0.0095
Error	16	457.2	28.57		
Corrected Total	35	2260.0			

* * * Decomposition of Variation Attributable to the Model * * *

Source	DF	Sum of Squares	F	Prob. of Larger F
Replicates	3	219.6	2.561	0.0913
Blocks within Replicates	8	127.1	0.556	0.7980
Treatments (adjusted)	8	1456.1	6.370	0.0009

* * * Replicate means * * *

Replicate	Mean (N=4)
1	14.8889
2	9.7778
3	16.0000
4	15.3333

* * * Block by Replicate Means * * *

Replicate	Block	Mean (N=3)
1	1	15.3333
1	2	17.3333
1	3	12.0000
2	1	12.0000
2	2	8.0000
2	3	9.3333
3	1	15.3333
3	2	16.6667
3	3	16.0000
4	1	16.0000
4	2	12.0000
4	3	18.0000

* * * Adjusted Treatment Means * * *

Treatment	Mean (N=1)
1	22.11
2	11.67
3	0.67
4	23.89
5	14.78
6	11.11
7	12.89
8	6.44
9	22.44

* * * Treatment by Block Means * * *

Replicate	Block	Treatment	Mean (N=1)
1	1	1	20.0000
1	1	4	15.0000
1	1	7	11.0000
1	2	3	8.0000
1	2	6	18.0000
1	2	9	26.0000
1	3	2	18.0000
1	3	5	16.0000
1	3	8	2.0000

2	1	7	8.0000
2	1	8	12.0000
2	1	9	16.0000
2	2	1	20.0000
2	2	2	2.0000
2	2	3	2.0000
2	3	4	20.0000
2	3	5	6.0000
2	3	6	2.0000
3	1	1	13.0000
3	1	9	19.0000
3	1	5	14.0000
3	2	8	14.0000
3	2	4	34.0000
3	2	3	2.0000
3	3	6	14.0000
3	3	2	20.0000
3	3	7	14.0000
4	1	5	19.0000
4	1	7	23.0000
4	1	3	6.0000
4	2	1	22.0000
4	2	6	12.0000
4	2	8	2.0000
4	3	9	27.0000
4	3	2	7.0000
4	3	4	20.0000

Comments

Workspace may be explicitly provided, if desired, by use of A2IBD/DA2IBD. The reference is:

```
CALL A2IBD (NTRT, NREP, NBLK, NTBLK, NRESP, Y, ITRT, INTER, IPRINT,
AOV, SQSS, SSALT, TESTLF, YMEANS, SETTRTD, EFNCY, WK)
```

The additional argument is:

WK — Work vector of length NTRT or 2 * NTRT.

Description

Routine ABIBD performs analyses for balanced incomplete block designs. The basic model used is the randomized block design with the source of variation for “blocks” subdivided into replications and blocks within replications. For INTER = 0, the model is

$$y_{ijtm} = \mu + \alpha_i + \beta_{jj} + \delta_t + \varepsilon_{ijkm} \quad i = 1, \dots, r; j = 1, \dots, k; t = 1, \dots, p; m = 1, \dots, n$$

where the observed value of y_{ijtm} constitutes the m -th response with treatment t in block j within the i replicate, $\mu + \alpha_i + \beta_{jj} + \delta_t$ is the population mean for the response, and the ε_{ijkm} 's are independently distributed normal errors with mean zero and variance σ^2 . This model assumes the block effects and treatment effects are additive. Often in practice, there are interactions between the blocks and treatments. For this reason, ABIBD computes a test for nonadditivity

(lack of fit), in addition to summary statistics for the additive model. This test requires at least two responses in each cell.

The analysis performed with the β_{ij} 's regarded as fixed effects in the model ($\text{INTER} = 0$) is called an "intra-block analysis." For $\text{INTER} = 1$, the β_{ij} 's are assumed to be random effects in the model, the analysis performed for this mixed model is called an "interblock analysis."

Routine `ABIBD` requires the y_{ijm} 's to be entered in a single vector Y ordered lexicographically, so that the i subscript varies least rapidly, the j subscript the next most rapidly, and so forth. Formulas and interpretations for the analysis of balanced incomplete block designs are discussed by Anderson and Bancroft (1952, Chapters 19 and 24) and Kempthorne (1975, pages 532–539).

ALATN

Analyzes a Latin square design.

Required Arguments

NTRT — Number of treatments. (Input)

NTRT must also be the number of rows and the number of columns.

NRESP — Number of repeated responses within each row-column position. (Input)

Y — Vector of length $\text{NTRT} * \text{NTRT} * \text{NRESP}$ containing the responses. (Input)

The first *NRESP* elements of *Y* contain the responses for row 1, column 1; the second *NRESP* elements of *Y* contain the responses for row 1, column 2. The last *NRESP* elements of *Y* contain the responses for row *NTRT*, column *NTRT*.

ITRT — Vector of length $\text{NTRT} * \text{NTRT}$ containing the treatment numbers for the responses in *Y*. (Input)

The treatment numbers must be from the set 1, 2, ..., *NTRT*. For $I = 1, 2, \dots, \text{NTRT} * 2$, element numbers $(I - 1) * \text{NRESP} + 1$ through $(I - 1) * \text{NRESP} + \text{NRESP}$ of *Y* correspond to treatment number *ITRT*(*I*).

AOV — Vector of length 15 containing statistics relating to the analysis of variance. (Output)

I ***AOV***(***I***)

1 Degrees of freedom for the model (rows, columns and treatments)

2 Degrees of freedom for error (experimental error pooled with the within-cell error)

3 Total (corrected) degrees of freedom

4 Sum of squares for the model (rows, columns, and treatments)

I	AOV(I)
5	Sum of squares for error (experimental error pooled with the within-cell error)
6	Total (corrected) sum of squares
7	Model mean square
8	Error mean square
9	<i>F</i> -statistic
10	<i>p</i> -value
11	R^2 (in percent)
12	Adjusted <i>R</i> (in percent)
13	Estimated standard deviation of the model error
14	Overall mean of <i>Y</i>
15	Coefficient of variation (in percent)

Optional Arguments

IPRINT — Printing option. (Input)
 Default: IPRINT = 0.

IPRINT Action

- 0 No printing is performed.
- 1 Print AOV, EFSS, and TESTLF (if NRESP > 1) only.
- 2 Print YMEANS only.
- 3 All print is performed.

EFSS — Vector of length 12 containing statistics relating to the sums of squares for the effects in the model. (Output)
 Elements of EFSS are described as follows:

Elem. Description

- 1, 2, 3 Degrees of freedom for rows, columns, and treatments, respectively.
- 4, 5, 6 Sum of squares for rows, columns, and treatments, respectively.

Elem. Description

7, 8, 9 *F*-statistics for rows, columns, and treatments, respectively. *F*-statistics are computed using `AOV(8)` as the estimated error variance.

10–12 *p*-values associated with the *F*-statistics.

TESTLF — Vector of length 10 containing statistics relating to the test for lack of fit of the model. (Output if `NRESP > 1`)

If `NRESP = 1`, **TESTLF** is not referenced and can be a vector of length one. Elements of **TESTLF** are described as follows:

Elem. Description

1 Degrees of freedom for experimental error

2 Degrees of freedom for within-cell error

3 Degrees of freedom for error (`TESTLF(1) + TESTLF(2)`)

4 Sum of squares for experimental error

5 Sum of squares for within-cell error

6 Sum of squares for error

7 Mean square for experimental error

8 Mean square for within-cell error

9 *F*-statistic

10 *p*-value

YMEANS — Vector of length $3 * NTRT + NTRT * NTRT$ containing the row means, column means, treatment means, and the row-column means, respectively. (Output)

FORTRAN 90 Interface

Generic: `CALL ALATN (NTRT, NRESP, Y, ITRT, AOV [,...])`

Specific: The specific interface names are `S_ALATN` and `D_ALATN`.

FORTRAN 77 Interface

Single: `CALL ALATN (NTRT, NRESP, Y, ITRT, IPRINT, AOV, EFSS, TESTLF, YMEANS)`

Double: The double precision name is DALATN.

Example

This example performs an analysis for a Latin square design using data discussed by Kirk (1982, Table 7.3-2, pages 312–317). The responses are thickness of tread remaining on each of 32 tires after 10,000 miles of driving. The tires are divided equally among four different types, labeled A, B, C, and D. Four cars are used in the study. The experiment is performed twice, sixteen tires are used in each experiment. Each of the sixteen tires occupies one of the four wheel positions on one of the cars. The data are given in the following table:

Wheel Position	Car 1	Car 2	Car 3	Car 4
Right Front	A: 1, 2	B: 2, 3	C: 5, 6	D: 9, 8
Left Front	B: 3, 4	C: 8, 6	D: 9, 8	A: 2, 3
Right Rear	C: 5, 7	D: 10, 11	A: 3, 2	B: 5, 4
Left Rear	D: 7, 10	A: 6, 3	B: 3, 4	C: 6, 6

```

USE ALATN_INT
INTEGER      NRESP, NTRT
PARAMETER   (NRESP=2, NTRT=4)
!
INTEGER      IPRINT, ITRT(NTRT*NTRT)
REAL        AOV(15), Y(NTRT*NTRT*NRESP)
!
DATA Y/1.0, 2.0, 2.0, 3.0, 5.0, 6.0, 9.0, 8.0, 3.0, 4.0, 8.0, &
      6.0, 9.0, 8.0, 2.0, 3.0, 5.0, 7.0, 10.0, 11.0, 3.0, 2.0, &
      5.0, 4.0, 7.0, 10.0, 6.0, 3.0, 3.0, 4.0, 6.0, 7.0/
DATA ITRT/1, 2, 3, 4, 2, 3, 4, 1, 3, 4, 1, 2, 4, 1, 2, 3/
DATA IPRINT/3/
!
CALL ALATN (NTRT, NRESP, Y, ITRT, AOV, IPRINT=IPRINT)
END

```

Output

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	89.809	85.640	1.044	5.375	19.43

```

* * * Analysis of Variance * * *
Source          DF      Sum of Squares      Mean Square      Overall F      Prob. of Larger F
Model           9         211.5           23.50           21.542         0.0000
Error          22          24.0            1.09
Corrected Total 31         235.5

```

```

* * * Decomposition of Variation Attributable to the Model * * *
Source          DF      Sum of Squares      F      Prob. of Larger F
Row             3          9.2           2.826     0.0622
Column          3          7.8           2.368     0.0983
Treatment       3         194.5          59.431     0.0000

```

Source	Test for Lack of Fit			F	Prob. of Larger F
	DF	Sum of Squares	Mean Square		
Experimental Error	6	5	0.833	0.702	0.6525
Within Cell	16	19	1.188		
Error	22	24			

* * * Row Means * * *

Row	Mean (N=4)
1	4.500
2	5.375
3	5.875
4	5.750

* * * Column Means * * *

Column	Mean (N=4)
1	4.875
2	6.125
3	5.000
4	5.500

* * * Treatment Means * * *

Treatment	Mean (N=4)
1	2.8
2	3.5
3	6.2
4	9.0

* * * Cell Means * * *

Row	Column	Mean (N=2)
1	1	1.500
1	2	2.500
1	3	5.500
1	4	8.500
2	1	3.500
2	2	7.000
2	3	8.500
2	4	2.500
3	1	6.000
3	2	10.500
3	3	2.500
3	4	4.500
4	1	8.500
4	2	4.500
4	3	3.500
4	4	6.500

Description

Routine ALATN performs an analysis for a Latin square design. The model is

$$y_{ijkm} = \mu + \alpha_i + \beta_j + \delta_k + \varepsilon_{ijkm} \quad i, j, k = 1, 2, \dots, p; m = 1, 2, \dots, n$$

where the observed value of y_{ijkm} constitutes the m -th response on the k -th treatment in row i column j of the Latin square design; $\mu + \alpha_i + \beta_j + \delta_k$ is the population mean for the response, and the ε_{ijkm} 's are identically and independently distributed normal errors with mean zero and variance σ^2 . This model assumes the row effects (α_i), column effects (β_j), and treatment effects (δ_k) are additive. Often in practice, there are interactions between two or more of these factors. For this reason, `ALATN` computes a test for nonadditivity (lack of fit), in addition to summary statistics for the additive model. This test requires at least two responses in each cell. A test for nonadditivity with one response per cell in a Latin square design is discussed by Snedecor and Cochran (1967, pages 334–337).

Routine `ALATN` requires y_{ijk} 's to be entered in single vector `Y` with the data for each cell occupying contiguous elements. The cells must be in standard order, i.e., (1, 1), (1, 2), ..., (1, p), (2, 1), (2, 2), ..., (2, p), ..., (p , 1), (p , 2), ..., (p , p). A discussion of formulas and interpretations for the analysis of a Latin square design appears in many elementary statistics texts, e.g., Snedecor and Cochran (1967, pages 312–317).

ANWAY

Analyzes a balanced n -way classification model with fixed effects.

Required Arguments

NF — Number of factors (number of subscripts) in the model including error. (Input)

NL — Vector of length **NF** containing the number of levels for each of the factors. (Input)

Y — Vector of length $NL(1) * NL(2) * \dots * NL(NF)$ containing the responses. (Input)
`Y` must not contain NaN (not a number) for any of its elements, i.e., missing values are not allowed.

INTERA — Interaction option. (Input)

The absolute value of **INTERA** is the number of factors to be included in the highest-way interaction in the model. The sign of **INTERA** indicates if factor **NF** is error.

INTERA **Meaning**

< 0 Factor **NF** is not error. Only $(-INTERA + 1)$ -way and higher-way interactions are included in error.

> 0 Factor **NF** is error. Its main effect and all its interaction effects are pooled into the error with the other $(INTERA + 1)$ -way and higher-way

AOV — Vector of length 15 containing statistics relating to the analysis of variance. (Output)

I	AOV(I)
1	Degrees of freedom for the model
2	Degrees of freedom for error
3	Total (corrected) degrees of freedom
4	Sum of squares for the model
5	Sum of squares for error
6	Total (corrected) sum of squares
7	Model mean square
8	Error mean square
9	<i>F</i> -statistic
10	<i>p</i> -value
11	R^2 (in percent)
12	Adjusted R^2 (in percent)
13	Estimated standard deviation of the model error
14	Overall mean of Y
15	Coefficient of variation (in percent)

Optional Arguments

IPRINT — Printing option. (Input)

Default: `IPRINT = 0`.

IPRINT	Action
---------------	---------------

0	Printing is not performed.
1	AOV and EFSS are printed.
2, -2	Only marginal means are printed. If <code>IPRINT = 2</code> , then all of <code>YMEANS</code> is printed. If <code>IPRINT = -2</code> , then marginal means higher than (<code> INTERA </code>) -way are not printed.

IPRINT Action

3, -3 AOV, EFSS, and all or some of YMEANS is printed. If IPRINT = 3, then all of YMEANS is printed. If IPRINT = -3, then marginal means higher than (INTERA) -way are not printed.

EFSS — NEF by 4 matrix containing statistics relating to the sums of squares for the effects in the model. (Output)

Here, $NEF = \text{BINOM}(n, 1) + \text{BINOM}(n, 2) + \dots + \text{BINOM}(n, |\text{INTERA}|)$ where the IMSL subroutine BINOM (IMSL MATH/LIBRARY Special Functions) returns the binomial coefficient, and n is given by

$$n = \begin{cases} \text{NF} & \text{if INTERA is negative} \\ \text{NF}-1 & \text{if INTERA is positive} \end{cases}$$

Suppose the factors are A, B, C , and error. With INTERA = 3, rows 1 through NEF would correspond to A, B, C, AB, AC, BC , and ABC , respectively. The columns of EFSS are as follows:

Column Description

1	Degrees of freedom
2	Sum of squares
3	F -statistic
4	p -value

LDEFSS — Leading dimension of EFSS exactly as specified in the dimension statement in the calling program. (Input)

Default: LDEFSS = size (EFSS , 1)

YMEANS — Vector of length $(NL(1) + 1) * (NL(2) + 1) * \dots * (NL(n) + 1)$ containing subgroup means. (Output)

See argument EFSS for a definition of n . Suppose that the factors are A, B, C , and error. The ordering of the means is grand mean, A means, B means, C means, AB means, AC means, BC means, and ABC means.

FORTRAN 90 Interface

Generic: CALL ANWAY (NF, NL, Y, INTERA, AOV [,...])

Specific: The specific interface names are S_ANWAY and D_ANWAY.

FORTRAN 77 Interface

Single: CALL ANWAY (NF, NL, Y, INTERA, IPRINT, AOV, EFSS, LDEFSS,
 YMEANS)

Double: The double precision name is DANWAY.

Example 1

A two-way analysis of variance is performed with balanced data discussed by Snedecor and Cochran (1967, Table 12.5.1, page 347). The responses are the weight gains (in grams) of rats fed diets varying in two components—source of protein (A) and level of protein (B). Here, INTERA = 2 is used. The model is

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \varepsilon_{ijk} \quad i = 1, 2; j = 1, 2, 3; k = 1, 2, \dots, 10$$

where

$$\sum_{i=1}^2 \alpha_i = 0; \sum_{j=1}^3 \beta_j = 0; \sum_{i=1}^2 \gamma_{ij} = 0$$

for $j = 1, 2, 3$; and

$$\sum_{j=1}^3 \gamma_{ij} = 0$$

for $i = 1, 2$.

The first responses in each cell in the two-way layout are given in the following table:

	Protein Source (A)		
Protein Level (B)	Beef	Cereal	Pork
High	73, 102, 118, 104, 81, 107, 100, 87, 117, 111	98, 74, 56, 111, 95, 88, 82, 77, 86, 92	94, 79, 96, 98, 102, 102, 108, 91, 120, 105
Low	90, 76, 90, 64, 86, 51, 72, 90, 95, 78	107, 95, 97, 80, 98, 74, 74, 67, 89, 58	49, 82, 73, 86, 81, 97, 106, 70, 61, 82

```

USE ANWAY_INT
INTEGER    NF, NOBS
PARAMETER  (NF=3, NOBS=60)
!
INTEGER    INTERA, IPRINT, NL(NF)
REAL       AOV(15), Y(NOBS)
!
DATA Y/73.0, 102.0, 118.0, 104.0, 81.0, 107.0, 100.0, 87.0, &
      117.0, 111.0, 90.0, 76.0, 90.0, 64.0, 86.0, 51.0, 72.0, &
      90.0, 95.0, 78.0, 98.0, 74.0, 56.0, 111.0, 95.0, 88.0, &
      82.0, 77.0, 86.0, 92.0, 107.0, 95.0, 97.0, 80.0, 98.0, &
      74.0, 74.0, 67.0, 89.0, 58.0, 94.0, 79.0, 96.0, 98.0, &
      102.0, 102.0, 108.0, 91.0, 120.0, 105.0, 49.0, 82.0, 73.0, &
      86.0, 81.0, 97.0, 106.0, 70.0, 61.0, 82.0/
DATA NL/3, 2, 10/
!
INTERA = 2
IPRINT = 3
CALL ANWAY (NF, NL, Y, INTERA, AOV, IPRINT=IPRINT)
END

```

Output

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	28.477	21.854	14.65	87.87	16.67

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Model	5	4612.9	922.6	4.300	0.0023
Error	54	11586.0	214.6		
Corrected Total	59	16198.9			

* * * Variation Due to the Model * * *

Source	DF	Sum of Squares	F	Prob. of Larger F
A	2	266.53	0.621	0.5411
B	1	3168.27	14.767	0.0003
A*B	2	1178.13	2.746	0.0732

* * * Subgroup Means * * *

A Means (N=20)	
1	89.6000
2	84.9000

```

3          89.1000
B Means  (N=30)
1          95.1333
2          80.6000
A*B Means (N=10)
1  1      100.0000
1  2       79.2000
2  1       85.9000
2  2       83.9000
3  1       99.5000
3  2       78.7000

```

Comments

Workspace may be explicitly provided, if desired, by use of `A2WAY/DA2WAY`. The reference is:

```
CALL A2WAY (NF, NL, Y, INTERA, IPRINT, AOV, EFSS, LDEFSS, YMEANS, WK,
IWK)
```

The additional arguments are as follows:

WK — Work vector of length $5 * 2^n + \text{NMEANS} + 4$.

IWK — Work vector of length $(\text{NF} + 2) * 2^{\text{NF}-1} + (n + 2) * 2^{n-1} + n - 2$.

Description

Routine `ANWAY` performs an analysis for an n -way classification design with balanced data. For balanced data, there must be an equal number of responses in each cell of the n -way layout. The effects are assumed to be fixed effects. The model is an extension of the twoway model to include n factors. The interactions (two-way, three-way, up to n -way) can be included in the model, or some of the higher-way interactions can be pooled into error. The argument `INTERA` specifies which interactions are to be pooled into error. For example, if three-way and higher-way interactions are to be pooled into error, set `INTERA = -2` or `INTERA = 2`. A positive `INTERA` indicates there are repeated responses within the n -way cells, while a negative `INTERA` indicates otherwise.

Routine `ANWAY` requires the responses as input into a single vector `Y` in lexicographical order so that the response subscript associated with the first factor varies least rapidly, the subscript associated with the second factor varies next most rapidly, and so forth. Hemmerle (1967, Chapter 5) discusses the computational method.

Additional Example

Example 2

This example performs a three-way analysis of variance using data discussed by John (1971, pages 91–92). The responses are weights (in grams) of roots of carrots grown with varying amounts of applied nitrogen (A), potassium (B), and phosphorus (C). There is one response within each cell of the three-way layout. `INTERA` is set to `-2` in order to pool the ABC three-

factor interaction into error. (Note that the *ABC* interaction sum of squares, which is 186, is given incorrectly by John [1971, Table 5.2].) *I*PRINT is set to -3 so that the *ABC* means will not be printed (since *|INTERA|* is equal to 2). The three-way layout is given in the following table:

	<i>A</i> ₀			<i>A</i> ₁			<i>A</i> ₂		
	<i>B</i> ₀	<i>B</i> ₁	<i>B</i> ₂	<i>B</i> ₀	<i>B</i> ₁	<i>B</i> ₂	<i>B</i> ₀	<i>B</i> ₁	<i>B</i> ₂
<i>C</i> ₀	88.76	91.41	97.85	94.83	100.49	99.75	99.90	100.23	104.51
<i>C</i> ₁	87.45	98.27	95.85	84.57	97.20	112.30	92.98	107.77	110.94
<i>C</i> ₂	86.01	104.20	90.09	81.06	120.80	108.77	94.72	118.39	102.87

```

USE ANWAY_INT
INTEGER    NF, NOBS
PARAMETER  (NF=3, NOBS=27)
!
INTEGER    INTERA, IPRINT, NL(NF)
REAL      AOV(15), Y(NOBS)
!
DATA Y/88.76, 87.45, 86.01, 91.41, 98.27, 104.20, 97.85, 95.85, &
      90.09, 94.83, 84.57, 81.06, 100.49, 97.20, 120.8, 99.75, &
      112.30, 108.77, 99.9, 92.98, 94.72, 100.23, 107.77, 118.39, &
      104.51, 110.94, 102.87/
DATA NL/3, 3, 3/
!
INTERA = -2
IPRINT = -3
CALL ANWAY (NF, NL, Y, INTERA, AOV, IPRINT=IPRINT)
END

```

Output

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	92.804	76.612	4.819	98.96	4.869

```

* * * Analysis of Variance * * *
Source          DF      Sum of Squares      Mean Square      Overall F      Prob. of Larger F
Model           18      2395.7             133.1           5.731          0.0083
Error            8       185.8             23.2
Corrected Total 26      2581.5

```

```

* * * Variation Due to the Model * * *
Source          DF      Sum of Squares      F      Prob. of Larger F
A                2       488.37            10.515    0.0058
B                2      1090.66            23.483    0.0004
C                2       49.15             1.058     0.3911
A*B             4      142.59             1.535     0.2804

```

A*C	4	32.35	0.348	0.8383
B*C	4	592.62	6.380	0.0131

* * * Subgroup Means * * *

A Means (N=9)

1	93.3211
2	99.9744
3	103.5900

B Means (N=9)

1	90.0311
2	104.3067
3	102.5478

C Means (N=9)

1	97.5256
2	98.5922
3	100.7678

A*B Means (N=3)

1	1	87.4067
1	2	97.9600
1	3	94.5967
2	1	86.8200
2	2	106.1633
2	3	106.9400
3	1	95.8667
3	2	108.7967
3	3	106.1067

A*C Means (N=3)

1	1	92.6733
1	2	93.8567
1	3	93.4333
2	1	98.3567
2	2	98.0233
2	3	103.5433
3	1	101.5467
3	2	103.8967
3	3	105.3267

B*C Means (N=3)

1	1	94.4967
1	2	88.3333
1	3	87.2633
2	1	97.3767
2	2	101.0800
2	3	114.4633
3	1	100.7033
3	2	106.3633
3	3	100.5767

ABALD

Analyzes a balanced complete experimental design for a fixed, random, or mixed model.

Required Arguments

NL — Vector of length NF containing the number of levels for each of the factors. (Input)

Y — Vector of length $NL(1) * NL(2) * \dots * NL(NF)$ containing the responses. (Input)
Y must not contain NaN (not a number) for any of its elements, i.e., missing values are not allowed.

NRF — For positive NRF , $-NRF$ is the number of random factors. (Input)
For negative NRF , $-NRF$ is the number of random effects (sources of variation).

INDRF — Index vector of length $|NRF|$ containing either the factor numbers to be considered random (for NRF positive) or containing the effect numbers to be considered random (for NRF negative). (Input)
If $NRF = 0$, **INDRF** is not referenced and can be a vector of length one.

NFEF — Vector of length NEF containing the number of factors associated with each effect in the model. (Input)

INDEF — Index vector of length $NFEF(1) + NFEF(2) + \dots + NFEF(NEF)$. (Input)
The first $NFEF(1)$ elements give the factor numbers in the first effect. The next $NFEF(2)$ elements give the the factor numbers in the second effect. The last $NFEF(NEF)$ elements give the factor numbers in the last effect. Main effects must appear before their interactions. In general, an effect E cannot appear after an effect F if all of the indices for E appear also in F .

AOV — Vector of length 15 containing statistics relating to the analysis of variance. (Output)

I	AOV(I)
1	Degrees of freedom for the model
2	Degrees of freedom for error
3	Total (corrected) degrees of freedom
4	Sum of squares for the model
5	Sum of squares for error
6	Total (corrected) sum of squares
7	Model mean square
8	Error mean square
9	F -statistic

I	AOV(I)
10	p -value
11	R^2 (in percent)
12	Adjusted R^2 (in percent)
13	Estimated standard deviation of the model error
14	Overall mean of Y
15	Coefficient of variation (in percent)

Optional Arguments

NF — Number of factors (number of subscripts) in the model, including error. (Input)
 Default: $NF = \text{size}(NL, 1)$.

NEF — Number of effects (sources of variation) due to the model excluding the overall mean and error. (Input)
 Default: $NEF = \text{size}(NFEF, 1)$.

CONPER — Confidence level for two-sided interval estimates on the variance components, in percent. (Input)
 CONPER percent confidence intervals are computed, hence, CONPER must be in the interval [0.0, 100.0). CONPER often will be 90.0, 95.0, or 99.0. For one-sided intervals with confidence level ONECL, ONECL in the interval [50.0, 100.0), set $CONPER = 100.0 - 2.0 * (100.0 - ONECL)$.
 Default: $CONPER = 95.0$.

IPRINT — Printing option. (Input)
 Default: $IPRINT = 0$.

IPRINT Action

0	No printing is performed.
1	All is performed.
$-k$	Printing restricted to exclude marginal means higher than k ways. For example, only one-way and two-way marginal means will be printed if $IPRINT = -2$.

Let

$$n = \begin{cases} \text{NF} & \text{if INDEF contains one or more elements equal to NF} \\ \text{NF} - 1 & \text{otherwise} \end{cases}$$

The value of `IPRINT` must be between $-n$ and 1, inclusively.

MODEL — Model Option. (Input)

Default: `MODEL = 0`.

MODEL **Meaning**

0 Searle model

1 Scheffe model

For the Scheffe model, effects corresponding to interactions of fixed and random factors have their sum over the subscripts corresponding to fixed factors equal to zero. Also, the variance of a random interaction effect involving some fixed factors has a multiplier for the associated variance component that involves the number of levels in the fixed factors. The Searle model has no summation restrictions on the random interaction effects and has a multiplier of one for each variance component.

EMS — Vector of length $(\text{NEF} + 1) * (\text{NEF} + 2)/2$ containing expected mean square coefficients. (Output)

Suppose the effects are *A*, *B*, and *AB*. The ordering of the coefficients in EMS is as follows:

Error	<i>AB</i>	<i>B</i>	<i>A</i>	
<i>A</i>	EMS(1)	EMS(2)	EMS(3)	EMS(4)
<i>B</i>	EMS(5)	EMS(6)	EMS(7)	
<i>AB</i>	EMS(8)	EMS(9)		
Error	EMS(10)			

VC — $\text{NEF} + 1$ by 9 matrix containing statistics relating to the particular variance components or effects in the model and the error. (Output)

Rows of `VC` correspond to the `NEF` effects plus error. Columns of `VC` are as follows:

Column **Description**

1 Degrees of freedom

2 Sum of squares

3 Mean squares

4 *F* -statistic

5 *p*-value for *F* test

Column	Description
--------	-------------

6	Variance component estimate
7	Percent of variance of y explained by random effect
8	Lower endpoint for a confidence interval on the variance component
9	Upper endpoint for a confidence interval on the variance component

Columns 6 through 9 contain NaN (not a number) if the effect is fixed, i.e., if there is no variance component to be estimated. If the variance component estimate is negative, columns 8 and 9 contain NaN.

LDVC — Leading dimension of VC exactly as specified in the dimension statement of the calling program. (Input)

Default: LDVC = size(VC ,1).

YMEANS — Vector of length $(NL(1) + 1) * (NL(2) + 1) * \dots * (NL(n) + 1)$ containing the subgroup means. (Output)

Suppose the factors are A , B , and C . The ordering of the means is grand mean, A means, B means, C means, AB means, AC means, BC means, and ABC means.

FORTRAN 90 Interface

Generic: CALL ABALD (NL, Y, NRF, INDRF, NFEF, INDEF, AOV [,...])

Specific: The specific interface names are S_ABALD and D_ABALD.

FORTRAN 77 Interface

Single: CALL ABALD (NF, NL, Y, NRF, INDRF, NEF, NFEF, INDEF, CONPER, IPRINT, MODEL, AOV, EMS, VC, LDVC, YMEANS)

Double: The double precision name is DABALD.

Example 1

An analysis of a generalized randomized block design is performed using data discussed by Kirk (1982, Table 6.10-1, pages 293–297). The model is

$$y_{ijk} = \mu + \alpha_i + b_j + c_{ij} + e_{ijk} \quad i = 1, 2, 3, 4; j = 1, 2, 3, 4; k = 1, 2$$

where y_{ijk} is the response for the k -th experimental unit in block j with treatment i ; the α_i 's are the treatment effects and are subject to the restriction

$$\sum_{i=1}^2 \alpha_i = 0$$

the b_j 's are block effects identically and independently distributed

$$N(0, \sigma_B^2)$$

c_{ij} are interaction effects each distributed

$$N(0, \frac{3}{4}\sigma_{AB}^2)$$

and are subject to the restrictions

$$\sum_{i=1}^4 c_{ij} = 0 \text{ for } j = 1, 2, 3, 4$$

and the e_{ijk} 's are errors, identically and independently distributed $N(0, \sigma^2)$. The interaction effects are assumed to be distributed independently of the errors.

The data are given in the following table:

Treatment	Block			
	1	2	3	4
1	3, 6	3, 1	2, 2	3, 2
2	4, 5	4, 2	3, 4	3, 3
3	7, 8	7, 5	6, 5	6, 6
4	7, 8	9, 10	10, 9	8, 11

```

USE ABALD_INT
INTEGER   LINDEF, NEF, NF, NOBS, NRF
PARAMETER (LINDEF=4, NEF=3, NF=3, NOBS=32, NRF=2)
!
INTEGER   INDEF(LINDEF), INDRF(NRF), IPRINT, MODEL, NFEF(NEF), &
          NL(NF)
REAL      AOV(15), Y(NOBS)
!
DATA NL/4, 4, 2/
DATA INDRF/2, 3/
DATA NFEF/1, 1, 2/
DATA INDEF/1, 2, 1, 2/
DATA Y/3.0, 6.0, 3.0, 1.0, 2.0, 2.0, 3.0, 2.0, 4.0, 5.0, 4.0, &
      2.0, 3.0, 4.0, 3.0, 3.0, 7.0, 8.0, 7.0, 5.0, 6.0, 5.0, &
      6.0, 6.0, 7.0, 8.0, 9.0, 10.0, 10.0, 9.0, 8.0, 11.0/
!
IPRINT = 1
MODEL = 1
CALL ABALD (NL, Y, NRF, INDRF, NFEF, INDEF, AOV, &
          IPRINT=IPRINT, MODEL=MODEL)
END

```

Output

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Coefficient of Mean Var. (percent)
Y	91.932	84.368	1.09	5.375 20.27

\newpage

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Model	15	216.5	14.43	12.154	0.0000
Error	16	19.0	1.19		
Corrected Total	31	235.5			

Source	DF	Sum of Squares	Mean Square	F	Prob. of Larger F
A	3	194.50	64.8333	32.873	0.0000
B	3	4.25	1.4167	1.193	0.3440
AB	9	17.75	1.9722	1.661	0.1802

* * * EMS * * *

	Error	AB	B	A
A	1	2	0	8
B	1	0	8	
AB	1	2		
Error	1			

* * * Variance Components * * *

Variance Component	Estimate	Percent	95.0% Confidence Interval	
			Lower Limit	Upper Limit
B	0.0286	1.897	0.00000	2.3168
AB	0.3924	19.483	0.00000	2.7580
Error	1.1875	78.621	0.65868	2.7506

* * * Subgroup Means * * *

A Means (N=8)	
1	2.7500
2	3.5000
3	6.2500
4	9.0000
B Means (N=8)	
1	6.0000
2	5.1250
3	5.1250
4	5.2500
AB Means (N=2)	
1 1	4.5000
1 2	2.0000
1 3	2.0000
1 4	2.5000
2 1	4.5000
2 2	3.0000
2 3	3.5000
2 4	3.0000
3 1	7.5000
3 2	6.0000
3 3	5.5000
3 4	6.0000
4 1	7.5000
4 2	9.5000

4 3 9.5000
 4 4 9.5000

Comments

Workspace may be explicitly provided, if desired, by use of A2ALD/DA2ALD. The reference is:

CALL A2ALD (NF, NL, Y, NRF, INDRF, NEF, NFEF, INDEF, CONPER, IPRINT,
 MODEL, AOV, EMS, VC, LDVC, YMEANS, WK, IWK, CHWK)

The additional arguments are as follows:

WK — Work vector of length $3 * 2^{NF} + 2 * NEF + m + 4$.

IWK — Work vector of length $\max(2^{NF}, NF + NEF + LINDEF) + 2^{NF} - 1 + NF * 2^{NF-1}$.

CHWK — CHARACTER * 13 vector of length $\max(NEF + 3, 2^n - 1)$. If IPRINT = 0, CHWK is not referenced and can be a vector of length one.

Description

Routine ABALD analyzes a balanced complete experimental design for a fixed, random, or mixed model. The analysis includes an analysis of variance table, and computation of subgroup means and variance component estimates. A choice of two parameterizations of the variance components for the model can be made.

Scheffé (1959, pages 274–289) discusses the parameterization for MODEL = 1. For example, consider the following model equation with fixed factor *A* and random factor *B*:

$$y_{ijk} = \mu + \alpha_i + b_j + c_{ij} + e_{ijk} \quad i = 1, 2, \dots, a; j = 1, 2, \dots, b; k = 1, 2, \dots, n$$

The fixed effects α_i 's are subject to the restriction

$$\sum_{i=1}^a \alpha_i = 0$$

the b_j 's are random effects identically and independently distributed

$$N(0, \sigma_B^2)$$

c_{ij} are interaction effects each distributed

$$N\left(0, \frac{a-1}{a} \sigma_{AB}^2\right)$$

and are subject to the restrictions

$$\sum_{i=1}^a c_{ij} = 0 \text{ for } j = 1, 2, \dots, b$$

and the e_{ijk} 's are errors identically and independently distributed $N(0, \sigma^2)$. In general, interactions of fixed and random factors have sums over subscripts corresponding to fixed factors equal to zero. Also in general, the variance of a random interaction effect is the associated variance component times a product of ratios for each fixed factor in the random

interaction term. Each ratio depends on the number of levels in the fixed factor. In the earlier example, the random interaction AB has the ratio $(a-1)/a$ as a multiplier of

$$\sigma_{AB}^2$$

and

$$\text{var}(y_{ijk}) = \sigma_B^2 + \frac{a-1}{a} \sigma_{AB}^2 + \sigma^2$$

In a three-way crossed classification model, an ABC interaction effect with A fixed, B random, and C fixed would have variance

$$\frac{(a-1)(c-1)}{ac} \sigma_{ABC}^2$$

Searle (1971, pages 400–401) discusses the parameterization for $\text{MODEL} = 0$. This parameterization does not have the summation restrictions on the effects corresponding to interactions of fixed and random factors. Also, the variance of each random interaction term is the associated variance component, i.e., without the multiplier. This parameterization is also used with unbalanced data, which is one reason for its popularity with balanced data also. In the earlier example,

$$\text{var}(y_{ijk}) = \tilde{\sigma}_B^2 + \tilde{\sigma}_{AB}^2 + \sigma^2$$

Searle (1971, pages 400–404) compares these two parameterizations. Hocking (1973) considers these different parameterizations and concludes they are equivalent because they yield the same variance-covariance structure for the responses. Differences in covariances for individual terms, differences in expected mean square coefficients and differences in F tests are just a consequence of the definition of the individual terms in the model and are not caused by any fundamental differences in the models. For the earlier two-way model, Hocking states that the relations between the two parameterizations of the variance components are

$$\begin{aligned} \sigma_B^2 &= \tilde{\sigma}_B^2 + \frac{1}{a} \tilde{\sigma}_{AB}^2 \\ \sigma_{AB}^2 &= \tilde{\sigma}_{AB}^2 \end{aligned}$$

where

$$\tilde{\sigma}_B^2 \text{ and } \tilde{\sigma}_{AB}^2$$

are the variance components in the parameterization with $\text{MODEL} = 0$.

The computations for degrees of freedom and sums of squares are the same regardless of the option specified by MODEL . `ABALD` first computes degrees of freedom and sum of squares for a full factorial design. Degrees of freedom for effects in the factorial design that are missing from the specified model are pooled into the model effect containing the fewest subscripts but still containing the factorial effect. If no such model effect exists, the factorial effect is pooled into error. If more than one such effect exists, a terminal error message is issued indicating a misspecified model.

The analysis of variance method is used for estimating the variance components. This method solves a linear system in which the mean squares are set to the expected mean squares. A problem that Hocking (1985, pages 324–330) discusses is that this method can yield a negative variance component estimate. Hocking suggests a diagnostic procedure for locating the cause of the negative estimate. It may be necessary to re-examine the assumptions of the model.

The percentage of variation explained by each random effect is computed (output in $VC(i, 7)$) as the variance of the associated random effect divided by the variance of y . The two parameterizations can lead to different values because of the different definitions of the individual terms in the model. For example, the percentage associated with the AB interaction term in the earlier two-way mixed model is computed for $MODEL = 1$ using the formula

$$VC(3,7) = \frac{\frac{a-1}{a} \sigma_{AB}^2}{\sigma_B^2 + \frac{a-1}{a} \sigma_{AB}^2 + \sigma^2}$$

while for the parameterization $MODEL = 0$, the percentage is computed using the formula

$$VC(3,7) = \frac{\tilde{\sigma}_{AB}^2}{\tilde{\sigma}_B^2 + \tilde{\sigma}_{AB}^2 + \sigma^2}$$

In each case, the variance components are replaced by their estimates (stored in $VC(i, 6)$).

Confidence intervals on the variance components are computed using the method discussed by Graybill (1976, Theorem 15.3.5, page 624, and Note 4, page 620). Routine `CIDMS` (page 469) is used.

Additional Examples

Example 2

An analysis of a split-plot design is performed using data discussed by Milliken and Johnson (1984, Table 24.1, page 297). Label the two treatment factors A and C . Denote the treatment combination $a_i c_k$ as that at the i -th level of A and the k -th level of C . The model is

$$y_{ijk} = \mu + \alpha_i + b_j + d_{ij} + \delta_{ik} + e_{ijk} \quad i = 1, 2; j = 1, 2; k = 1, 2, 3, 4$$

where y_{ijk} is the response for the j -th experimental unit with treatment combination $a_i c_k$; the α_i 's are the effects due to treatment factor A , the b_j 's are block effects identically and independently distributed

$$N(0, \sigma_b^2)$$

the d_{ij} are split plot errors that are identically and independently distributed

$$N(0, \sigma_{AB}^2)$$

the γ_k 's are the effects due to treatment factor C , the δ_{ik} 's are interaction effects between factors A and C , and the e_{ijk} 's are identically and independently distributed $N(0, \sigma^2)$. The block effects, whole plot errors, and split plot errors are independent.

The data are given in the following table.

A	Block	C	
		1	2
1	1	35.4	37.9
	2	41.6	40.3
2	1	36.7	38.2
	2	42.7	41.6
3	1	34.8	36.4
	2	43.6	42.8
4	1	39.5	40.0
	2	44.5	47.6

```

USE ABALD_INT
INTEGER LDVC, LINDEF, NEF, NF, NOBS, NRF
PARAMETER (LINDEF=7, NEF=5, NF=3, NOBS=16, NRF=1)
!
INTEGER INDEF(LINDEF), INDRF(NRF), IPRINT, NFEF(NEF), NL(NF)
REAL AOV(15), Y(NOBS)
!
DATA NL/4, 2, 2/
DATA INDRF/2/
DATA NFEF/1, 1, 2, 1, 2/
DATA INDEF/1, 2, 1, 2, 3, 1, 3/
DATA Y/35.4, 37.9, 41.6, 40.3, 36.7, 38.2, 42.7, 41.6, 34.8, &
      36.4, 43.6, 42.8, 39.5, 40.0, 44.5, 47.6/
!
IPRINT = -2
CALL ABALD (NL, Y, NRF, INDRF, NFEF, INDEF, AOV, IPRINT=IPRINT)
END

```

Output

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	95.574	83.401	1.452	40.22	3.609

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Model	11	182.0	16.55	7.852	0.0306
Error	4	8.4	2.11		
Corrected Total	15	190.4			

Source	DF	Sum of Squares	Mean Square	F	Prob. of Larger F
A	3	40.190	13.397	5.802	0.0914
B	1	131.102	131.102	56.775	0.0048
AB	3	6.928	2.309	1.096	0.4476
C	1	2.250	2.250	1.068	0.3599
AC	3	1.550	0.517	0.245	0.8612

```

      * * * EMS * * *
Error AC C AB B A
A      1  0  0  2  0  4
B      1  0  0  2  8
AB     1  0  0  2
C      1  0  8
AC     1  2
Error  1

```

* * * Variance Components * * *

```

                                     95.0% Confidence Interval
Variance
Component      Estimate      Percent      Lower Limit      Upper Limit
B              16.099        87.938        2.2597          16686.7
AB              0.101         0.551         0.0000           15.1
Error          2.108         11.512        0.7565           17.4

```

* * * Subgroup Means * * *

```

A Means (N=4)
1      38.8000
2      39.8000
3      39.4000
4      42.9000
B Means (N=8)
1      37.3625
2      43.0875
C Means (N=8)
1      39.8500
2      40.6000
AB Means (N=2)
1 1      36.6500
1 2      40.9500
2 1      37.4500
2 2      42.1500
3 1      35.6000
3 2      43.2000
4 1      39.7500
4 2      46.0500
AC Means (N=2)
1 1      38.5000
1 2      39.1000
2 1      39.7000
2 2      39.9000
3 1      39.2000
3 2      39.6000
4 1      42.0000
4 2      43.8000
BC Means (N=4)
1 1      36.6000
1 2      38.1250
2 1      43.1000
2 2      43.0750

```

Example 3

An analysis of a split-plot factorial design is performed using data discussed by Kirk (1982, Table 11.2-1, pages 493–496). Label the two treatment factors A and C . Denote the treatment combination $a_i c_k$ as that at the i -th level of A and the k -th level of C . The model is

$$y_{ijk} = \mu + \alpha_i + b_{jj} + \gamma_k + \delta_{ik} + e_{ijk} \quad i = 1, 2; j = 1, 2, 3, 4; k = 1, 2, 3, 4$$

where y_{ijk} is the response for the j -th experimental unit with treatment combination $a_i c_k$; the α_i 's are the effects due to treatment factor A and are subject to the restriction

$$\sum_{i=1}^2 \alpha_i = 0$$

the b_{jj} 's are block effects identically and independently distributed

$$N(0, \sigma_b^2)$$

the γ_k 's are the effects due to treatment factor C and are subject to the restriction

$$\sum_{k=1}^4 \gamma_k = 0$$

the δ_{ik} 's are interaction effects between factors A and C and are subject to the restrictions

$$\sum_{i=1}^2 \delta_{ik} = 0$$

for each k , and

$$\sum_{k=1}^4 \delta_{ik} = 0$$

for each i , and the e_{ijk} 's are identically and independently distributed $N(0, \sigma^2)$. The block effects are assumed to be distributed independently of the errors.

The data are given in the following table:

A	Block	C			
		1	2	3	4
1	1	3	4	7	7
	2	6	5	8	8
	3	3	4	7	9
	4	3	3	6	8
2	5	1	2	5	10
	6	2	3	6	10
	7	2	4	5	9
	8	2	3	6	11

```
USE ABALD_INT
INTEGER LINDEF, NEF, NF, NOBS, NRF
```

```

PARAMETER (LINDEF=6, NEF=4, NF=3, NOBS=32, NRF=-1)
!
INTEGER INDEF(LINDEF), INDRF(-NRF), IPRINT, MODEL, NFEF(NEF), &
NL(NF)
REAL AOV(15), Y(NOBS)
!
DATA NL/2, 4, 4/
DATA INDRF/2/
DATA NFEF/1, 2, 1, 2/
DATA INDEF/1, 1, 2, 3, 1, 3/
DATA Y/3.0, 4.0, 7.0, 7.0, 6.0, 5.0, 8.0, 8.0, 3.0, 4.0, 7.0, 9.0, &
3.0, 3.0, 6.0, 8.0, 1.0, 2.0, 5.0, 10.0, 2.0, 3.0, 6.0, &
10.0, 2.0, 4.0, 5.0, 9.0, 2.0, 3.0, 6.0, 11.0/
!
IPRINT = 1
MODEL = 1
CALL ABALD (NL, Y, NRF, INDRF, NFEF, INDEF, AOV, IPRINT=IPRINT,
MODEL=MODEL)
END

```

Output

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	96.125	93.327	0.712	5.375	13.25

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Model	13	226.4	17.41	34.350	0.0000
Error	18	9.1	0.51		
Corrected Total	31	235.5			

Source	DF	Sum of Squares	Mean Square	F	Prob. of Larger F
A	1	3.125	3.1250	2.000	0.2070
AB	6	9.375	1.5625	3.082	0.0296
C	3	194.500	64.8333	127.890	0.0000
AC	3	19.375	6.4583	12.740	0.0001

* * * EMS * * *

Error	AC	C	AB	A
A	1	0	0	4
AB	1	0	0	4
C	1	0	8	
AC	1	4		
Error	1			

* * * Variance Components * * *

Variance Component	Estimate	Percent	95.0% Confidence Interval	
			Lower Limit	Upper Limit
AB	0.26389	34.234	0.00000	1.7760
Error	0.50694	65.766	0.28944	1.1086

* * * Subgroup Means * * *

A Means (N=16)
1 5.6875

```

2          5.0625
B Means (N=8)
1          4.8750
2          6.0000
1          5.3750
2          5.2500
C Means (N=8)
1          2.7500
2          3.5000
1          6.2500
2          9.0000
AB Means (N=4)
1 1          5.2500
1 2          6.7500
1 3          5.7500
1 4          5.0000
2 1          4.5000
2 2          5.2500
2 3          5.0000
2 4          5.5000
AC Means (N=4)
1 1          3.7500
1 2          4.0000
1 3          7.0000
1 4          8.0000
2 1          1.7500
2 2          3.0000
2 3          5.5000
2 4          10.0000
BC Means (N=2)
1 1          2.0000
1 2          3.0000
1 3          6.0000
1 4          8.5000
2 1          4.0000
2 2          4.0000
2 3          7.0000
2 4          9.0000
1 1          2.5000
1 2          4.0000
1 3          6.0000
1 4          9.0000
2 1          2.5000
2 2          3.0000
2 3          6.0000
2 4          9.5000
ABC Means (N=1)
1 1 1          3.0000
1 1 2          4.0000
1 1 3          7.0000
1 1 4          7.0000
1 2 1          6.0000
1 2 2          5.0000
1 2 3          8.0000
1 2 4          8.0000

```

1	3	1	3.0000
1	3	2	4.0000
1	3	3	7.0000
1	3	4	9.0000
1	4	1	3.0000
1	4	2	3.0000
1	4	3	6.0000
1	4	4	8.0000
2	1	1	1.0000
2	1	2	2.0000
2	1	3	5.0000
2	1	4	10.0000
2	2	1	2.0000
2	2	2	3.0000
2	2	3	6.0000
2	2	4	10.0000
2	3	1	2.0000
2	3	2	4.0000
2	3	3	5.0000
2	3	4	9.0000
2	4	1	2.0000
2	4	2	3.0000
2	4	3	6.0000
2	4	4	11.0000

ANEST

Analyzes a completely nested random model with possibly unequal numbers in the subgroups.

Required Arguments

NF — Number of factors (number of subscripts) in the model including error. (Input)

IEQ — Equal numbers option. (Input)

IEQ Description

0 Unequal numbers in the subgroups

1 Equal numbers in the subgroups

NL — Vector with the number of levels. (Input)

If *IEQ* = 1, *NL* is of length *NF* and contains the number of levels for each of the factors. In this case, the following additional variables are referred to in the description of ANEST:

Variable Description

LNL	$NL(1) + NL(1) * NL(2) + \dots + NL(1) * NL(2) * \dots * NL(NF - 1)$
LNLNF	$NL(1) * NL(2) * \dots * NL(NF - 1)$
NOBS	The number of observations. NOBS equals $NL(1) * NL(2) * \dots * NL(NF)$.

If $IEQ = 0$, NL contains the number of levels of each factor at each level of the factor in which it is nested. In this case, the following additional variables are referred to in the description of ANEST:

Variable Description

LNL	Length of NL.
LNLNF	Length of the subvector of NL for the last factor.
NOBS	Number of observations. NOBS equals the sum of the last LNLNF elements of NL.

For example, a random one-way model with two groups, five responses in the first group and ten in the second group, would have $LNL = 3$, $LNLNF = 2$, $NOBS = 15$, $NL(1) = 2$, $NL(2) = 5$, and $NL(3) = 10$.

Y— Vector of length NOBS containing the responses. (Input)
The elements of Y are ordered lexicographically, i.e., the last model subscript changes most rapidly, the next next to last model subscript changes the next most rapidly, and so forth, with the first subscript changing the slowest.

AOV— Vector of length 15 containing statistics relating to the analysis of variance. (Output)

I	AOV(I)
1	Degrees of freedom for the model
2	Degrees of freedom for error
3	Total (corrected) degrees of freedom
4	Sum of squares for the model
5	Sum of squares for error
6	Total (corrected) sum of squares

- | | |
|----------|---|
| I | AOV(I) |
| 7 | Model mean square |
| 8 | Error mean square |
| 9 | <i>F</i> -statistic |
| 10 | <i>p</i> -value |
| 11 | R^2 |
| 12 | Adjusted R^2 |
| 13 | Estimated standard deviation of the model error |
| 14 | Overall mean of Y |
| 15 | Coefficient of variation (in percent) |

Optional Arguments

CONPER — Confidence level for two-sided interval estimates on the variance components, in percent. (Input)

Default: CONPER = 95.0.

CONPER percent confidence intervals are computed, hence, CONPER must be in the interval [0.0, 100.0). CONPER often will be 90.0, 95.0, or 99.0. For one-sided intervals with confidence level ONECL, ONECL in the interval [50.0, 100.0), set

CONPER = 100.0 - 2.0 * (100.0 - ONECL).

IPRINT — Printing option. (Input)

Default: IPRINT = 0.

IPRINT Action

0 No printing is performed.

1 Printing is performed.

EMS — Vector of length $(NF + 1) * NF/2$ with expected mean square coefficients. (Output)

VC — NF by 9 matrix containing statistics relating to the particular variance components in the model. (Output)

Rows of VC correspond to the NF factors. Columns of VC are as follows:

Column Description

1	Degrees of freedom
2	Sum of squares
3	Mean squares
4	<i>F</i> -statistic
5	<i>p</i> -value for <i>F</i> test
6	Variance component estimate
7	Percent of variance explained by variance component
8	Lower endpoint for a confidence interval on the variance component
9	Upper endpoint for a confidence interval on the variance component

A test for the error variance equal to zero cannot be performed. $VC(NF, 4)$ and $VC(NF, 5)$ are set to NaN (not a number).

LDVC — Leading dimension of VC exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDVC = size(VC, 1)$

YMEANS — Vector containing the subgroup means. (Output)

IEQ Length of YMEANS

0 $1 + NL(1) + NL(2) + \dots + NL(LNL - LNLNF)$ (See the description of argument NL for definitions of LNL and $LNLNF$.)

1 $1 + NL(1) + NL(1) * NL(2) + \dots + NL(1) * NL(2) * \dots * NL(NF - 1)$

If the factors are labeled *A*, *B*, *C*, and error, the ordering of the means is grand mean, *A* means, *AB* means, and then *ABC* means.

NMISS — Number of missing values in Y . (Output)
Elements of Y equal to NaN (not a number) are omitted from the computations.

FORTRAN 90 Interface

Generic: `CALL ANEST (NF, IEQ, NL, Y, AOV [, ...])`

Specific: The specific interface names are `S_ANEST` and `D_ANEST`.

FORTRAN 77 Interface

Single: CALL ANEST (NF, IEQ, NL, Y, CONPER, IPRINT, AOV, EMS, VC,
 LDVC, YMEANS, NMISS)

Double: The double precision name is DANEST.

Example 1

An analysis of a three-factor nested random model with equal numbers in the subgroups is performed using data discussed by Snedecor and Cochran (1967, Table 10.16.1, pages 285–288). The responses are calcium concentrations (in percent, dry basis) as measured in the leaves of turnip greens. Four plants are taken at random, then three leaves are randomly selected from each plant. Finally, from each selected leaf two samples are taken to determine calcium concentration. The model is

$$y_{ijk} = \mu + \alpha_i + \beta_{jj} + e_{ijk} \quad i = 1, 2, 3, 4; j = 1, 2, 3; k = 1, 2$$

where y_{ijk} is the calcium concentration for the k -th sample of the j -th leaf of the i -th plant, the α_i 's are the plant effects and are taken to be independently distributed

$$N(0, \sigma^2)$$

the β_{ij} 's are leaf effects each independently distributed

$$N(0, \sigma_\beta^2)$$

and the ε_{ijk} 's are errors each independently distributed $N(0, \sigma^2)$. The effects are all assumed to be independently distributed. The data are given in the following table:

Plant	Leaf	Samples	
1	1	3.28	3.09
	2	3.52	3.48
	3	2.88	2.80
2	1	2.46	2.44
	2	1.87	1.92
	3	2.19	2.19
3	1	2.77	2.66
	2	3.74	3.44
	3	2.55	2.55
4	1	3.78	3.87
	2	4.07	4.12
	3	3.31	3.31

USE ANEST_INT

```

INTEGER    NF, NOBS
PARAMETER  (NF=3, NOBS=24)
!
INTEGER    IEQ, IPRINT, NL(NF)
REAL       AOV(15), Y(NOBS)
!
DATA Y/3.28, 3.09, 3.52, 3.48, 2.88, 2.80, 2.46, 2.44, 1.87, &
      1.92, 2.19, 2.19, 2.77, 2.66, 3.74, 3.44, 2.55, 2.55, 3.78, &
      3.87, 4.07, 4.12, 3.31, 3.31/
DATA NL/4, 3, 2/
!
IEQ       = 1
IPRINT    = 1
CALL ANEST (NF, IEQ, NL, Y, AOV, IPRINT=IPRINT)
END

```

Output

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	99.222	98.510	0.08158	3.012	2.708

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Model	11	10.19	0.9264	139.216	0.0000
Error	12	0.08	0.0067		
Corrected Total	23	10.27			

Source	DF	Sum of Squares	Mean Square	F	Prob. of Larger F
<i>A</i>	3	7.56034	2.52011	7.665	0.0097
<i>B</i>	8	2.63020	0.32878	49.406	0.0000

* * * Expected Mean Square Coefficients * * *

	Error	Effect B	Effect A
Effect A	1	2	6
Effect B	1	2	
Error	1		

* * * Variance Components * * *

Variance Component	Estimate	Percent	95.0% Confidence Interval	
			Lower Limit	Upper Limit
A	0.36522	68.530	0.039551	5.7867
B	0.16106	30.221	0.069669	0.6004
Error	0.00665	1.249	0.003422	0.0181

A Means

1	3.1750
2	2.1783
3	2.9517
4	3.7433

AB Means

1 1	3.1850
-----	--------

```

1  2      3.5000
1  3      2.8400
2  1      2.4500
2  2      1.8950
2  3      2.1900
3  1      2.7150
3  2      3.5900
3  3      2.5500
4  1      3.8250
4  2      4.0950
4  3      3.3100

```

Comments

Workspace may be explicitly provided, if desired, by use of A2EST/DA2EST. The reference is:

```
CALL A2EST (NF, IEQ, NL, Y, CONPER, IPRINT, AOV, EMS, VC, LDVC,
YMEANS, NMISS, WK, IWK, CHWK)
```

The additional arguments are as follows:

WK — Work vector of length NOBS.

IWK — Work vector of length $5 * NF + (2 * LNL - LNLNF)$.

CHWK — CHARACTER * 10 vector of length $2 * NF + 1$. If IPRINT = 0, CHWK is not referenced and can be a vector of length one.

Description

Routine ANEST analyzes a nested random model with equal or unequal numbers in the subgroups. The analysis includes an analysis of variance table and computation of subgroup means and variance component estimates. Anderson and Bancroft (1952, pages 325–330) discuss the methodology. The analysis of variance method is used for estimating the variance components. This method solves a linear system in which the mean squares are set to the expected mean squares. A problem that Hocking (1985, pages 324–330) discusses is that this method can yield negative variance component estimates. Hocking suggests a diagnostic procedure for locating the cause of a negative estimate. It may be necessary to reexamine the assumptions of the model.

Additional Example

Example 2

An analysis of a three-factor nested random model with unequal numbers in the subgroups is performed. The data are given in the following table:

A	B	C
---	---	---

A	B	C	
1	1	23.0	19.0
	2	31.0	37.0
2	1	33.0	29.0
	2	29.0	
3	1	36.0	29.0 33.0
4	1	11.0	21.0
	2	23.0	18.0
	3	33.0	
	4	23.0	
	5	26.0	
	6	39.0	
	7	20.0	
	8	24.0	
	9	36.0	
5	1	25.0	33.0
6	1	28.0	31.0
	2	25.0	42.0
	3	32.0	36.0
	4	41.0	
	5	35.0	
	6	16.0	
	7	30.0	
	8	40.0	
	9	32.0	
	10	44.0	

```

USE ANEST_INT
INTEGER    LNL, NF, NOBS
PARAMETER (LNL=32, NF=3, NOBS=36)
!
INTEGER    IEQ, IPRINT, NL(LNL)
REAL      AOV(15), Y(NOBS)
!
DATA Y/23.0, 19.0, 31.0, 37.0, 33.0, 29.0, 29.0, 36.0, 29.0, &
      33.0, 11.0, 21.0, 23.0, 18.0, 33.0, 23.0, 26.0, 39.0, 20.0, &
      24.0, 36.0, 25.0, 33.0, 28.0, 31.0, 25.0, 42.0, 32.0, 36.0, &
      41.0, 35.0, 16.0, 30.0, 40.0, 32.0, 44.0/
DATA NL/6, 2, 2, 1, 9, 1, 10, 2, 2, 2, 1, 3, 2, 2, 1, 1, 1, 1, &
      1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1/
!

```

```

IEQ      = 0
IPRINT  = 1
CALL ANEST (NF, IEQ, NL, Y, AOV, IPRINT=IPRINT)
END

```

Output

Dependent Variable	R-squared (percent)	Adjusted R-squared	Est. Std. Dev. of Model Error	Mean	Coefficient of Var. (percent)
Y	85.376	53.470	5.31	29.53	17.98

* * * Analysis of Variance * * *

Source	DF	Sum of Squares	Mean Square	Overall F	Prob. of Larger F
Model	24	1810.8	75.45	2.676	0.0459
Error	11	310.2	28.20		
Corrected Total	35	2121.0			

Source	DF	Sum of Squares	Mean Square	F	Prob. of Larger F
A	5	461.42	92.2845	0.988	0.4601
B	19	1349.38	71.0202	2.519	0.0597

* * * Expected Mean Square Coefficients * * *

	Error	Effect B	Effect A
Effect A	1.00000	1.96503	5.37778
Effect B	1.00000	1.28990	
Error	1.00000		

* * * Variance Components * * *

Variance Component	Estimate	Percent	95.0% Confidence Interval	
			Lower Limit	Upper Limit
A	-0.214	NaN	NaN	NaN
B	33.199	54.073	0.00	100.59
Error	28.197	45.927	14.15	81.29

A Means

1	27.5000
2	30.3333
3	32.6667
4	24.9091
5	29.0000
6	33.2308

AB Means

1	1	21.0000
1	2	34.0000
2	1	31.0000
2	2	29.0000
3	1	32.6667
4	1	16.0000
4	2	20.5000
4	3	33.0000
4	4	23.0000
4	5	26.0000
4	6	39.0000

4	7	20.0000
4	8	24.0000
4	9	36.0000
5	1	29.0000
6	1	29.5000
6	2	33.5000
6	3	34.0000
6	4	41.0000
6	5	35.0000
6	6	16.0000
6	7	30.0000
6	8	40.0000
6	9	32.0000
6	10	44.0000

CTRST

Computes contrast estimates and sums of squares.

Required Arguments

NI — Vector of length *NGROUP* containing the number of responses for each of the *NGROUP* groups. (Input)

YMEANS — Vector of length *NGROUP* containing the sample mean for each group or each level of a classification variable. (Input)

C — *NGROUP* by *NCTRST* matrix containing in each column the coefficients for a particular contrast. (Input)

EST — Vector of length *NCTRST* containing the contrast estimates. (Output)

SS — Vector of length *NCTRST* containing the sum of squares associated with each contrast. (Output)

Optional Arguments

NGROUP — Number of groups or number of sample means involved in the contrasts. (Input)
Default: *NGROUP* = size (*NI*,1).

NCTRST — Number of contrasts. (Input)
Default: *NCTRST* = size (*C*,2).

LDC — Leading dimension of *C* exactly as specified in the dimension statement in the calling program. (Input)
Default: *LDC* = size (*C*,1).

FORTRAN 90 Interface

Generic: CALL CTRST (NI, YMEANS, C, EST, SS [,...])

Specific: The specific interface names are S_CTRST and D_CTRST.

FORTRAN 77 Interface

Single: CALL CTRST (NGROUP, NI, YMEANS, NCTRST, C, LDC, EST, SS)

Double: The double precision name is DCTRST.

Example

The following example is taken from Neter and Wasserman (1974, Table 13.1, page 432, Table 14.3, page 463, pages 470-471). Three orthogonal contrasts are defined that partition the among-group sum of squares (258.0) from a one-way analysis of variance. The first contrast compares groups 1 and 2, the second contrast compares groups 3 and 4, the third contrast compares a weighted average of groups 1 and 2 with a weighted average of groups 3 and 4.

```
USE CTRST_INT
USE UMACH_INT

INTEGER      NGROUP, LDC, NCTRST
PARAMETER   (NGROUP=4, LDC=NGROUP, NCTRST=3)
INTEGER      NI(NGROUP), J, NOUT
REAL         EST(NCTRST), SS(NCTRST), C(LDC,NCTRST), YMEANS(NGROUP)
!
DATA YMEANS/15.0, 13.0, 19.0, 27.0/
DATA NI/2, 3, 3, 2/
DATA (C(I,1),I=1,NGROUP)/1.0, -1.0, 0.0, 0.0/
DATA (C(I,2),I=1,NGROUP)/0.0, 0.0, 1.0, -1.0/
DATA (C(I,3),I=1,NGROUP)/0.4, 0.6, -0.6, -0.4/
!
CALL CTRST (NI, YMEANS, C, EST, SS)
CALL UMACH (2, NOUT)
WRITE (NOUT,*) 'Contrast Estimate Sum of Squares'
DO 10 J=1, NCTRST
   WRITE (NOUT,'(1X,I4,5X,F7.1,3X,F10.1)') J, EST(J), SS(J)
10 CONTINUE
END
```

Output

Contrast	Estimate	Sum of Squares
1	2.0	4.8
2	-8.0	76.8
3	-8.4	176.4

Comments

Informational error

Type Code

1 1 A column of C does not sum to zero within the computed tolerance. Customarily, contrasts (linear combinations of means whose coefficients sum to zero) are of interest.

Description

Routine `CTRST` computes an estimate of a linear combination of means using the sample means input in `YMEANS`. The sum of squares associated with each estimate is also computed.

Contrasts (linear combinations of means whose coefficients sum to zero) are customarily of interest. Orthogonal contrasts (Neter and Wasserman 1974, pages 470–471) are often used to partition the among-groups sum of squares from a one-way analysis of variance. The following discussion uses the term “contrast”, however, the term “linear combination of means,” which places no restriction on the coefficients, is equally valid.

Let

$$\bar{y}_1, \bar{y}_2, \dots, \bar{y}_k$$

be the $k(= \text{NGROUP})$ sample means, and let $\mu_1, \mu_2, \dots, \mu_k$ be the associated population means. Let $c_{1j}, c_{2j}, \dots, c_{kj}$ be the contrast coefficients for contrast j (stored in column j of the matrix C). The estimate of

$$l_j = \sum_{i=1}^k c_{ij} \mu_i$$

is

$$\hat{l}_j$$

(stored as the j -th element of `EST`) computed by

$$\hat{l}_j = \sum_{i=1}^k c_{ij} \bar{y}_i$$

The associated sum of squares Q_j (stored as the j -th element of `SS`) is computed by

$$Q_j = \frac{\hat{l}_j^2}{\sum_{i=1}^k c_{ij}^2 / n_i}$$

SCIPM

Computes simultaneous confidence intervals on all pairwise differences of means.

Required Arguments

NI — Vector of length `NGROUP` containing the number of observations in each mean. (Input)

YMEANS — Vector of length `NGROUP` containing the means. (Input)

DFS2 — Degrees of freedom for s^2 . (Input)

S2 — s^2 , the estimated variance of an observation. (Input)
The variance of `YMEANS(I)` is estimated by $S2/NI(I)$.

STAT — `NGROUP * (NGROUP - 1)/2` by 5 matrix containing the statistics relating to the difference of means. (Output)

Col. Description

- | | |
|---|---|
| 1 | Group number for the i -th mean |
| 2 | Group number for the j -th mean |
| 3 | Difference of means (i -th mean) (j -th mean) |
| 4 | Lower confidence limit for the difference |
| 5 | Upper confidence limit for the difference |

Optional Arguments

NGROUP — Number of means. (Input)
Default: `NGROUP = size(NI,1)`.

IMETH — Method used for constructing confidence intervals on all pairwise differences of means. (Input)
Default: `IMETH = 0`.

IMETH Method

- | | |
|---|---|
| 0 | Tukey (if equal group sizes), Tukey-Kramer method (otherwise) |
| 1 | Dunn-Sidak method |
| 2 | Bonferroni method |
| 3 | Scheffe method |
| 4 | One-at-a-time t method— <i>LSD</i> test |

CONPER — Confidence percentage for the simultaneous interval estimation. (Input)
Default: CONPER = 95.0.

IMETH	CONPER
0	Percentage must be greater than or equal to 90.0 and less than or equal to 99.0.
≥ 1	Percentage must be greater than or equal to 0.0 and less than 100.0.

IPRINT — Printing option. (Input)
Default: IPRINT = 0.

IPRINT	Action
0	No printing is performed.
1	Printing is performed.

LDSTAT — Leading dimension of STAT exactly as specified in the dimension statement in the calling program. (Input)
Default: LDSTAT = size (STAT,1).

FORTRAN 90 Interface

Generic: CALL SCIPM (NI, YMEANS, DFS2, S2, STAT [,...])

Specific: The specific interface names are S_SCIPM and D_SCIPM.

FORTRAN 77 Interface

Single: CALL SCIPM (NGROUP, NI, YMEANS, DFS2, S2, IMETH, CONPER, IPRINT, STAT, LDSTAT)

Double: The double precision name is DSCIPM.

Example

Simultaneous 99% confidence intervals are computed for all pairwise comparisons of 5 means from a one-way analysis of variance design. In order to compare the results of each method, all the options for IMETH are used for input. The data are given by Kirk (1982, Table 3.5-1, page 117). In the output, pairs of means declared not equal are indicated by the letter N. The other pairs of means (for which there is insufficient evidence from the data to declare the means are unequal) are indicated by an equal sign (=).

```
USE SCIPM_INT
INTEGER LDSTAT, NGROUP
PARAMETER (NGROUP=5, LDSTAT=NGROUP*(NGROUP-1)/2)
```

!

```

INTEGER      IMETH, IPRINT, NI (NGROUP)
REAL         CONPER, DFS2, S2, STAT (LDSTAT,5), YMEANS (NGROUP)
!
DATA YMEANS/36.7, 48.7, 43.4, 47.2, 40.3/
DATA NI/10, 10, 10, 10, 10/
!
DFS2   = 45.0
S2     = 28.8
CONPER = 99.0
IPRINT = 1
DO 10  IMETH=0, 4

      CALL SCIPM (NI, YMEANS, DFS2, S2, STAT, IMETH=IMETH, &
                  CONPER=CONPER, IPRINT=IPRINT)
10 CONTINUE
END

```

Output

Simultaneous Confidence Intervals
for All Pairwise Differences of Means
(Tukey Method)

				99.0% Confidence Interval	

	Group I	Group J	Mean I - Mean J	Lower Limit	Upper Limit
N	1	2	-12.0	-20.261	-3.739
=	1	3	-6.7	-14.961	1.561
N	1	4	-10.5	-18.761	-2.239
=	1	5	-3.6	-11.861	4.661
=	2	3	5.3	-2.961	13.561
=	2	4	1.5	-6.761	9.761
N	2	5	8.4	0.139	16.661
=	3	4	-3.8	-12.061	4.461
=	3	5	3.1	-5.161	11.361
=	4	5	6.9	-1.361	15.161

Simultaneous Confidence Intervals
for All Pairwise Differences of Means
(Dunn-Sidak Method)

				99.0% Confidence Interval	

	Group I	Group J	Mean I - Mean J	Lower Limit	Upper Limit
N	1	2	-12.0	-20.445	-3.555
=	1	3	-6.7	-15.145	1.745
N	1	4	-10.5	-18.945	-2.055
=	1	5	-3.6	-12.045	4.845
=	2	3	5.3	-3.145	13.745
=	2	4	1.5	-6.945	9.945
=	2	5	8.4	-0.045	16.845
=	3	4	-3.8	-12.245	4.645
=	3	5	3.1	-5.345	11.545

= 4 5 6.9 -1.545 15.345

Simultaneous Confidence Intervals
for All Pairwise Differences of Means
(Bonferroni Method)

			99.0% Confidence Interval		

	Group I	Group J	Mean I - Mean J	Lower Limit	Upper Limit
N	1	2	-12.0	-20.449	-3.551
=	1	3	-6.7	-15.149	1.749
N	1	4	-10.5	-18.949	-2.051
=	1	5	-3.6	-12.049	4.849
=	2	3	5.3	-3.149	13.749
=	2	4	1.5	-6.949	9.949
=	2	5	8.4	-0.049	16.849
=	3	4	-3.8	-12.249	4.649
=	3	5	3.1	-5.349	11.549
=	4	5	6.9	-1.549	15.349

Simultaneous Confidence Intervals
for All Pairwise Differences of Means
(Scheffe Method)

			99.0% Confidence Interval		

	Group I	Group J	Mean I - Mean J	Lower Limit	Upper Limit
N	1	2	-12.0	-21.317	-2.683
=	1	3	-6.7	-16.017	2.617
N	1	4	-10.5	-19.817	-1.183
=	1	5	-3.6	-12.917	5.717
=	2	3	5.3	-4.017	14.617
=	2	4	1.5	-7.817	10.817
=	2	5	8.4	-0.917	17.717
=	3	4	-3.8	-13.117	5.517
=	3	5	3.1	-6.217	12.417
=	4	5	6.9	-2.417	16.217

Simultaneous Confidence Intervals
for All Pairwise Differences of Means
(One-at-a-Time t Method--LSD Test)

			99.0% Confidence Interval		

	Group I	Group J	Mean I - Mean J	Lower Limit	Upper Limit
N	1	2	-12.0	-18.455	-5.545
N	1	3	-6.7	-13.155	-0.245
N	1	4	-10.5	-16.955	-4.045
=	1	5	-3.6	-10.055	2.855
=	2	3	5.3	-1.155	11.755
=	2	4	1.5	-4.955	7.955
N	2	5	8.4	1.945	14.855
=	3	4	-3.8	-10.255	2.655
=	3	5	3.1	-3.355	9.555
N	4	5	6.9	0.445	13.355

Comments

Workspace may be explicitly provided, if desired, by use of `S2IPM/DS2IPM`. The reference is:

```
CALL S2IPM (NGROUP, NI, YMEANS, DFS2, S2, IMETH, CONPER,  
IPRINT, STAT, LDSTAT, WK, IWK)
```

The additional arguments are as follows:

WK — Real work vector of length `NGROUP`.

IWK — Integer work vector of length `NGROUP`.

Description

Routine `SCIPM` computes simultaneous confidence intervals on all $k^* = k(k-1)/2$ pairwise comparisons of k means $\mu_1, \mu_2, \dots, \mu_k$ in the one-way analysis of variance model. Any of several methods can be chosen. A good review of these methods is given by Stoline (1981). Also the methods are discussed in many elementary statistics texts, e.g., Kirk (1982, pages 114–127).

Let s^2 (input in `S2`) be the estimated variance of a single observation. Let ν be the degrees of freedom (input in `DFS2`) associated with s^2 : Let $\alpha = 1 - \text{CONPER}/100.0$. The methods are summarized as follows:

Tukey method: The Tukey method gives the narrowest simultaneous confidence intervals for all pairwise differences of means $\mu_i - \mu_j$ in balanced ($n_1 = n_2 = \dots = n_k = n$) one-way designs. The method is exact and uses the Studentized range distribution. The formula for the difference $\mu_i - \mu_j$ is given by

$$\bar{y}_i - \bar{y}_j \pm q_{1-\alpha; k, \nu} \sqrt{\frac{s^2}{n}}$$

where $q_{1-\alpha; k, \nu}$ is the $(1 - \alpha)100$ percentage point of the Studentized range distribution with parameters k and ν .

Tukey-Kramer method: The Tukey-Kramer method is an approximate extension of the Tukey method for the unbalanced case. (The method simplifies to the Tukey method for the balanced case.) The method always produces confidence intervals narrower than the Dunn-Sidak and Bonferroni methods. Hayter (1984) proved that the method is conservative, *i.e.*, the method guarantees a confidence coverage of at least $(1 - \alpha)100\%$. Hayter's proof gave further support to earlier recommendations for its use (Stoline 1981). (Methods that are currently better are restricted to special cases and only offer improvement in severely unbalanced cases, see, e.g., Spurrier and Isham 1985). The formula for the difference $\mu_i - \mu_j$ is given by

$$\bar{y}_i - \bar{y}_j \pm q_{1-\alpha; k, \nu} \sqrt{\frac{s^2}{2n_i} + \frac{s^2}{2n_j}}$$

Dunn-Šidák method The Dunn-Šidák method is a conservative method. The method gives wider intervals than the Tukey-Kramer method. (For large ν and small α and k , the difference is

only slight.) The method is slightly better than the Bonferroni method and is based on an improved Bonferroni (multiplicative) inequality (Miller, pages 101, 254–255). The method uses the t distribution (see IMSL routine `TIN`, in Chapter 17, Probability and Distribution Functions and Inverses). The formula for the difference $\mu_i - \mu_j$ is given by

$$\bar{y}_i - \bar{y}_j \pm t_{\frac{1}{2}, \frac{1}{2}(1-\alpha)^{1/k^*}, v} \sqrt{\frac{s^2}{n_i} + \frac{s^2}{n_j}}$$

where $t_{f;v}$ is the 100 f percentage point of the t distribution with v degrees of freedom.

Bonferroni method: The Bonferroni method is a conservative method based on the Bonferroni (additive) inequality (Miller, page 8). The method uses the t distribution. The formula for the difference $\mu_i - \mu_j$ is given by

$$\bar{y}_i - \bar{y}_j \pm t_{1-\alpha/(2k^*)v} \sqrt{\frac{s^2}{n_i} + \frac{s^2}{n_j}}$$

Scheffé method: The Scheffé method is an overly conservative method for simultaneous confidence intervals on pairwise difference of means. The method is applicable for simultaneous confidence intervals on all contrasts, i.e., all linear combinations

$$\sum_{i=1}^k c_i \mu_i \text{ where } \sum_{i=1}^k c_i = 0$$

The method can be recommended here only if a large number of confidence intervals on contrasts in addition to the pairwise differences of means are to be constructed. The method uses the F distribution (see IMSL routine `FIN`, in Chapter 17, Probability and Distribution Functions and Inverses). The formula for the difference $\mu_i - \mu_j$ is given by

$$\bar{y}_i - \bar{y}_j \pm \sqrt{(k-1)F_{1-\alpha; k-1, v} \left(\frac{s^2}{n_i} + \frac{s^2}{n_j} \right)}$$

where $F_{1-\alpha; k-1, v}$ is the $(1-\alpha)100$ percentage point of the F distribution with $k-1$ and v degrees of freedom.

One-at-a-time t method (Fisher's LSD): The one-at-a-time t method is the method appropriate for constructing a single confidence interval. The confidence percentage input is appropriate for one interval at a time. The method has been used widely in conjunction with the overall test of the null hypothesis $\mu_1 = \mu_2 = \dots = \mu_k$ by the use of the F statistic. Fisher's LSD (least significant difference) test is a two-stage test that proceeds to make pairwise comparisons of means only if the overall F test is significant.

Milliken and Johnson (1984, page 31) recommend LSD comparisons after a significant F only if the number of comparisons is small and the comparisons were planned prior to the analysis. If many unplanned comparisons are made, they recommend Scheffé's method. If the F test is insignificant, a few planned comparisons for differences in means can still be performed by using either Tukey, Tukey-Kramer, Dunn-Šidák or Bonferroni methods. Because the F test is insignificant, Scheffé's method will not yield any significant differences. The formula for the difference $\mu_i - \mu_j$ is given by

$$\bar{y}_i - \bar{y}_j \pm t_{1-\frac{\alpha}{2};v} \sqrt{\frac{s^2}{n_i} + \frac{s^2}{n_j}}$$

SNKMC

Performs Student-Newman-Keuls multiple comparison test.

Required Arguments

YMEANS — Vector of length `NGROUP` containing the means. (Input)

SEMEAN — Effective estimated standard error of a mean. (Input)

In fixed effects models, `SEMEAN` equals the estimated standard error of a mean. For example, in a one-way model

$$\text{SEMEAN} = \sqrt{s^2 / n}$$

where s^2 is the estimate of σ^2 and n is the number of responses in a sample mean. In models with random components, use

$$\text{SEMEAN} = \text{SEDIF} / \sqrt{2}$$

where `SEDIF` is the estimated standard error of the difference of two means.

DFSE — Degrees of freedom associated with `SEMEAN`. (Input)

ALPHA — Significance level of test. (Input)

`ALPHA` must be in the interval [0.01, 0.10].

IEQMNS — Vector of length `NGROUP - 1` indicating the size of groups of means declared to be equal. (Output)

`IEQMNS(I) = J` indicates the I -th smallest mean and the next $J - 1$ larger means are declared equal. `IEQMNS(I) = 0` indicates no group of means starts with the I -th smallest mean.

Optional Arguments

NGROUP — Number of groups under consideration. (Input)

Default: `NGROUP = size(YMEANS,1)`.

IPRINT — Printing option. (Input)

Default: `IPRINT = 0`.

IPRINT Action

- 0 No printing is performed.
- 1 Printing is performed.

FORTRAN 90 Interface

Generic: CALL SNKMC (YMEANS, SEMEAN, DFSE, ALPHA, IEQMNS [,...])

Specific: The specific interface names are S_SNKMC and D_SNKMC.

FORTRAN 77 Interface

Single: CALL SNKMC (NGROUP, YMEANS, SEMEAN, DFSE, ALPHA, IPRINT, IEQMNS)

Double: The double precision name is DSNKMC.

Example

A multiple comparisons analysis is performed using data discussed by Kirk (1982, pages 123–125). In the output, means that are not connected by a common underline are declared different.

```
USE UMACH_INT
USE SNKMC_INT

INTEGER IEQMNS(4), IPRINT, N, NOUT
REAL ALPHA, DFSE, S2, SEMEAN, SQRT, YMEANS(5)
INTRINSIC SQRT

!
DATA YMEANS/36.7, 48.7, 43.4, 47.2, 40.3/
!

CALL UMACH (2, NOUT)
S2 = 28.8
N = 10
SEMEAN = SQRT(S2/N)
DFSE = 45.0
ALPHA = .01
IPRINT = 1
CALL SNKMC (YMEANS, SEMEAN, DFSE, ALPHA, IEQMNS, IPRINT=IPRINT)
WRITE (NOUT,99999) IEQMNS
99999 FORMAT (' IEQMNS = ', 4I3)
END
```

Output

Group	1	5	3	4	2
Mean	36.70	40.30	43.40	47.20	48.70

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

IEQMNS = 3 3 3 0

Comments

Workspace may be explicitly provided, if desired, by use of `S2KMC/DS2KMC`. The reference is:

```
CALL S2KMC (NGROUP, YMEANS, SEMEAN, DFSE, ALPHA, IPRINT,  
IEQMNS, WK, IWK)
```

The additional arguments are as follows:

WK — Vector of length `NGROUP` containing `YMEANS` in ascending order. (Output)

IWK — Work vector of length $2 * \text{NGROUP}$.

Description

Routine `SNKMC` performs a multiple comparison analysis of means using the Student-Newman-Keuls method. The null hypothesis is equality of all possible ordered subsets of a set of means. This null hypothesis is tested using the studentized range for each of the corresponding subsets of sample means. The method is discussed in many elementary statistics texts, e.g., Kirk (1982, pages 123–125).

CIDMS

Computes a confidence interval on a variance component estimated as proportional to the difference in two mean squares in a balanced complete experimental design.

Required Arguments

DF1 — Degrees of freedom for effect 1. (Input)

EFMS1 — Mean square for effect 1. (Input)

DF2 — Degrees of freedom for effect 2. (Input)

EFMS2 — Mean square for effect 2. (Input)

VCHAT — Estimated variance component. (Input)

$VCHAT = (EFMS1 - EFMS2)/a$, where a is some positive constant.

CONINT — Vector of length 2 containing the lower and upper endpoints of the confidence interval, respectively. (Output)

Optional Arguments

CONPER — Confidence level for two-sided interval estimate on the variance component, in percent. (Input)

Default: CONPER = 95.0.

A CONPER percent interval is computed, hence, CONPER must be in the interval [0.0, 100.0). CONPER often will be 90.0, 95.0, or 99.0. For a one-sided interval with confidence level ONECL, ONECL in the interval [50.0, 100.0), set

CONPER = 100.0 - 2.0 * (100.0 - ONECL).

IMETH — Method option. (Input)

Default: IMETH = 0.

IMETH	Method
0	Graybill's Method
1	Bross' Method

FORTRAN 90 Interface

Generic: CALL CIDMS (DF1, EFMS1, DF2, EFMS2, VCHAT, CONINT [,...])

Specific: The specific interface names are S_CIDMS and D_CIDMS.

FORTRAN 77 Interface

Single: CALL CIDMS (DF1, EFMS1, DF2, EFMS2, VCHAT, CONPER, IMETH, CONINT)

Double: The double precision name is DCIDMS.

Example

This example computes a confidence interval on a variance component estimated by a difference of mean squares using a nested design discussed by Graybill (1976, pages 635–636). The nested design gave the following analysis of variance table:

Source	DF	MS	EMS
A	5	385.4	$\gamma_1^2 = \sigma^2 + 3\sigma_B^2 + 12\sigma_A^2$
B within A	18	85.4	$\gamma_2^2 = \sigma^2 + 3\sigma_B^2$
Error	48	12.3	$\gamma_3^2 = \sigma^2$

A confidence interval of

$$\sigma_A^2$$

is computed using the method of Graybill. (Note that the lower endpoint of the confidence interval, which is 3.136, is given incorrectly by Graybill [page 636]. Graybill uses an incorrect value for $F_{0.975;5,18}$ in his computations.)

```

USE CIDMS_INT
USE UMACH_INT
INTEGER      NOUT
REAL         CONINT(2), DF1, DF2, EFMS1, EFMS2, VCHAT
!
DF1          = 5.0
EFMS1        = 385.4
DF2          = 18.0
EFMS2        = 85.4
VCHAT        = (EFMS1-EFMS2)/12.0
!
CALL CIDMS (DF1, EFMS1, DF2, EFMS2, VCHAT, CONINT)
!
CALL UMACH (2, NOUT)
WRITE (NOUT,99999) CONINT
99999 FORMAT (' Lower confidence limit', F9.3, '// Upper confidence ', &
             'limit', F9.3)
END

```

Output

```

Lower confidence limit    3.136
Upper confidence limit   186.464

```

Comments

Informational error
Type Code

1 1 One or more endpoints of CI are set to zero.

Description

Routine CIDMS computes a confidence interval on a variance component that has been estimated as proportional to the difference of two mean squares. Let

$$\hat{\gamma}_1^2 \text{ and } \hat{\gamma}_2^2$$

(stored in EFMS1 and EFMS2, respectively) be the two mean squares. The variance component estimate

$$\hat{\sigma}^2$$

(stored in VCHAT) is assumed to be of the form

$$\hat{\sigma}^2 = \frac{\hat{\gamma}_1^2 - \hat{\gamma}_2^2}{a}$$

where a is some positive constant. Two methods for computing a confidence interval on σ^2 can be used. For IMETH = 0, the method discussed by Graybill (1976, Theorem 15.3.5, page 624,

and Note 4, page 620) is used. The result was proposed by Williams (1962). For `IMETH = 1`, the method due to Bross (1950) and discussed by Anderson and Bancroft (1952, page 322) is used.

Routine `CIDMS` can also be used when a variance component is estimated by the difference of two linear combinations of mean squares, each linear combination contains nonnegative coefficients, and the two linear combinations do not use any of the same mean squares. Let

$$\sum_{i=1}^k c_i \hat{\gamma}_i^2 \text{ and } \sum_{i=1}^k d_i \hat{\gamma}_i^2$$

be the two linear combinations (stored in `EFMS1` and `EFMS2`, respectively). The variance component estimate

$$\hat{\sigma}^2$$

(stored in `VCHAT`) is assumed to be of the form

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^k c_i \hat{\gamma}_i^2 - \sum_{i=1}^k d_i \hat{\gamma}_i^2}{a}$$

where a is some positive constant, the c_i 's and d_i 's are nonnegative, and for $i = 1, 2, \dots, k$, $c_i d_i = 0$. Satterthwaite (1946) approximations as discussed by Graybill (1976, pages 642–643) can be used to arrive at approximate degrees of freedom for each linear combination of mean squares for input into `CIDMS`. Let ν_i be the degrees of freedom associated with the i -th mean square

$$\hat{\gamma}_i^2$$

The degrees of freedom stored in `DF1` and `DF2` should be taken to be

$$\frac{(\sum_{i=1}^k c_i \hat{\gamma}_i^2)^2}{\sum_{i=1}^k c_i^2 (\hat{\gamma}_i^2)^2 / \nu_i}$$

and

$$\frac{(\sum_{i=1}^k d_i \hat{\gamma}_i^2)^2}{\sum_{i=1}^k d_i^2 (\hat{\gamma}_i^2)^2 / \nu_i},$$

respectively.

ROREX

Reorders the responses from a balanced complete experimental design.

Required Arguments

NL — Vector of length `NF` containing the number of levels for each of the `NF` factors. (Input)
`NL(I)` is the number of levels for the `I`-th factor.

IORD — Vector of length NF indicating the ordering of the responses in vector **YIN**. (Input)
 $IORD(I) = J$ means the model subscript corresponding to factor I is altering J -th most rapidly.

YIN — Vector of length $NL(1) * NL(2) * \dots * NL(NF)$ containing the responses in the order specified by **IORD**. (Input)

JORD — Vector of length NF indicating the new ordering of the responses in vector **YOUT**. (Input)
 $JORD(K) = L$ means the model subscript corresponding to factor K is altering L -th most rapidly.

YOUT — Vector of length $NL(1) * NL(2) * \dots * NL(NF)$ containing the responses in the order specified by **JORD**. (Output)

Optional Arguments

NF — Number of factors (number of subscripts) in the model, including error. (Input)
Default: $NF = \text{size}(NL, 1)$.

FORTRAN 90 Interface

Generic: `CALL ROREX (NL, IORD, YIN, JORD, YOUT [, ...])`

Specific: The specific interface names are `S_ROREX` and `D_ROREX`.

FORTRAN 77 Interface

Single: `CALL ROREX (NF, NL, IORD, YIN, JORD, YOUT)`

Double: The double precision name is `DROREX`.

Example

The input responses y_{ijk} are ordered in **YIN** so that the subscript i varies most rapidly, j the next most rapidly, and k the least rapidly. Routine `ROREX` is used to reorder the responses into standard order, i.e., with the subscript i varying least rapidly, j the next most rapidly, and k the most rapidly.

```
USE ROREX_INT
USE WRRRL_INT

INTEGER NF, NOBS
PARAMETER (NF=3, NOBS=24)
!
INTEGER IORD(NF), JORD(NF), NL(NF)
REAL YIN(NOBS), YOUT(NOBS)
CHARACTER CLABEL(1)*6, RLABEL(1)*4
DATA CLABEL/'NUMBER' /, RLABEL/'NONE' /
```

```

!
DATA NL/2, 3, 4/
DATA IORD/1, 2, 3/
DATA JORD/3, 2, 1/
DATA YIN/1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, &
      11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, &
      21.0, 22.0, 23.0, 24.0/
!
CALL ROREX (NL, IORD, YIN, JORD, YOUT)
!
CALL WRRRL ('YOUT', YOUT, RLABEL, CLABEL, 1, NOBS, 1, 0, '(F4.1)')
END

```

Output

						YOUT						
1	2	3	4	5	6	7	8	9	10	11	12	13
1.0	7.0	13.0	19.0	3.0	9.0	15.0	21.0	5.0	11.0	17.0	23.0	2.0
14	15	16	17	18	19	20	21	22	23	24		
8.0	14.0	20.0	4.0	10.0	16.0	22.0	6.0	12.0	18.0	24.0		

Comments

Workspace may be explicitly provided, if desired, by use of `R2REX/DR2REX`. The reference is:

```
CALL R2REX (NF, NL, IORD, YIN, JORD, YOUT, IWK)
```

The additional argument is:

IWK — Work vector of length $4 * NF$.

Description

Typically, responses from a balanced complete experimental design are stored in a pattern that takes advantage of the design structure, consequently, the full set of model subscripts is not needed to identify each response. Routine `ROREX` assumes the usual pattern, which requires that one model subscript changes most rapidly, another changes next most rapidly, and so on, throughout the input data vector `YIN`. In many programs, including IMSL programs for this kind of data, the computations and ordering of output are dependent on which subscripts are moving most rapidly relative to others, within the pattern, in the input data. Data may be available in a form that needs reordering within the pattern before entry to an analysis routine. Routine `ROREX` reorders data in `YIN`, as controlled by `JORD`, and returns the reordered data in `YOUT`.

Let k (stored in `NF`) be the number of factors, and for $j = 1, 2, \dots, k$, let n_j (stored as the j -th element of `NL`) be the number of levels in the j -th factor. Let the data in `YIN` be denoted by

$$y_{i_1 i_2 \dots i_k}$$

where for $j = 1, 2, \dots, k$, $i_j = 1, 2, \dots, n_j$. For every response in `YIN`, let p_r denote the model subscript i_j that is altering r -th most rapidly for r and j in the set $\{1, 2, \dots, k\}$. For every response in `YOUT`, let q_s have a similar definition. Let P_r and Q_s equal the number of levels for the factor whose model subscript is altering r -th and s -th most rapidly in `YIN` and `YOUT`, respectively.

The m -th element of Y_{IN} , denoted by

$$y_{p_1 p_2 \dots p_k}$$

with

$$m = p_1 + \sum_{u=2}^k (p_u - 1) \prod_{v=1}^{u-1} p_v$$

can be found using p_1 given by

$$m_1 = m$$

$$p_1 = \begin{cases} m_1 \text{ modulo } P_1 & \text{if } m_1 \text{ modulo } P_1 \text{ is zero} \\ P_1 & \text{if } m_1 \text{ modulo } P_1 \text{ is zero} \end{cases}$$

and then for $r = 2, 3, \dots, k, p_r$ given by

$$m_r = \frac{m_{r-1} - p_{r-1}}{P_{r-1}} + 1,$$

$$p_r = \begin{cases} m_r \text{ modulo } P_r & \text{if } m_r \text{ modulo } P_r \text{ is nonzero} \\ P_r & \text{if } m_r \text{ modulo } P_r \text{ is zero} \end{cases}$$

The m -th element of Y_{OUT} , denoted by

$$y_{q_1 q_2 \dots q_k}$$

is given by replacing the p 's by q 's in the formulas in the preceding equations.

Chapter 5: Categorical and Discrete Data Analysis

Routines

5.1. Statistics in the Two-Way Contingency Table		
Statistics in a 2×2 table.....	CTTWO	480
Chi-squared analysis in a $r \times c$ table	CTCHI	491
Exact probabilities in a $r \times c$ table: total enumeration.....	CTPRB	501
Exact probabilities in a $r \times c$ table: network algorithm	CTEPR	504
5.2. Log-Linear Models		
The iterative proportional fitting algorithm	PRPFT	509
Statistics for a given model.....	CTLLN	513
Parameter estimates for a given model.....	CTPAR	522
Partial association statistics.....	CTASC	528
Hierarchical stepping	CTSTP	536
5.3. Randomization Tests		
Generalized Mantel-Haenszel statistics	CTTRAN	548
5.4. Generalized Categorical Models		
Generalized linear models.....	CTGLM	557
5.5. Weighted Least Squares Analysis		
Analysis by weighted least squares.....	CTWLS	574

Usage Notes

Routines for modeling and analyzing a two- or higher-dimensional contingency table are described in this chapter. Also included are routines for modeling responses from some discrete distributions when discrete or continuous covariates are measured.

The Basic Data Structures

The most common of the three data structures used by the routines in this chapter is a multidimensional (or multi-way) contingency table input as a real vector with length equal to the

product of the number of categories for each dimension. This structure may be obtained from a data matrix X via the routine `FREQ` in Chapter 1, Basic Statistics. Alternatively, multi-way tables may be created and input directly by the user. The multi-way structure is used by all of the log-linear modeling routines (`PRPFT`, [page 509](#); `CTLLN`, [page 513](#); `CTPAR`, [page 522](#); `CTASC`, [page 528](#); and `CTSTP`, [page 536](#)), and is also used in the randomization tests routine, `CTRAN` ([page 548](#)).

A second data structure used by the categorical generalized linear models routine, `CTGLM` ([page 557](#)), is the data matrix X . In `CTGLM` (and elsewhere), if X has many identical rows, at least on the variables of interest, consider using Chapter 1 routine `CSTAT` to add a frequency variable to a reduced matrix X . The transposed output from this routine can replace X as input to `CTGLM`, and `CTGLM` will perform its computations faster (with a linear speed up) on the reduced matrix.

Finally, two-way tables are input into routines `CTCHI` ([page 491](#)), `CTTWO` ([page 480](#)), `CTPRB` ([page 501](#)), `CTEPR` ([page 504](#)), and `CTWLS` ([page 574](#)) as two-dimensional real arrays. As with the multidimensional arrays, two-dimensional arrays may be created via Chapter 1 routine `FREQ`, in which case the leading dimension must equal the number of categories for the first dimension in the table, or they can be created and input directly by the user. Alternatively, the routine `TWFRQ` from Chapter 1 may be used to obtain the two-way frequency table.

Types of Analysis

Routines `CTCHI` ($r \times c$) ([page 491](#)) and `CTTWO` (2×2) (see Chapter 1) compute many statistics of interest in a two-way table. Statistics computed by these routines include the usual chi-squared statistics, measures of association, Kappa, and many others. Asymptotic statistics for a two-way table that are not computed by either `CTCHI` or `CTTWO` can probably be computed by routines `CTRAN` ([page 548](#)) or `CTWLS` ([page 574](#)), but note that these latter two routines require more setup since they require that the user indicate how the statistics are to be computed. Exact probabilities for two-way tables can be computed by `CTPRB` ([page 501](#)), but this routine uses the total enumeration algorithm and, thus, often uses orders of magnitude more computer time than `CTEPR` ([page 504](#)), which computes the same probabilities by use of the network algorithm (but can still be quite expensive).

The routines in the second section are all concerned with hierarchical log-linear models (see, e.g., Bishop, Fienberg, and Holland 1975). The routines in Chapter 1 will often be used to obtain the multi-dimensional tables input into these routines, or the table will be input directly by the user. If the hierarchical is not known, routine `CTASC` ([page 528](#)) will often be the first routine considered. The partial association statistics computed by this routine can be used to obtain a rough estimate of the model to be used. This rough model can then be refined through the use of `CTSTP` ([page 536](#)), which does stepwise model building. Of course, both of these routines are subject to the usual problems associated with building models once the data have been collected: the resulting models may not be correct.

Once a model has been selected (provisional or otherwise), routine `CTLLN` ([page 513](#)) can be used to compute and print many model statistics (parameter estimates, residuals, goodness of fit tests, etc.). If only the parameter estimates and associated variance/covariance matrix are needed, `CTPAR` ([page 522](#)) can be used instead. Both of these routines can compute estimates when sampling and/or structural zeros (cells in the table with observed or restricted counts of zero, respectively) are present in the table, as can all routines in this section.

The algorithm underlying all of the routines in the second section is the iterative proportional fitting algorithm, which is implemented in routine `PRPFT` (page 509). When structural or sampling zeros are present in the table, this algorithm can be quite slow to converge. Also, only the expected cell counts are returned by `PRPFT`, it can be quite difficult to determine degrees of freedom when structural zeros are present in the data. Because a structural zero is a restriction on the parameter space, 1 degree of freedom must be subtracted for each structural zero in the multiway table. The difficulty is in determining where the subtraction should occur. All routines in this section use a Cholesky factorization of $X^T X$ where X is the “design matrix.” This is used to determine which effects should lose degrees of freedom because of structural zeros. Sampling zeros, although they can lead to infinite parameter estimates, do not subtract from the total degrees of freedom. See Clarkson and Jennrich (1991), or Baker, Clarke, and Lane (1985) for details.

Routine `CTRAN` (page 548) computes generalized Mantel-Haenszel statistics in stratified $r \times c$ tables. Generalized Mantel-Haenszel statistics assume that the “direction” of departure from the null hypothesis is consistent from one table to the next. Under this assumption, statistics computed for each table are pooled across all strata yielding a more powerful test than could be obtained otherwise. The statistics computed include measures of correlation, location, and independence using user selected row and/or column scores. Details can be found in (Koch, Amara, and Atkinson 1983) or in the “Algorithm” section for `CTRAN`.

The routine `CTGLM` (page 557) in the fourth section is concerned with generalized linear models (see McCullagh and Nelder 1983) in discrete data. This routine may be used to compute estimates and associated statistics in probit, logistic, minimum extreme value, Poisson, negative binomial (with known number of successes), and logarithmic models. Classification variables as well as weights, frequencies and additive constants may be used so that quite general linear models can be fit. Residuals, a measure of influence, the coefficient estimates, and other statistics are returned for each model fit. When infinite parameter estimates are required, extended maximum likelihood estimation may be used. Log-linear models may be fit in `CTGLM` through the use of Poisson regression models. Results from Poisson regression models involving structural and sampling zeros will be identical to the results obtained from the log-linear model routines but will be fit by a quasi-Newton algorithm rather than through iterative proportional fitting.

The weighted least-squares analysis of Grizzle, Starmer, and Koch (1969) is implemented in routine `CTWLS` (page 574). In this routine, the user first transforms the observed probability estimates (in predefined ways) and then fits a linear model to the transformed estimates using generalized least squares. Multivariate hypotheses associated with the coefficient estimates for the linear model fit may then be tested. In this way, many statistics of interest such as generalized Kappa statistics and parameter estimates in logistic models may be estimated. Of course, the logistic models fit by `CTWLS` use a generalized least-squares criterion rather than the maximum likelihood criterion used to compute the logistic model estimates in `CTGLM`. The generalized least-squares estimates will generally differ somewhat from estimates computed via maximum likelihood.

Other Routines

The routines in Chapter 1, “Basic Statistics,” may be used to create the data structures discussed above. These routines can also create one-dimensional frequency tables, which may then be used by routine `CHIGF` (see Chapter 7, Tests of Goodness of Fit and Randomness), to compute chi-squared goodness-of-fit test statistics or with routines `VHSTP` or `HHSTP` (see Chapter 16, Line

Printer Graphics) to prepare histograms. Routines `CTRHO`, `TETCC`, `BSCAT`, and `BSPBS` (see Chapter 3, Correlation) may be used to compute some measures of correlation in two-way contingency tables.

CTTWO

Performs a chi-squared analysis of a 2 by 2 contingency table.

Required Arguments

TABLE — 2 by 2 matrix containing the observed counts in the contingency table. (Input)

EXPECT — 3 by 3 matrix containing the expected values of each cell in **TABLE** under the null hypothesis of independence, in the first 2 rows and 2 columns, and the marginal totals in the last row and column. (Output)

CHICTR — 3 by 3 matrix containing the contributions to chi-squared for each cell in **TABLE** in the first 2 rows and 2 columns. (Output)
The last row and column contain the total contribution to chi-squared for that row or column.

CHISQ — Vector of length 15 containing statistics associated with this contingency table. (Output)

I	CHISQ(I)
1	Pearson chi-squared statistic
2	Probability of a larger Pearson chi-squared
3	Degrees of freedom for chi-squared
4	Likelihood ratio G^2 (chi-squared)
5	Probability of a larger G^2
6	Yates corrected chi-squared
7	Probability of a larger corrected chi-squared
8	Fisher's exact test (one tail)
9	Fisher's exact test (two tail)
10	Exact mean
11	Exact standard deviation

The following statistics are based upon the chi-squared statistic $\text{CHISQ}(1)$.

- 1 $\text{CHISQ}(1)$
- 12 Phi (Φ)
- 13 The maximum possible Φ
- 14 Contingency coefficient P
- 15 The maximum possible contingency coefficient

STAT— 24 by 5 matrix containing statistics associated with this table. (Output)
Each row of the matrix corresponds to a statistic.

- | Row | Statistic |
|-----|---|
| 1 | Gamma |
| 2 | Kendall's τ_b |
| 3 | Stuart's τ_c |
| 4 | Somers' D (row) |
| 5 | Somers' D (column) |
| 6 | Product moment correlation |
| 7 | Spearman rank correlation |
| 8 | Goodman and Kruskal τ (row) |
| 9 | Goodman and Kruskal τ (column) |
| 10 | Uncertainty coefficient U (normed) |
| 11 | Uncertainty $U_{r c}$ (row) |
| 12 | Uncertainty $U_{c r}$ (column) |
| 13 | Optimal prediction λ (symmetric) |
| 14 | Optimal prediction $\lambda_{r c}$ (row) |
| 15 | Optimal prediction $\lambda_{c r}$ (column) |

Row Statistic

16 Optimal prediction

$\lambda_{r|c}^*$ (row)

17 Optimal prediction

$\lambda_{c|r}^*$ (column)

18 Yule's Q

19 Yule's Y

20 Crossproduct ratio

21 Log of crossproduct ratio

22 Test for linear trend

23 Kappa

24 McNemar test of symmetry

If a statistic is not computed, its value is reported as NaN (not a number). The columns are as follows:

Column Statistic

1 Estimated statistic

2 Its estimated standard error for any parameter value

3 Its estimated standard error under the null hypothesis

4 z -score for testing the null hypothesis

5 p -value for the test in column 4

In the McNemar test, column 1 contains the statistic, column 2 contains the chi-squared degrees of freedom, column 4 contains the exact p -value, and column 5 contains the chi-squared asymptotic p -value.

Optional Arguments

LDTABL — Leading dimension of `TABLE` exactly as specified in the dimension statement of the calling program. (Input)
Default: `LDTABL = size (TABLE,1)`.

ICMPT — Computing option. (Input)
If `ICMPT = 0`, all of the values in `CHISQ` and `STAT` are computed. `ICMPT = 1` means compute only the first 11 values of `CHISQ`, and no values of `STAT` are computed.
Default: `ICMPT = 0`.

IPRINT — Printing option. (Input)
`IPRINT = 0` means no printing is performed. If `IPRINT = 1`, printing is performed.
Default: `IPRINT = 0`.

LDEXPE — Leading dimension of `EXPECT` exactly as specified in the dimension statement of the calling program. (Input)
Default: `LDEXPE = size (EXPECT,1)`.

LDCHIC — Leading dimension of `CHI` exactly as specified in the dimension statement of the calling program. (Input)
Default: `LDCHI = size (CHI,1)`.

LDSTAT — Leading dimension of `STAT` exactly as specified in the dimension statement of the calling program. (Input)
Default: `LDSTAT = size (STAT,1)`.

FORTRAN 90 Interface

Generic: `CALL CTTWO (TABLE, EXPECT, CHICTR, CHISQ, STAT [,...])`

Specific: The specific interface names are `S_CTTWO` and `D_CTTWO`.

FORTRAN 77 Interface

Single: `CALL CTTWO (TABLE, LDTABL, ICMPT, IPRINT, EXPECT, LDEXPE, CHICTR, LDCHIC, CHISQ, STAT, LDSTAT)`

Double: The double precision name is `DCTTWO`.

Example

The following example from Kendall and Stuart (1979, pages 582-583) compares the teeth in breast-fed versus bottle-fed babies.

```
USE CTTWO_INT
INTEGER IPRINT, LDCHIC, LDEXPE, LDSTAT, LDTABL
PARAMETER (IPRINT=1, LDCHIC=3, LDEXPE=3, LDSTAT=24, LDTABL=2)
```



```

!
REAL      CHICTR(LDCHIC,3), CHISQ(15), EXPECT(LDEXPE,3), &
          STAT(LDSTAT,5), TABLE(LDTABL,2)
!
DATA TABLE/4, 1, 16, 21/

!
CALL CTTWO (TABLE, EXPECT, CHICTR, CHISQ, STAT, IPRINT=IPRINT)
END

```

Output

```

TABLE
  1      2
1  4.00 16.00
2  1.00 21.00

```

```

          Expected values
          Col 1      Col 2      Marginal
Row 1      2.3810     17.6190     20.0000
Row 2      2.6190     19.3810     22.0000
Marginal    5.0000     37.0000     42.0000

```

```

          Contributions to chi-squared
          Col 1      Col 2      Total
Row 1      1.1010     0.1488     1.2497
Row 2      1.0009     0.1353     1.1361
Total      2.1018     0.2840     2.3858

```

```

          CHISQ
          1
Pearson chi-squared      2.3858
p-value                  0.1224
Degrees of freedom      1.0000
Likelihood ratio        2.5099
p-value                  0.1131
Yates chi-squared      1.1398
p-value                  0.2857
Fisher (one tail)      0.1435
Fisher (two tail)     0.1745
Exact mean              1.0244
Exact std dev          1.3267
Phi                     0.2383
Max possible phi       0.3855
Contingency coef.     0.2318
Max possible coef.    0.3597

```

```

          STAT
          Statistic      Std err.  Std err. 0      t-value      p-value
Gamma          0.6800      0.3135     0.4395     1.5472     0.1218
Kendall's tau B 0.2383      0.1347     0.1540     1.5472     0.1218
Stuart's tau C  0.1542      0.0997     NaN        1.5472     0.1218
Somers' D row   0.1545      0.0999     0.0999     1.5472     0.1218
Somers' D col   0.3676      0.1966     0.2376     1.5472     0.1218
Correlation     0.2383      0.1347     0.1540     1.5472     0.1218

```

Spearman rank	0.2383	0.1347	0.1540	1.5472	0.1218
GK tau row	0.0568	0.0641	NaN	NaN	NaN
GK tau col	0.0568	0.0609	NaN	NaN	NaN
U normed	0.0565	0.0661	NaN	NaN	NaN
U row	0.0819	0.0935	NaN	NaN	NaN
U col	0.0432	0.0516	NaN	NaN	NaN
Lamda sym	0.1200	0.0779	NaN	NaN	NaN
Lamda row	0.0000	0.0000	NaN	NaN	NaN
Lamda col	0.1500	0.1031	NaN	NaN	NaN
Lamda star row	0.0000	0.0000	NaN	NaN	NaN
Lamda star col	0.1761	0.1978	NaN	NaN	NaN
Yule's Q	0.6800	0.3135	0.4770	1.4255	0.1540
Yule's Y	0.3923	0.2467	0.2385	1.6450	0.1000
Ratio	5.2500	NaN	NaN	NaN	NaN
Log ratio	1.6582	1.1662	0.9540	1.7381	0.0822
Linear trend	-0.1545	0.1001	NaN	-1.5446	0.1224
Kappa	0.1600	0.1572	0.1600	1.0000	0.3173
McNemar	13.2353	1.0000	NaN	0.0000	0.0003

*** WARNING ERROR 11 from CTTWO. Twenty percent of the table expected
 *** values are less than 5.0.

Comments

Informational errors

Type	Code	
4	8	At least one marginal total is zero. The remainder of the analysis cannot proceed.
3	9	Some expected table values are less than 1.0. Some asymptotic p -values may not be good.
3	10	Some expected table values are less than 2.0. Some asymptotic p -values may not be good.
3	11	20% of the table expected values are less than 5.

Description

Routine CTTWO computes statistics associated with 2×2 contingency tables. Always computed are chi-squared tests of independence, expected values based upon the independence assumption, contributions to chi-squared in a test of independence, and row and column marginal totals. Optionally, when ICMPT = 0, CTTWO can compute some measures of association, correlation, prediction, uncertainty, the McNemar test for symmetry, a test for linear trend, the odds and the log odds ratio, and the Kappa statistic.

Other IMSL routines that may be of interest include TETCC in Chapter 3, Correlation (for computing the tetrachoric correlation coefficient) and CTCHI (page 491) in this chapter (for computing statistics in other than 2×2 contingency tables).

Notation

Let x_{ij} denote the observed cell frequency in the ij cell of the table and n denote the total count in the table. Let $p_{ij} = p_{i\bullet}p_{\bullet j}$ denote the predicted cell probabilities (under the null hypothesis of

independence) where $p_{i\bullet}$ and $p_{\bullet j}$ are the row and column relative marginal frequencies, respectively. Next, compute the expected cell counts as $e_{ij} = n p_{ij}$.

Also required in the following are a_{uv} and b_{uv} , $u, v = 1, \dots, n$. Let (r_s, c_s) denote the row and column response of observation s . Then, $a_{uv} = 1, 0,$ or -1 , depending upon whether $r_u < r_v$, $r_u = r_v$, or $r_u > r_v$, respectively. The b_{uv} are similarly defined in terms of the c_s 's.

The Chi-squared Statistics

For each cell of the four cells in the table, the contribution to chi-squared is given as $(x_{ij} - e_{ij})^2/e_{ij}$. The Pearson chi-squared statistic (denoted is χ^2) is computed as the sum of the cell contributions to chi-squared. It has, of course, 1 degree of freedom and tests the null hypothesis of independence, i.e., of $H_0 : p_{ij} = p_{i\bullet}p_{\bullet j}$. Reject the null hypothesis if the computed value of χ^2 is too large.

Compute G^2 , the maximum likelihood equivalent of χ^2 , as

$$-2.0 \sum_{i,j} x_{ij} \ln(x_{ij} / np_{ij})$$

G^2 is asymptotically equivalent to χ^2 and tests the same hypothesis with the same degrees of freedom.

Measures Related to Chi-squared (Phi and the Contingency Coefficient)

Two measures related to chi-squared but which do not depend upon sample size are phi,

$$\phi = \sqrt{\chi^2 / n}$$

and the contingency coefficient,

$$P = \sqrt{\chi^2 / (n + \chi^2)}$$

Since these statistics do not depend upon sample size and are large when the hypothesis of independence is rejected, they may be thought of as measures of association and may be compared across tables with different sized samples. While P has a range between 0.0 and 1.0 for any given table, the upper bound of P is actually somewhat less than 1.0 (see Kendall and Stuart 1979, page 577). In order to understand association within a table, consider also the maximum possible $P(\text{CHISQ}(15))$ and the maximum possible $\phi(\text{CHISQ}(13))$. The significance of both statistics is the same as that of the χ^2 statistic, $\text{CHISQ}(1)$.

The distribution of the χ^2 statistic in finite samples approximates a chi-squared distribution. To compute the expected mean and standard deviation of the χ^2 statistic, Haldane (1939) uses the multinomial distribution with fixed table marginals. The exact mean and standard deviation generally differ little from the mean and standard deviation of the associated chi-squared distribution.

Fisher's exact test

Fisher's exact test is a conservative but uniformly most powerful unbiased test of equal row (or column) cell probabilities in the 2×2 table. In this test, the row and column marginals are

assumed fixed, and the hypergeometric distribution is used to obtain the significance level of the test. A one- or a two-sided test is possible. See Kendall and Stuart (1979, page 582) for a discussion.

Standard Errors and p -values for Some Measures of Association

In rows 1 through 7 of `STAT`, estimated standard errors and asymptotic p -values are reported. Routine `CTTWO` computes these standard errors in two ways. The first estimate, in column 2 of matrix `STAT`, is asymptotically valid for any value of the statistic. The second estimate, in column 3 of `STAT`, is only correct under the null hypothesis of no association. The z -scores in column 4 are computed using this second estimate of the standard errors, and the p -values in column 5 are computed from these z -scores. See Brown and Benedetti (1977) for a discussion and formulas for the standard errors in column 3.

Measures of Association for Ranked Rows and Columns

The measures of association ϕ and P do not require any ordering of the row and column categories. Routine `CTTWO` also computes several measures of association for tables in which the rows and column categories correspond to ranked observations. Two of these measures, the product-moment correlation and the Spearman correlation, are correlation coefficients that are computed using assigned scores for the row and column categories. In the product-moment correlation, this score is the cell index, while in the Spearman rank correlation, this score is the average of the tied ranks of the row or column marginals. Other scores are possible.

Other measures of associations, Gamma, Kendall's τ_b , Stuart's τ_c and Somers' D , are also computed similarly to a correlation coefficient in that the numerator in these statistics in some sense is a "covariance." In fact, these measures differ only in their denominators, their numerators being the "covariance" between the a_{uv} 's and the b_{uv} 's defined earlier. The numerator is computed as

$$\sum_u \sum_v a_{uv} b_{uv}$$

Since the product $a_{uv}b_{uv} = 1$ if both a_{uv} and b_{uv} are 1 or -1 , it is easy to show that the "covariance" is twice the total number of agreements minus the number disagreements between the row and column variables where a disagreement occurs when $a_{uv}b_{uv} = -1$.

Kendall's τ_b is computed as the correlation between the a_{uv} 's and the b_{uv} 's (see Kendall and Stuart 1979, page 583). Stuart suggested a modification to the denominator of τ in which the denominator becomes the largest possible value of the "covariance." This value turns out to be approximately $2n^2$ in 2×2 tables, and this is the value used in the denominator of Stuart's τ_c . For large n , $\tau_c \approx 2\tau_b$.

Gamma can be motivated in a slightly different manner. Because the "covariance" of the a_{uv} 's and the b_{uv} 's can be thought of as two times the number of agreements minus the number of disagreements [$2(A - D)$, where A is the number of agreements and D is the number of disagreements], gamma is motivated as the probability of agreement minus the probability of disagreement, given that either agreement or disagreement occurred. This is just $(A - D)/(A + D)$.

Two definitions of Somers' D are possible, one for rows and a second for columns. Somers' D for rows can be thought of as the regression coefficient for predicting a_{uv} from b_{uv} . Moreover, Somers' D for rows is the probability of agreement minus the probability of disagreement, given that the column variable, b_{uv} , is not zero. Somers' D for columns is defined in a similar manner.

A discussion of all of the measures of association in this section can be found in Kendall and Stuart (1979, starting on page 592).

The crossproduct ratio is also sometimes thought of as a measure of association (see Bishop, Feinberg and Holland 1975, page 14). It is computed as:

$$\frac{p_{11} \cdot p_{22}}{p_{12} \cdot p_{21}}$$

The log of the crossproduct ratio is the log of this quantity.

The Yule's Q and Yule's Y are related to the cross product ratio. They are computed as:

$$Q = \frac{p_{11} \cdot p_{22} - p_{12} \cdot p_{21}}{p_{11} \cdot p_{22} + p_{12} \cdot p_{21}}$$

$$Y = \frac{\sqrt{p_{11} \cdot p_{22}} - \sqrt{p_{12} \cdot p_{21}}}{\sqrt{p_{11} \cdot p_{22}} + \sqrt{p_{12} \cdot p_{21}}}$$

Measures of Prediction and Uncertainty

The Optimal Prediction Coefficients

The measures in this section do not require any ordering of the row or column variables. They are based entirely upon probabilities. Most are discussed in Bishop, Feinberg, and Holland (1975, page 385).

Consider predicting or classifying the column variable for a given value of the row variable. The best classification for each row under the null hypothesis of independence is the column that has the highest marginal probability (and thus the highest probability for the row under the independence assumption). The probability of misclassification is then one minus this marginal probability. On the other hand, if independence is not assumed so that the row and columns variables are dependent, then within each row one would classify the column variables according to the category with the highest row conditional probability. The probability of misclassification for the row is then one minus this conditional probability.

Define the optimal prediction coefficient $\lambda_{c|r}$ for predicting columns from rows as the proportion of the probability of misclassification that is eliminated because the random variables are not independent. It is estimated by:

$$\lambda_{c|r} = \frac{(1 - p_{\bullet m}) - (1 - \sum_i p_{im})}{1 - p_{\bullet m}}$$

where m is the index of the maximum estimated probability in the row (p_{im}) or row margin ($p_{\bullet m}$). A similar coefficient is defined for predicting the rows from the columns. The symmetric version of the optimal prediction λ is obtained by summing the numerators and denominators of

$\lambda_{r|c}$ and $\lambda_{c|r}$ and dividing. Standard errors for these coefficients are given in Bishop, Feinberg, and Holland (1975, page 388).

A problem with the optimal prediction coefficients λ is that they vary with the marginal probabilities. One way to correct for this is to use row conditional probabilities. The optimal prediction λ^* coefficients are defined as the corresponding λ coefficients in which one first adjusts the row (or column) marginals to the same number of observations. This yields

$$\lambda_{c|r}^* = \frac{\sum_i \max_j p_{ji} - \max_j (\sum_i p_{ji})}{R - \max_j \sum_i p_{ji}}$$

where i indexes the rows and j indexes the columns, and p_{ji} is the (estimated) probability of column j given row i .

$$\lambda_{r|c}^*$$

is similarly defined.

Goodman and Kruskal τ

A second kind of prediction measure attempts to explain the proportion of the explained variation of the row (column) measure given the column (row) measure. Define the total variation in the rows to be

$$n/2 - (\sum_i x_{i\cdot}^2)/(2n)$$

This is $1/(2n)$ times the sums of squares of the a_{uv} 's.

With this definition of variation, the Goodman and Kruskal τ coefficient for rows is computed as the reduction of the total variation for rows accounted for by the columns divided by the total variation for the rows. To compute the reduction in the total variation of the rows accounted for by the columns, define the total variation for the rows within column j as

$$q_j = x_{\cdot j} / 2 - \left(\sum_i x_{ij}^2 \right) / (2x_{i\cdot})$$

Define the total variation for rows within columns as the sum of the q_j 's. Consistent with the usual methods in the analysis of variance, the reduction in the total variation is the difference between the total variation for rows and the total variation for rows within the columns.

Goodman and Kruskal's τ columns is similarly defined. See Bishop, Feinberg, and Holland (1975, page 391) for the standard errors.

The Uncertainty Coefficients

The uncertainty coefficient for rows is the increase in the log-likelihood that is achieved by the most general model over the independence model divided by the marginal log-likelihood for the rows. This is given by

$$U_{r|c} = \frac{\sum_{i,j} x_{ij} \log(x_{i\bullet} x_{\bullet j} / (nx_{ij}))}{\sum_i x_{i\bullet} \log(x_{i\bullet} / n)}$$

The uncertainty coefficient for columns is similarly defined. The symmetric uncertainty coefficient contains the same numerator as $U_{r|c}$ and $U_{c|r}$ but averages the denominators of these two statistics. Standard errors for U are given in Brown (1983).

Kruskal-Wallis

The Kruskal-Wallis statistic for rows is a one-way analysis-of-variance-type test that assumes that the column variable is monotonically ordered. It tests the null hypothesis that the row populations are identical, using average ranks for the column variable. This amounts to a test of $H_0 : p_{1\bullet} = p_{2\bullet}$. The Kruskal-Wallis statistic for columns is similarly defined. Conover (1980) discusses the Kruskal-Wallis test.

Test for Linear Trend

The test for a linear trend in the column probabilities assumes that the row variable is monotonically ordered. In this test, the probability for column 1 is predicted by the row index using weighted simple linear regression. The slope is given by

$$\hat{\beta} = \frac{\sum_j x_{\bullet j} (x_{1j} / x_{\bullet j} - x_{1\bullet} / n)(j - \bar{j})}{\sum_j x_{\bullet j} (j - \bar{j})^2}$$

where

$$\bar{j} = \sum_j x_{\bullet j} j / n$$

is the average row index. An asymptotic test that the slope is zero may be obtained as the usual large sample regression test of zero slope.

Kappa

Kappa is a measure of agreement. In the Kappa statistic, the rows and columns correspond to the responses of two judges. The judges agree along the diagonal and disagree off the diagonal.

Let $p_o = p_{11} + p_{22}$ denote the probability that the two judges agree, and let $p_c = p_{1\bullet} p_{\bullet 1} + p_{2\bullet} p_{\bullet 2}$ denote the expected probability of agreement under the independence model. Kappa is then given by $(p_o - p_c) / (1 - p_c)$.

McNemar Test

The McNemar test is also a test of symmetry in square contingency tables. It tests the null hypothesis $H_0 : \theta_{ij} = \theta_{ji}$. The test statistic with 1 degree of freedom is computed as

$$\sum_{i < j} \frac{(x_{ij} - x_{ji})^2}{(x_{ij} + x_{ji})}$$

Its exact probability may be computed via the binomial distribution.

CTCHI

Performs a chi-squared analysis of a two-way contingency table.

Required Arguments

TABLE — `NROW` by `NCOL` matrix containing the observed counts in the contingency table.
(Input)

EXPECT — $(\text{NROW} + 1)$ by $(\text{NCOL} + 1)$ matrix containing the expected values of each cell in `TABLE`, under the null hypothesis, in the first `NROW` rows and `NCOL` columns and the marginal totals in the last row and column. (Output)

CHICTR — $(\text{NROW} + 1)$ by $(\text{NCOL} + 1)$ matrix containing the contributions to chi-squared for each cell in `TABLE` in the first `NROW` rows and `NCOL` columns. (Output)
The last row and column contain the total contribution to chi-squared for that row or column.

CHISQ — Vector of length 10 containing chi-squared statistics associated with this contingency table. (Output)

- | I | CHISQ(I) |
|----------|---|
| 1 | Pearson chi-squared statistic |
| 2 | Probability of a larger Pearson chi-squared |
| 3 | Degrees of freedom for chi-squared |
| 4 | Likelihood ratio G^2 (chi-squared) |
| 5 | Probability of a larger G^2 |
| 6 | Exact mean |
| 7 | Exact standard deviation |

The following statistics are based upon the chi-squared statistic `CHISQ(1)`. If `ICMPT = 1`, NaN (not a number) is reported.

- | I | CHISQ(I) |
|----------|-------------------------|
| 8 | Phi |
| 9 | Contingency coefficient |
| 10 | Cramer's V |

STAT — 23 by 5 matrix containing statistics associated with this table. (Output)
 If `ICMPT = 1`, **STAT** is not referenced and may be a vector of length 1. Each row of the matrix corresponds to a statistic.

Row Statistic

- 1 Gamma
- 2 Kendall's τ_b
- 3 Stuart's τ_c
- 4 Somers' D for rows given columns
- 5 Somers' D for columns given rows
- 6 Product moment correlation
- 7 Spearman rank correlation
- 8 Goodman and Kruskal τ for rows given columns
- 9 Goodman and Kruskal τ for columns given rows
- 10 Uncertainty coefficient U (symmetric)
- 11 Uncertainty $U_{r|c}$ (rows)
- 12 Uncertainty $U_{c|r}$ (columns)
- 13 Optimal prediction λ (symmetric)
- 14 Optimal prediction $\lambda_{r|c}$ (rows)
- 15 Optimal prediction $\lambda_{c|r}$ (columns)
- 16 Optimal prediction
 $\lambda_{r|c}^*$ (rows)
- 17 Optimal prediction
 $\lambda_{c|r}^*$ (columns)
- 18 Test for linear trend in row probabilities if `NROW = 2`. If `NROW` is not 2, a test for linear trend in column probabilities if `NCOL = 2`.

Row Statistic

- 19 Kruskal-Wallis test for no row effect
- 20 Kruskal-Wallis test for no column effect
- 21 Kappa (square tables only)
- 22 McNemar test of symmetry (square tables only)
- 23 McNemar one degree of freedom test of symmetry (square tables only)

If a statistic cannot be computed, its value is reported as NaN (not a number). The columns are as follows:

Column Statistic

- 1 The estimated statistic
- 2 Its standard error for any parameter value
- 3 Its standard error under the null hypothesis
- 4 The t value for testing the null hypothesis
- 5 p -value of the test in column 4

In the McNemar tests, column 1 contains the statistic, column 2 contains the chi-squared degrees of freedom, column 4 contains the exact p -value (one degree of freedom only), and column 5 contains the chi-squared asymptotic p -value. The Kruskal-Wallis test is the same except no exact p -value is computed.

Optional Arguments

NROW — Number of rows in the table. (Input)
Default: $NROW = \text{size}(\text{TABLE},1)$.

NCOL — Number of columns in the table. (Input)
Default: $NCOL = \text{size}(\text{TABLE},2)$.

LDTABL — Leading dimension of **TABLE** exactly as specified in the dimension statement of the calling program. (Input)
Default: $LDTABL = \text{size}(\text{TABLE},1)$.

ICMPT — Computing option. (Input)
If $ICMPT = 0$, all of the values in **CHISQ** and **STAT** are computed. $ICMPT = 1$ means compute only the first 5 values of **CHISQ** and none of the values in **STAT**. (All values

not computed are set to NaN (not a number).
Default: ICMPT = 0.

IPRINT — Printing option. (Input)
IPRINT = 0 means no printing is performed. If IPRINT = 1, printing is performed.
Default: IPRINT = 0.

LDEXPE — Leading dimension of EXPECT exactly as specified in the dimension statement in the calling program. (Input)
Default: LDEXPE = size (EXPECT,1).

LDCHIC — Leading dimension of CHICTR exactly as specified in the dimension statement in the calling program. (Input)
Default: LDCHIC = size (CHICTR,1).

LDSTAT — Leading dimension of STAT exactly as specified in the dimension statement in the calling program. (Input)
Default: LDSTAT = size (STAT,1).

FORTRAN 90 Interface

Generic: CALL CTCHI (TABLE, EXPECT, CHICTR, CHISQ, STAT [,...])

Specific: The specific interface names are S_CTCHI and D_CTCHI.

FORTRAN 77 Interface

Single: CALL CTCHI (NROW, NCOL, TABLE, LDSTAT, ICMPT, IPRINT, EXPECT, LDEXPE, CHICTR, LDCHIC, CHISQ, STAT, LDSTAT)

Double: The double precision name is DCTCHI.

Example

The following example is taken from Kendall and Stuart (1979). It involves the distance vision in the right and left eyes, and especially illustrates the use of Kappa and McNemar tests. Most other test statistics are also computed.

```
USE CTCHI_INT
INTEGER IPRINT, LDSTAT, NCOL, NROW
PARAMETER (IPRINT=1, LDSTAT=23, NCOL=4, NROW=4)
!
REAL CHICTR(NROW+1,NCOL+1), CHISQ(10), EXPECT(NROW+1,NCOL+1), &
STAT(LDSTAT,5), TABLE(NROW,NCOL)
!
DATA TABLE/821, 116, 72, 43, 112, 494, 151, 34, 85, 145, 583, &
106, 35, 27, 87, 331/
```

!

```
CALL CTCHI (TABLE, EXPECT, CHICTR, CHISQ, STAT, IPRINT=IPRINT)
END
```

Output

Table Values

	1	2	3	4
1	821.0	112.0	85.0	35.0
2	116.0	494.0	145.0	27.0
3	72.0	151.0	583.0	87.0
4	43.0	34.0	106.0	331.0

Expected Values

row totals in column 5, column totals in row 5

	1	2	3	4	5
1	341.69	256.92	298.49	155.90	1053.00
2	253.75	190.80	221.67	115.78	782.00
3	289.77	217.88	253.14	132.21	893.00
4	166.79	125.41	145.70	76.10	514.00
5	1052.00	791.00	919.00	480.00	3242.00

Contributions to Chi-squared

row totals in column 5, column totals in row 5

	1	2	3	4	5
1	672.36	81.74	152.70	93.76	1000.56
2	74.78	481.84	26.52	68.08	651.21
3	163.66	20.53	429.85	15.46	629.50
4	91.87	66.63	10.82	853.78	1023.10
5	1002.68	650.73	619.88	1031.08	3304.37

Chi-square Statistics

Pearson 3304.3682

p-value 0.0000

DF 9.0000

G**2 2781.0188

p-value 0.0000

Exact mean 9.0028

Exact std. 4.2402

Phi 1.0096

P 0.7105

Cramer's V 0.5829

Table Statistics

	statistic	standard error	std. error under Ho	t-value testing Ho	p-value
Gamma	0.7757	0.0123	0.0149	52.19	0.0000
Tau B	0.6429	0.0122	0.0123	52.19	0.0000
Tau C	0.6293	0.0121	NaN	52.19	0.0000
D-Row	0.6418	0.0122	0.0123	52.19	0.0000
D-Column	0.6439	0.0122	0.0123	52.19	0.0000
Correlation	0.6926	0.0128	0.0172	40.27	0.0000
Spearman	0.6939	0.0127	0.0127	54.66	0.0000
GK tau rows	0.3420	0.0123	NaN	NaN	NaN
GK tau col.	0.3430	0.0122	NaN	NaN	NaN

U - Sym.	0.3171	0.0110	NaN	NaN	NaN
U - rows	0.3178	0.0110	NaN	NaN	NaN
U - cols.	0.3164	0.0110	NaN	NaN	NaN
Lambda-sym.	0.5373	0.0124	NaN	NaN	NaN
Lambda-row	0.5374	0.0126	NaN	NaN	NaN
Lambda-col.	0.5372	0.0126	NaN	NaN	NaN
l-star-rows	0.5506	0.0136	NaN	NaN	NaN
l-star-col.	0.5636	0.0127	NaN	NaN	NaN
Lin. trend	NaN	NaN	NaN	NaN	NaN
Kruskal row	1561.4861	3.0000	NaN	NaN	0.0000
Kruskal col	1563.0300	3.0000	NaN	NaN	0.0000
Kappa	0.5744	0.0111	0.0106	54.36	0.0000
McNemar	4.7625	6.0000	NaN	NaN	0.5746
McNemar df=1	0.9487	1.0000	NaN	0.35	0.3301

Comments

Informational errors

Type Code

- 3 1 Twenty percent of the expected values are less than 5.
- 3 2 The degrees of freedom for chi-squared are greater than 30. The exact mean, standard deviation, and normal distribution function should be used.
- 3 3 Some expected values are less than 2. Some asymptotic p -values may not be good.
- 3 4 Some expected values are less than 1. Some asymptotic p -values may not be good.

Description

Routine `CTCHI` computes statistics associated with an $r \times c$ (`NROW` \times `NCOL`) contingency table. The routine `CTCHI` always computes the chi-squared test of independence, expected values, contributions to chi-squared, and row and column marginal totals. Optionally, when `ICMPT` = 0, `CTCHI` can compute some measures of association, correlation, prediction, uncertainty, the McNemar test for symmetry, a test for linear trend, the odds and the log odds ratio, and the Kappa statistic.

Other IMSL routines that may be of interest include `TETCC` in Chapter 3, for computing the tetrachoric correlation coefficient, `CTTWO` (page 480), for computing statistics in a 2×2 contingency table, and `CTPRB` (page 501), for computing the exact probability of an $r \times c$ contingency table.

Notation

Let x_{ij} denote the observed cell frequency in the ij cell of the table and n denote the total count in the table. Let $p_{ij} = p_{i\bullet}p_{\bullet j}$ denote the predicted cell probabilities under the null hypothesis of

independence where $p_{i\bullet}$ and $p_{\bullet j}$ are the row and column marginal relative frequencies, respectively. Next, compute the expected cell counts as $e_{ij} = n p_{ij}$.

Also required in the following are a_{uv} and b_{uv} , $u, v = 1, \dots, n$. Let (r_s, c_s) denote the row and column response of observation s . Then, $a_{uv} = 1, 0$, or -1 , depending upon whether $r_u < r_v, r_u = r_v$, or $r_u > r_v$, respectively. The b_{uv} are similarly defined in terms of the c_s 's.

The Chi-squared Statistics

For each cell in the table, the contribution to χ^2 is given as $(x_{ij} - e_{ij})^2/e_{ij}$. The Pearson chi-squared statistic (denoted χ^2) is computed as the sum of the cell contributions to chi-squared. It has $(r - 1)(c - 1)$ degrees of freedom and tests the null hypothesis of independence, i.e., that $H_0 : p_{ij} = p_{i\bullet}p_{\bullet j}$. The null hypothesis is rejected if the computed value of χ^2 is too large.

Compute G^2 , the maximum likelihood equivalent of χ^2 , as

$$G^2 = -2 \sum_{i,j} x_{ij} \ln(x_{ij} / np_{ij})$$

G^2 is asymptotically equivalent to χ^2 and tests the same hypothesis with the same degrees of freedom.

Measures Related to Chi-squared (Phi, Contingency Coefficient, and Cramer's V)

Three measures related to chi-squared but that do not depend upon the sample size are phi,

$$\phi = \sqrt{\chi^2 / n}$$

the contingency coefficient,

$$P = \sqrt{\chi^2 / (n + \chi^2)}$$

and Cramer's V ,

$$V = \sqrt{\chi^2 / (n \min(r, c))}$$

Since these statistics do not depend upon sample size and are large when the hypothesis of independence is rejected, they may be thought of as measures of association and may be compared across tables with different sized samples. While both P and V have a range between 0.0 and 1.0, the upper bound of P is actually somewhat less than 1.0 for any given table (see Kendall and Stuart 1979, page 587). The significance of all three statistics is the same as that of the χ^2 statistic, `CHISQ(1)`.

The distribution of the χ^2 statistic in finite samples approximates a chi-squared distribution. To compute the exact mean and standard deviation of the χ^2 statistic, Haldane (1939) uses the multinomial distribution with fixed table marginals. The exact mean and standard deviation generally differ little from the mean and standard deviation of the associated chi-squared distribution.

Standard Errors and p -values For Some Measures of Association

In rows 1 through 7 of `STAT`, estimated standard errors and asymptotic p -values are reported. Estimates of the standard errors are computed in two ways. The first estimate, in column 2 of matrix `STAT`, is asymptotically valid for any value of the statistic. The second estimate, in column 3 of the matrix, is only correct under the null hypothesis of no association. The z -scores in column 4 of matrix `STAT` are computed using this second estimate of the standard errors. The p -values in column 5 are computed from this z -score. See Brown and Benedetti (1977) for a discussion and formulas for the standard errors in column 3.

Measures of Association for Ranked Rows and Columns

The measures of association, ϕ , P , and V , do not require any ordering of the row and column categories. Routine `CTCHI` also computes several measures of association for tables in which the rows and column categories correspond to ranked observations. Two of these tests, the product-moment correlation and the Spearman correlation, are correlation coefficients computed using assigned scores for the row and column categories. The cell indices are used for the product-moment correlation while the average of the tied ranks of the row and column marginals is used for the Spearman rank correlation. Other scores are possible.

Gamma, Kendall's τ_b , Stuart's τ_c , and Somers' D are measures of association that are computed like a correlation coefficient in the numerator. In all of these measures, the numerator is computed as the "covariance" between the a_{uv} 's and b_{uv} 's defined above, i.e., as

$$\sum_u \sum_v a_{uv} b_{uv}$$

Recall that a_{uv} and b_{uv} can take values -1 , 0 , or 1 . Since the product $a_{uv}b_{uv} = 1$ only if a_{uv} and b_{uv} are both 1 or are both -1 , it is easy to show that this "covariance" is twice the total number of agreements minus the number of disagreements where a disagreement occurs when $a_{uv}b_{uv} = -1$.

Kendall's τ_b is computed as the correlation between the a_{uv} 's and the b_{uv} 's (see Kendall and Stuart 1979, page 593). In a rectangular table ($r \neq c$), Kendall's τ_b cannot be 1.0 (if all marginal totals are positive). For this reason, Stuart suggested a modification to the denominator of τ in which the denominator becomes the largest possible value of the "covariance." This maximizing value is approximately $n^2m/(m-1)$, where $m = \min(r, c)$. Stuart's τ_c uses this approximate value in its denominator. For large n , $\tau_c \approx m\tau_b/(m-1)$.

Gamma can be motivated in a slightly different manner. Because the "covariance" of the a_{uv} 's and the b_{uv} 's can be thought of as twice the number of agreements minus the disagreements, $(2(A - D))$, where A is the number of agreements and D is the number of disagreements, gamma is motivated as the probability of agreement minus the probability of disagreement, given that either agreement or disagreement occurred. This is just $\gamma = (A - D)/(A + D)$.

Two definitions of Somers' D are possible, one for rows and a second for columns. Somers' D for rows can be thought of as the regression coefficient for predicting a_{uv} from b_{uv} . Moreover, Somers' D for rows is the probability of agreement minus the probability of disagreement, given that the column variable, b_{uv} , is not zero. Somers' D for columns is defined in a similar manner.

A discussion of all of the measures of association in this section can be found in Kendall and Stuart (1979, starting on page 592).

Measures of Prediction and Uncertainty

The Optimal Prediction Coefficients

The measures in this section do not require any ordering of the row or column variables. They are based entirely upon probabilities. Most are discussed in Bishop, Feinberg, and Holland (1975, page 385).

Consider predicting (or classifying) the column for a given row in the table. Under the null hypothesis of independence, one would choose the column with the highest column marginal probability for all rows. In this case, the probability of misclassification for any row is one minus this marginal probability. If independence is not assumed, then within each row one would choose the column with the highest row conditional probability, and the probability of misclassification for the row becomes one minus this conditional probability.

Define the optimal prediction coefficient $\lambda_{c|r}$ for predicting columns from rows as the proportion of the probability of misclassification that is eliminated because the random variables are not independent. It is estimated by

$$\lambda_{c|r} = \frac{(1 - p_{\bullet m}) - (1 - \sum_i p_{im})}{1 - p_{\bullet m}}$$

where m is the index of the maximum estimated probability in the row (p_{im}) or row margin ($p_{\bullet m}$). A similar coefficient is defined for predicting the rows from the columns. The symmetric version of the optimal prediction λ is obtained by summing the numerators and denominators of $\lambda_{r|c}$ and $\lambda_{c|r}$ and by dividing. Standard errors for these coefficients are given in Bishop, Feinberg, and Holland (1975, page 388).

A problem with the optimal prediction coefficients λ is that they vary with the marginal probabilities. One way to correct for this is to use row conditional probabilities. The optimal prediction λ^* coefficients are defined as the corresponding λ coefficients in which one first adjusts the row (or column) marginals to the same number of observations. This yields

$$\lambda_{c|r}^* = \frac{\sum_i \max_j p_{ji} - \max_j (\sum_i p_{ji})}{R - \max_j \sum_i p_{ji}}$$

where i indexes the rows, j indexes the columns, and p_{ji} is the (estimated) probability of column j given row i .

$$\lambda_{r|c}^*$$

is similarly defined.

Goodman and Kruskal τ

A second kind of prediction measure attempts to explain the proportion of the explained variation of the row (column) measure given the column (row) measure. Define the total variation in the rows to be

$$n/2 - (\sum_i x_{i\bullet}^2)/(2n)$$

Note that this is $1/(2n)$ times the sums of squares of the a_{uv} 's.

With this definition of variation, the Goodman and Kruskal τ coefficient for rows is computed as the reduction of the total variation for rows accounted for by the columns, divided by the total variation for the rows. To compute the reduction in the total variation of the rows accounted for by the columns, note that the total variation for the rows within column j is defined as

$$q_j = x_{\bullet j} / 2 - (\sum_i x_{ij}^2)/(2x_{i\bullet})$$

The total variation for rows within columns is the sum of the q_j 's. Consistent with the usual methods in the analysis of variance, the reduction in the total variation is given as the difference between the total variation for rows and the total variation for rows within the columns.

Goodman and Kruskal's τ for columns is similarly defined. See Bishop, Feinberg, and Holland (1975, page 391) for the standard errors.

The Uncertainty Coefficients

The uncertainty coefficient for rows is the increase in the log-likelihood that is achieved by the most general model over the independence model, divided by the marginal log-likelihood for the rows. This is given by

$$U_{r|c} = \frac{\sum_{i,j} x_{ij} \log(x_{i\bullet} x_{\bullet j} / (n x_{ij}))}{\sum_i x_{i\bullet} \log(x_{i\bullet} / n)}$$

The uncertainty coefficient for columns is similarly defined. The symmetric uncertainty coefficient contains the same numerator as $U_{r|c}$ and $U_{c|r}$ but averages the denominators of these two statistics. Standard errors for U are given in Brown (1983).

Kruskal-Wallis

The Kruskal-Wallis statistic for rows is a one-way analysis-of-variance-type test that assumes the column variable is monotonically ordered. It tests the null hypothesis that no row populations are identical, using average ranks for the column variable. The Kruskal-Wallis statistic for columns is similarly defined. Conover (1980) discusses the Kruskal-Wallis test.

Test for Linear Trend

When there are two rows, it is possible to test for a linear trend in the row probabilities if one assumes that the column variable is monotonically ordered. In this test, the probabilities for row 1 are predicted by the column index using weighted simple linear regression. This slope is given by

$$\hat{\beta} = \frac{\sum_j x_{\bullet j} (x_{1j} / x_{\bullet j} - x_{1\bullet} / n) (j - \bar{j})}{\sum_j x_{\bullet j} (j - \bar{j})^2}$$

where

$$\bar{j} = \sum_j x_{\bullet j} j / n$$

is the average column index. An asymptotic test that the slope is zero may then be obtained (in large samples) as the usual regression test of zero slope.

In two-column data, a similar test for a linear trend in the column probabilities is computed. This test assumes that the rows are monotonically ordered.

Kappa

Kappa is a measure of agreement computed on square tables only. In the Kappa statistic, the rows and columns correspond to the responses of two judges. The judges agree along the diagonal and disagree off the diagonal. Let

$$p_o = \sum_i x_{ii} / n$$

denote the probability that the two judges agree, and let

$$p_c = \sum_i e_{ii} / n$$

denote the expected probability of agreement under the independence model. Kappa is then given by $(p_o - p_c) / (1 - p_c)$.

McNemar Tests

The McNemar test is a test of symmetry in a square contingency table, that is, it is a test of the null hypothesis $H_o : \theta_{ij} = \theta_{ji}$. The multiple-degrees-of-freedom version of the McNemar test with $r(r - 1)/2$ degrees of freedom is computed as

$$\sum_{i < j} \frac{(x_{ij} - x_{ji})^2}{(x_{ij} + x_{ji})}$$

The single-degree-of-freedom test assumes that the differences $x_{ij} - x_{ji}$ are all in one direction. The single-degree-of-freedom test will be more powerful than the multiple-degrees-of-freedom test when this is the case. The test statistic is given as

$$\frac{(\sum_{i < j} (x_{ij} - x_{ji}))^2}{\sum_{i < j} (x_{ij} + x_{ji})}$$

Its exact probability may be computed via the binomial distribution.

CTPRB

Computes exact probabilities in a two-way contingency table.

Required Arguments

TABLE — *NROW* by *NCOL* matrix containing the contingency table cell frequencies. (Input)

PRE — Probability of a more extreme table where “extreme” is taken in the Neyman-Pearson sense. (Output)
A table is more extreme if its probability (for fixed marginals) is less than or equal to *PRT*.

Optional Arguments

NROW — Number of rows in the contingency table. (Input)
Default: *NROW* = size (*TABLE*,1).

NCOL — Number of columns in the contingency table. (Input)
Default: *NCOL* = size (*TABLE*,2).

LDTABL — Leading dimension of *TABLE* exactly as specified in the dimension statement in the calling program. (Input)
Default: *LDTABL* = size (*TABLE*,1).

PRT — Probability of the observed table assuming fixed row and column marginal totals. (Output)

PCHEK — Sum of the probabilities of all tables with the same marginal totals. (Output)
PCHEK should be 1.0. Deviation from 1.0 indicates a numerical error.

FORTRAN 90 Interface

Generic: CALL CTPRB (TABLE, PRE [,...])

Specific: The specific interface names are S_CTPRB and D_CTPRB.

FORTRAN 77 Interface

Single: CALL CTPRB (NROW, NCOL, TABLE, LDTABL, PRT, PRE, PCHEK)

Double: The double precision name is DCTPRB.

Example

In this example, *CTPRB* is used to compute the exact conditional probability for a 2×2 contingency table. The input table is given as:

$$\begin{bmatrix} 8 & 12 \\ 8 & 2 \end{bmatrix}$$

```

USE UMACH_INT
USE CTPRB_INT
INTEGER    NCOL, LDTABL
PARAMETER  (NCOL=2, LDTABL=2)
!
INTEGER    NOUT
REAL       PCHEK, PRE, PRT, TABLE(LDTABL,NCOL)
!
DATA TABLE/8, 8, 12, 2/
!
CALL UMACH (2, NOUT)
!
CALL CTPRB (TABLE, PRE, PRT=PRT, PCHEK=PCHEK)
!
WRITE(NOUT, '( " PRT = ', F12.4, ', ', ' PRE = ', F12.4, ', ', &
& ' PCHEK = ', F10.4) ' ) PRT, PRE, PCHEK
END

```

Output

```

PRT =      0.0390
PRE =      0.0577
PCHEK =    1.0000

```

Comments

1. Workspace may be explicitly provided, if desired, by use of C2PRB/DC2PRB. The reference is:

```
CALL C2PRB (NROW, NCOL, TABLE, LDTABL, PRT, PRE, PCHCK, IWK)
```

The additional argument is:

IWK — Work vector of length $(NROW + 2)(NCOL + 2)$.

2. Informational error

Type	Code	
3	1	There are no observed counts in TABLE. PRE, PRT, and PCHEK are set to NaN (not a number).
3. Routine CTPRB computes a two-tailed Fisher exact probability in 2 by 2 tables. For one-tailed Fisher exact probabilities, use routine CTTWO ([page 480](#)).

Description

Routine CTPRB computes exact probabilities for an $r \times c$ contingency table for fixed row and column marginals where $r = NROW$ and $c = NCOL$. Let f_{ij} denote the element in row i and column j of a table, and let $f_{i\bullet}$ and $f_{\bullet j}$ denote the row and column marginals. Under the independence hypothesis, the (conditional) probability for fixed marginals of a table is given by

$$P_f = \frac{\prod_{i=1}^r f_{i\bullet}! \prod_{j=1}^c f_{\bullet j}!}{f_{\bullet\bullet}! \prod_{i=1}^r \prod_{j=1}^c f_{ij}!}$$

where $f_{\bullet\bullet}$ is the total number of counts in the table and $x!$ denotes x factorial. When the f_{ij} are obtained from the input table ($f_{ij} = \text{TABLE}(i, j)$), $P_f = \text{PRT}$. PRE is the sum over all more extreme tables of the probability of each table.

In `CTPRB`, a more extreme table is defined in the probabilistic sense. Table X is more extreme than the input table if the conditional probability computed for table X (for the same marginal sums) is less than the conditional probability computed for the input table. The user should note that this definition of “more extreme” can be considered as “two-sided” in the cell counts.

Because `CTPRB` uses total enumeration in computing the probability of a more extreme table, the amount of computer time required increases very rapidly with the size of the table. Tables, with either a large total count $f_{\bullet\bullet}$ or in which the product rc is not small, should not be analyzed with `CTPRB`. Rather, either the approximate methods of Agresti, Wackerly, and Boyett (1979) should be used or algorithms that do not require total enumeration should be used (see Pagano and Halvorsen [1981], or Mehta and Patel [1983]).

CTEPR

Computes Fisher’s exact test probability and a hybrid approximation to the Fisher exact test probability for a contingency table using the network algorithm.

Required Arguments

TABLE — `NROW` by `NCOL` matrix containing the contingency table. (Input)

PRE — Table p -value. (Output)

PRE is the probability of a more extreme table, where “extreme” is in a probabilistic sense. If `EXPECT < 0`, then the Fisher exact probability is returned. Otherwise, a hybrid approximation to Fisher’s exact probability is computed.

Optional Arguments

NROW — The number of rows in the table. (Input)

Default: `NROW = size (TABLE,1)`.

NCOL — The number of columns in the table. (Input)

Default: `NCOL = size (TABLE,2)`.

LDTABL — Leading dimension of `TABLE` exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDTABL = size (TABLE,1)`.

EXPECT — Expected value used in the hybrid approximation to Fisher’s exact test algorithm for deciding when to use asymptotic probabilities when computing path lengths.

(Input)

Default: `EXPECT = 5.0`.

If `EXPECT ≤ 0.0`, then asymptotic theory probabilities are not used and Fisher exact test probabilities are computed. Otherwise, asymptotic probabilities are used in computing path lengths whenever `PERCNT` or more of the cells in the table for which path lengths are to be computed have estimated expected values of `EXPECT` or more, with no cell having expected value less than `EMIN`. See the “Algorithm” section for details. Use `EXPECT = 5.0` to obtain the “Cochran” condition.

PERCNT — Percentage of remaining cells that must have estimated expected values greater than `EXPECT` before asymptotic probabilities can be used in computing path lengths.

(Input)

Default: `PERCNT = 80.0`.

See argument `EXPECT` for details. Use `PERCNT = 80.0` to obtain the “Cochran” condition.

EMIN — Minimum cell estimated expected value allowed for asymptotic chi-squared probabilities to be used. (Input)

Default: `EMIN = 1.0`.

See argument `EXPECT` for details. Use `EMIN = 1.0` to obtain the “Cochran” condition.

PRT — Probability of the observed table for fixed marginal totals. (Output)

FORTRAN 90 Interface

Generic: `CALL CTEPR (TABLE, PRE [, ...])`

Specific: The specific interface names are `S_CTEPR` and `D_CTEPR`.

FORTRAN 77 Interface

Single: `CALL CTEPR (NROW, NCOL, TABLE, LD_TBL, EXPECT, PERCNT, EMIN, PRT, PRE)`

Double: The double precision name is `DCTEPR`.

Example

In this example, `CTEPR` is used to compute the hybrid approximation to the Fisher exact probability for a 3×6 contingency table using the Cochran condition. Because of the large initial counts and the input arguments `EXPECT = 5`, `PERCNT = 80`, and `EMIN = 1`, the hybrid algorithm significantly reduces the computation effort in this example. The input table is given as

$$\begin{bmatrix} 20 & 20 & 0 & 0 & 0 \\ 10 & 10 & 2 & 2 & 1 \\ 20 & 20 & 0 & 0 & 0 \end{bmatrix}$$

```

USE UMACH_INT
USE CTEPR_INT

INTEGER    LDTABL, NCOL
PARAMETER (NCOL=5, LDTABL=3)
!
INTEGER    NOUT
REAL      PRE, PRT, TABLE(LDTABL,NCOL)
!
DATA TABLE/20.0, 10.0, 20.0, 20.0, 10.0, 20.0, 0.0, 2.0, 0.0, &
        0.0, 2.0, 0.0, 0.0, 1.0, 0.0/
!
CALL UMACH (2, NOUT)
!
CALL CTEPR (TABLE, PRE, PRT=PRT)
!
WRITE (NOUT,99999) PRT, PRE
!
99999 FORMAT (' PRT = ', E12.4, ' PRE = ', F8.4)
!
END

```

Output

```
PRT = 0.1915E-04 PRE = 0.0601
```

For comparison, the usual asymptotic chi-squared p -value (which may be computed through the use of routine `CTCHI` ([page 491](#)), do not use `CTEPR`) is computed as 0.0323, and the Fisher exact probability (which may be computed through `CTEPR` by setting `EXPECT = 0.0`) is computed as 0.0598 and requires approximately ten times more computer time than the hybrid method. The Fisher exact probability and the usual asymptotic chi-squared probability will often be quite different. When it may be used, the hybrid algorithm can lead to significantly greater savings in computer time.

Comments

1. Workspace may be explicitly provided, if desired, by use of `C2EPR/DC2EPR`. The reference is:

```
CALL C2EPR (NROW, NCOL, TABLE, LDTABL, EXPECT, PERCNT, EMIN,
PRT, PRE, FACT, ICO, IRO, KYY, IDIF, IRN, KEY, LDKEY, IPOIN,
STP, LDSTP, IFRQ, DLP, DSP, TM, KEY2, IWK, RWK)
```

The additional arguments are as follows:

FACT — Work vector of length `NTOT + 1` where `NTOT` is the total count in the table.

ICO — Work vector of length `MX` where `MX = max(NROW, NCOL)`.

IRO — Work vector of length `MX`.

KYY — Work vector of length `MX`.

IDIF — Work vector of length MN where $MN = \max(NROW, NCOL)$.

IRN — Work vector of length MN .

KEY — Work vector of length $2 * LDKEY$.

LDKEY — Leading dimension of **KEY** exactly as specified in the dimension statement in the calling program. (Input)

IPOIN — Work vector of length $2 * LDKEY$.

STP — Work vector of length $2 * LDSTP$.

LDSTP — Leading dimension of **STP** exactly as specified in the dimension statement in the calling program. (Input)

IFRQ — Work vector of length $6 * LDSTP$.

DLP — Work vector of length $2 * LDKEY$.

DSP — Work vector of length $2 * LDKEY$.

TM — Work vector of length $2 * LDKEY$.

KEY2 — Work vector of length $2 * LDKEY$.

IWK — Work vector of length $\max((NROW + NCOL + 1)(5 + 2 * MX), 800 + 7 * MX)$.

RWK — Work vector of length $\max(400 + MX + 1, NROW + NCOL + 1)$.

The exact value of **LDKEY** and **LDSTP** required is not known in advance. Common values to try are **LDKEY** = 1000 and **LDSTP** = 30000.

2. Informational errors

Type Code

- | | | |
|---|---|--|
| 3 | 1 | All of the elements of TABLE are zero. |
| 4 | 2 | The product of the marginal totals is greater than can be exactly represented in an integer variable so the hash table key cannot be computed. The computations cannot proceed. |
| 4 | 3 | LDKEY is too small. To increase LDKEY when invoking CTEPR/DCTEPR , increase the total workspace used. A doubling of the total workspace is a good place to begin. |
| 4 | 4 | LDSTP is too small. To increase LDSTP when invoking CTEPR/DCTEPR , increase the total workspace used. A doubling of the total workspace is a good place to begin. |

- 4 5 The current value for `IWKIN` is too small. It is not possible to give the value for `IWKIN` required, but you might try doubling the amount. Refer to `IWKIN` in the Reference Material section.
3. Routine `CTEPR/DCTEPR` will use all available workspace. It is not unusual for `CTEPR/DCTEPR` to require 200,000 floating-point units of workspace.
4. When `C2EPR/DC2EPR` is called by `CTEPR/DCTEPR`, `LDSTP = 30 * LDKEY`.
5. Although not a restriction, it is not generally practical to call this routine with large tables that are not sparse and in which the hybrid approximation to Fisher's exact test (see the "Description" section) has little effect. For example, although it is feasible to compute exact probabilities for the table

1	8	5	4	4	2	2
5	3	3	4	3	1	0
10	1	4	0	0	0	0

computing exact probabilities for a similar table that has been enlarged by the addition of an extra row (or column) may not be feasible.

Description

Routine `CTEPR` computes Fisher exact probabilities or a hybrid algorithm approximation to Fisher exact probabilities for a $r \times c$ contingency tables with fixed row and column marginals where $r = \text{NROW}$ is the number of rows in the table and $c = \text{NCOL}$ is the number of columns in the table. Let f_{ij} denote the frequency count in row i and column j of a table, and let $f_{i\bullet}$ and $f_{\bullet j}$ denote the total row and column frequency count for row i and column j , respectively. Under the independence hypothesis, the (conditional) probability of the observed table for fixed row and column marginal totals is given by

$$P_f = \frac{\prod_{i=1}^r f_{i\bullet}! \prod_{j=1}^c f_{\bullet j}!}{f_{\bullet\bullet}! \prod_{i=1}^r \prod_{j=1}^c f_{ij}!}$$

where $f_{\bullet\bullet}$ is the total number of counts in the table and $x!$ denotes x factorial. When the f_{ij} are equal to the input table so that $f_{ij} = \text{TABLE}(i, j)$, then let $P_o = \text{PRT}$ be the resulting value for P_f .

In `CTEPR`, a more extreme table is defined in the probabilistic sense. Table X is more extreme than the input table if the conditional probability computed for table X (for the same marginal sums) is less than the conditional probability computed for the input table. Let $p = \text{PRE}$ be the probability of a more extreme table. Then

$$p = \sum_{P \leq P_o} P_f$$

The user should note that this definition of "more extreme" can be considered as "two-sided" in the cell counts.

Routine `CTEPR` uses the hybrid network algorithm of Mehta and Patel (1983, 1986a, 1986b) with the Clarkson and Fan (1989) modifications to compute the probability of a more extreme table. The hybrid algorithm uses asymptotic probabilities for tables encountered in which `PERCNT` percent of the table expected values are greater than or equal to `EXPECT`, and all expected values are greater than `EMIN`. When `PERCNT = 80`, `EXPECT = 5`, and `EMIN = 1`, this is the “Cochran” rule. Although the hybrid network algorithm can be orders of magnitude faster than the total enumeration algorithm used in routine `CTPRB` (page 501), the amount of computer time required by `CTEPR` still increases very rapidly with the size of the table. Caution should be used whenever computer time is a consideration.

PRPFT

Performs iterative proportional fitting of a contingency table using a loglinear model.

Required Arguments

- NCLVAL*** — Vector of length `NCLVAR` containing, in its i -th element, the number of levels or categories of the i -th classification variable. (Input)
- TABLE*** — Vector of length `NCLVAL(1) * NCLVAL(2) * ... * NCLVAL(NCLVAR)` containing the entries in the cells of the table to be fit. (Input)
See Comment 3 for comments on the ordering of the elements of `TABLE`.
- NVEF*** — Vector of length `NEF` that contains the number of classification variables associated with each effect. (Input)
- INDEF*** — Vector of length `NVEF(1) + ... + NVEF(NEF)` that contains, in consecutive positions, the indices of the variables that are included in each effect. (Input)
The entries in `INDEF` are sequenced so that the first `NVEF(1)` elements contain the indices of the variables in effect 1, the next `NVEF(2)` elements of `INDEF` contain the indices of the variables in effect 2, etc. See Comment 4 for an example.
- FIT*** — Vector of length `NCLVAL(1) * NCLVAL(2) * ... * NCLVAL(NCLVAR)`. (Input/Output)
On input, `FIT` contains the initial estimates of the cell counts. Structural zeros in the model are specified by setting the corresponding element of `FIT` to 0.0. All other elements of `FIT` must be positive. 1.0 may be used if no other estimate of the cell counts is available. See Comment 3 for the ordering of the elements of `FIT`. On output, `FIT` contains the fitted table.

Optional Arguments

- NCLVAR*** — Number of classification variables. (Input)
Default: `NCLVAR = size(NCLVAL,1)`.
- NEF*** — Number of effects in the model. (Input)
A marginal table is implied by each effect in the model. Lower order effects should not be included since their inclusion is automatic (e.g., do not include effects A or B if

effect AB is in the model).
 Default: $NEF = \text{size}(\text{NVEF}, 1)$.

EPS — Convergence criterion. (Input)
 Convergence is assumed when the maximum deviation between an observed and a fitted marginal total is less than EPS . $EPS = 0.10$ is a typical value.
 Default: $EPS = .10$.

MAXIT — Maximum number of iterations. (Input)
 $MAXIT = 15$ is a typical value.
 Default: $MAXIT = 30$.

FORTRAN 90 Interface

Generic: `CALL PRPFT (NCLVAL, TABLE, NVEF, INDEF, FIT [, ...])`

Specific: The specific interface names are `S_PRPFT` and `D_PRPFT`.

FORTRAN 77 Interface

Single: `CALL PRPFT (NCLVAR, NCLVAL, TABLE, NEF, NVEF, INDEF, EPS, MAXIT, FIT)`

Double: The double precision name is `DPRPFT`.

Example

The following example is taken from Bishop, Feinberg, and Holland (1975, page 87). The data are originally from Bartlett (1935). This example examines the survival of plants (factor $A =$ factor 2) at different values for time of planting (factor $C =$ factor 3) and length of cutting (factor $B =$ factor 1). The sample size for each level of B and C is fixed at 240.

		B					
		1				2	
		A				A	
		1	2			1	2
C	1	156	84	C	1	84	156
	2	107	133		2	31	209

The model to be fit is given by:

$$\ln(m_{ijk}) = \mu + \alpha_i + \beta_j + \alpha\beta_{ij} + \gamma_k + \alpha\gamma_{ik} + \beta\gamma_{jk}$$

where m_{ijk} is the cell expected value for levels $i, j,$ and k of factors $A, B,$ and $C,$ respectively.

```
USE PRPFT_INT
USE UMACH_INT
```

```

INTEGER    NCLVAR, NEF
PARAMETER  (NCLVAR=3, NEF=3)
!
INTEGER    INDEF(6), MAXIT, NCLVAL(NCLVAR), NOUT, NVEF(NEF)
REAL       EPS, FIT(8), TABLE(8)
!
DATA NCLVAL/2, 2, 2/, NVEF/2, 2, 2/
DATA INDEF/1, 2, 1, 3, 2, 3/, EPS/0.0001/, MAXIT/15/
DATA TABLE/156, 107, 84, 31, 84, 133, 156, 209/
DATA FIT/8*1.0/
!
CALL PRPFT (NCLVAL, TABLE, NVEF, INDEF, FIT, EPS=EPS, MAXIT=MAXIT)
!
CALL UMACH (2, NOUT)
WRITE (NOUT,99999) FIT
99999 FORMAT (' FIT =', 8F7.1)
END

```

Output

```
FIT = 161.1 101.9 78.9 36.1 78.9 138.1 161.1 203.9
```

Comments

1. Workspace may be explicitly provided, if desired, by use of P2PFT/DP2PFT. The reference is:

```
CALL P2PFT (NCLVAR, NCLVAL, TABLE, NEF, NVEF, INDEF, EPS, MAXIT,
FIT, AMAR, INDEX, WK, IWK)
```

The additional arguments are as follows.

AMAR — Work vector with length equal to the sum from $J = 1$ to NEF of the product of the nonzero elements of $NCLVAL(INDEF(I))$ for $I = 1$ to $NVEF(J)$.

INDEX — Work vector of length NEF .

WK — Work vector with length equal to the maximum over $J = 1$ to NEF of the product of the nonzero elements of $NCLVAL(INDEF(I))$, for $I = 1$ to $NVEF(J)$.

IWK — Work vector of length $2 * NCLVAR$.

2. Informational errors

Type Code

- | | | |
|---|----|--|
| 3 | 11 | The algorithm did not converge to the desired accuracy within <code>MAXIT</code> iterations. |
| 4 | 12 | A marginal total for an effect is zero. Since <code>FIT</code> indicates this is not a structural zero, the algorithm will not converge properly. One way to proceed is to add a constant to all cells in the table. |

3. The cells of the vectors `TABLE` and `FIT` are sequenced so that the first variable cycles from 1 to $NCLVAL(1)$, which is the slowest, the second variable cycles from 1 to

NCLVAL(2), which is the next slowest, etc., up to the NCLVAR-th variable, which cycles from 1 to NCLVAL(NCLVAR) the fastest.

Example. For NCLVAR = 3, NCLVAL(1) = 2, NCLVAL(2) = 3, and NCLVAL(3) = 2, the cells of table X(I, J, K) are entered into TABLE(1) through TABLE(12) in the following order.

X(1, 1, 1), X(1, 1, 2), X(1, 2, 1), X(1, 2, 2), X(1, 3, 1), X(1, 3, 2), X(2, 1, 1), X(2, 1, 2), X(2, 2, 1), X(2, 2, 2), X(2, 3, 1), X(2, 3, 2). The elements of FIT are similarly sequenced.

- 4 INDEF is used to describe the marginal tables to be fit. For example, if NCLVAR = 3 and the first effect is to fit the marginal table for variables 1 and 3 and the second effect is to fit the marginal table for variable 2, then: NEF = 2, NVEF(1) = 2, and NVEF(2) = 1.

Since the sum of the NVEF(I) is 3, then INDEF is a vector of length 3 with values. INDEF(1) = 1, INDEF(2) = 3, and INDEF(3) = 2.

5. Typically, MAXIT = 5 is sufficient. If PRPFT does not converge, try using double precision, increasing MAXIT, or using the values output in FIT as input for another call to PRPFT.

Description

Routine PRPFT uses the iterative proportional-fitting algorithm to fit a log-linear hierarchical model to a contingency table. Structural zeros are allowed. A hierarchical model is a factorial model in which lower-order terms are always present. Thus, in a three-way table with classification variable names *A*, *B*, and *C*, the following models are all hierarchical models.

$$\begin{array}{cccc}
 A & B & C & AB \\
 A & B & C & AB & BC \\
 A & C & AC \\
 A & B & C & AB & AC & BC
 \end{array}$$

Many other hierarchical models exist for the three-way table. Since all hierarchical models can be completely specified by the higher-order interactions (the lower-order interactions will always be present), no lower-order effects are included in model specification.

Corresponding to each hierarchical interaction is a marginal table. Iterations in PRPFT proceed by fitting marginal tables successively until the desired precision is achieved.

A structural zero is a cell in the table that, by design or otherwise, can have no observations, i.e., the count for the cell must be zero. Structural zeros are specified by setting the corresponding element in FIT to zero on input. Routine PRPFT is best suited for tables with no structural zeros and in which the initial estimates input in FIT are all 1. The user should be aware that the algorithm may take (much) longer to converge when this is not the case.

Sampling zeros are cells that are not structural zeros, but for which no count is observed. Routine PRPFT requires the absence of sampling zeros in all marginal tables that are fit. One common way method of achieving this is to add a constant, often 0.5, to each cell prior to fitting the table.

CTLLN

Computes model estimates and associated statistics for a hierarchical log-linear model.

Required Arguments

NCLVAL — Vector of length `NCLVAR` containing, in its i -th element, the number of levels or categories of the i -th classification variable. (Input)

TABLE — Vector of length `NCLVAL(1) * NCLVAL(2) * ... * NCLVAL(NCLVAR)` containing the entries in the cells of the table to be fit. (Input)
See Comment 3 for comments on the ordering of the elements of `TABLE`.

NVEF — Vector of length `NEF` containing the number of classification variables associated with each effect. (Input)

INDEF — Vector of length `NVEF(1) + ... + NVEF(NEF)` containing, in consecutive positions, the indices of the variables that are included in each effect. (Input)
The entries in `INDEF` are sequenced so that the first `NVEF(1)` elements contain the indices of the variables in effect 1, the next `NVEF(2)` elements of `INDEF` contain the indices of the variables in effect 2, etc. See Comment 4 for an example.

FIT — Vector of length `NCLVAL(1) * NCLVAL(2) * ... * NCLVAL(NCLVAR)` containing the model estimates of the cell frequencies. (Input/Output)
On input, `FIT` contains the initial estimates of the cell counts. Structural zeros in the model are specified by setting the corresponding element of `FIT` to 0.0. All other elements of `FIT` may be set to 1.0 if no other estimate of the expected cell counts is available. On output, `FIT` contains the fitted table. See Comment 3 for the ordering of the elements of `FIT`. If an element of `FIT` is positive but the corresponding element in `TABLE` is zero, then the element is called a sampling zero. Sampling zeros may effect the number of parameters that can be estimated, but they will not effect the degrees of freedom in chi-squared tests. See the “Algorithm” section.

NCOEF — Number of regression coefficients in the model. (Output)

COEF — `NCOEF` by 4 matrix containing the estimated coefficients and associated statistics. (Output)
Dummy variables used in fitting the log-linear model are generated using the `IDUMMY = 3` option of routine `GRGLM` (see Chapter 2, Regression). For this option, the k -th dummy variable for classification variable I is the $(0, 1)$ indicator variable for the k -th level of the classification variable minus the $(0, 1)$ indicator variable for the `NCLVAL(I)`-th level of the classification variable.

Column Statistic

- 1 Coefficient estimate
- 2 Estimated standard error of the estimated coefficient
- 3 Asymptotic normal score for testing that the coefficient is zero
- 4 p -value associated with the normal score in column 3 (two-sided alternative).

COV — NCOEF by NCOEF covariance matrix for the estimated parameters. (Output)

RESID — NCLVAL(1) * NCLVAL(2) * ... * NCLVAL(NCLVAR) by 4 matrix containing residual statistics for each cell in the table. (Output)

Column Statistic

- 1 Signed square root of the contribution to chi-squared
- 2 Contribution to the likelihood ratio
- 3 Freeman-Tukey deviate
- 4 Residual difference

STAT — Vector of length 4 containing output statistics for the model. (Output)

I STAT(I)

- 1 Log-likelihood.
- 2 Likelihood ratio statistic for testing the fit of the model.
- 3 Degrees of freedom in the likelihood ratio statistic. This statistic corrects for parameters that cannot be estimated because of sampling zeros.
- 4 p -value corresponding to the likelihood ratio statistic.

Optional Arguments

NCLVAR — Number of classification variables. (Input)

A variable specifying a margin in the table is a classification variable. The first classification variable is named *A*, the second classification variable is named *B*, etc. Default: NCLVAR = size (NCLVAL,1).

NEF — Number of effects in the model. (Input)

A marginal table is implied by each effect in the model. Lower-order effects should not

be included since their inclusion is automatic in the hierarchical models fit here (e.g., do not include effects *A* or *B* if effect *AB* is in the model).

Default: `NEF = size (NVEF,1)`.

EPS — Convergence criterion. (Input)

Convergence is assumed when the maximum deviation between an observed and a fitted marginal total is less than `EPS`. `EPS = 0.10` is a typical value.

Default: `EPS = 0.10`.

MAXIT — Maximum number of iterations. (Input)

`MAXIT = 15` is a typical value.

Default: `MAXIT = 30`

TOL — Tolerance used in determining linear dependence in `COV`. (Input)

`TOL = 100.0 AMACH(4)` is a common choice. See the documentation for routine `AMACH` (Reference Material).

Default: `TOL = 1.19e-5` for single precision and `2.d-14` for double precision.

IPRINT — Printing option. (Input)

Default: `IPRINT = 0`.

IPRINT	Action
---------------	---------------

0	No printing is performed.
---	---------------------------

1	<code>TABLE</code> , <code>FIT</code> , <code>RESID</code> , <code>COEF</code> , <code>COV</code> , and <code>STAT</code> are printed.
---	--

LDCOEF — Leading dimension of `COEF` exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDCOEF = size (COEF,1)`.

LDCOV — Leading dimension of `COV` exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDCOV = size (COV,1)`.

LDRESI — Leading dimension of `RESID` exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDRESI = size (RESID,1)`.

FORTRAN 90 Interface

Generic: `CALL CTTLN (NCLVAL, TABLE, NVEF, INDEF, FIT, NCOEF, COEF, COV, RESID, STAT[,...])`

Specific: The specific interface names are `S_CTTLN` and `D_CTTLN`.

FORTRAN 77 Interface

Single: CALL CTLN (NCLVAR, NCLVAL, TABLE, NEF, NVEF, INDEF, EPS,
 MAXIT, TOL, IPRINT, FIT, NCOEF, COEF, LDCOEF, COV, LDCOV,
 RESID, LDRESI, STAT)

Double: The double precision name is DCTLN.

Example

The example illustrates the use of CTLN in a simple four-way table in which the first three factors have two levels, and the fourth factor has three levels. The data, taken from Lee (1977), involve brand preference in different situations.

```
USE CTLN_INT
INTEGER  IPRINT, LDCOEF, LDCOV, LDRESI, LTAB, MAXIT, NCLVAR
REAL    EPS
PARAMETER (EPS=0.01, IPRINT=1, LDCOEF=10, LDCOV=10, LDRESI=24, &
           LTAB=24, MAXIT=10, NCLVAR=4)
!
INTEGER  INDEF(6), NCLVAL(NCLVAR), NCOEF, NEF, NVEF(3)
REAL    COEF(LDCOEF,4), COV(LDCOV,LDCOV), FIT(LTAB), &
        RESID(LDRESI,4), STAT(4), TABLE(LTAB)
!
DATA TABLE/19, 57, 29, 63, 29, 49, 27, 53, 23, 47, 33, 66, 47, &
        55, 23, 50, 24, 37, 42, 68, 43, 52, 30, 42/
DATA NEF/3/, NVEF/2, 2, 2/, INDEF/2, 4, 1, 4, 2, 3/
DATA NCLVAL/3, 2, 2, 2/, FIT/24*1.0/
!
CALL CTLN (NCLVAL, TABLE, NVEF, INDEF, FIT, NCOEF, COEF, &
          COV, RESID, STAT, EPS=EPS, MAXIT=MAXIT, IPRINT=IPRINT)
!
END
```

Output

Fitted Model: (B*D, A*D, B*C)

Variable Number of Levels

1 A	3
2 B	2
3 C	2
4 D	2

Model Statistics

Log-likelihood	3.7906
Likelihood ratio	11.89
Degrees of freedom	14.0
P-value	0.6154

Coefficient Statistics

	Coefficient	Standard Error	Asymptotic Z-statistic	P-value
1 intercept	3.6827	0.0333	110.66	0.0000

2	A(1)	-0.0591	0.0475	-1.24	0.2341
3	A(2)	0.0278	0.0461	0.60	0.5562
4	B	-0.0166	0.0331	-0.50	0.6242
5	C	-0.0434	0.0319	-1.36	0.1943
6	D	-0.2783	0.0329	-8.45	0.0000
7	A*D(1)	-0.1016	0.0475	-2.14	0.0506
8	A*D(2)	0.0034	0.0461	0.07	0.9414
9	B*C	-0.1438	0.0319	-4.51	0.0005
10	B*D	-0.0684	0.0328	-2.09	0.0558

Table 1: C = 1 D = 1
B = 1 by A (column)

		1	2	3
Observed		19.00	23.00	24.00
Fit		19.52	23.65	26.09
Root chi-square		-0.12	-0.13	-0.41
Likelihood		-1.03	-1.29	-4.02
Freeman-Tukey		-0.06	-0.08	-0.37
Residual		-0.52	-0.65	-2.09

B = 2 by A (column)

		1	2	3
Observed		29.00	47.00	43.00
Fit		30.85	37.37	41.23
Root chi-square		-0.33	1.57	0.28
Likelihood		-3.58	21.54	3.62
Freeman-Tukey		-0.29	1.52	0.31
Residual		-1.85	9.63	1.77

Table 2: C = 1 D = 2
B = 1 by A (column)

		1	2	3
Observed		57.00	47.00	37.00
Fit		47.85	46.99	42.89
Root chi-square		1.32	0.00	-0.90
Likelihood		19.95	0.03	-10.93
Freeman-Tukey		1.29	0.04	-0.89
Residual		9.15	0.01	-5.89

B = 2 by A (column)

		1	2	3
Observed		49.00	55.00	52.00
Fit		57.52	56.48	51.56
Root chi-square		-1.12	-0.20	0.06
Likelihood		-15.70	-2.92	0.89
Freeman-Tukey		-1.13	-0.16	0.10
Residual		-8.52	-1.48	0.44

Table 3: C = 2 D = 1
B = 1 by A (column)

		1	2	3
--	--	---	---	---

Observed	29.00	33.00	42.00
Fit	28.39	34.40	37.94
Root chi-square	0.11	-0.24	0.66
Likelihood	1.23	-2.73	8.53
Freeman-Tukey	0.16	-0.20	0.68
Residual	0.61	-1.40	4.06

B = 2 by A (column)

	1	2	3
Observed	27.00	23.00	30.00
Fit	25.24	30.58	33.73
Root chi-square	0.35	-1.37	-0.64
Likelihood	3.64	-13.10	-7.04
Freeman-Tukey	0.39	-1.41	-0.61
Residual	1.76	-7.58	-3.73

Table 4: C = 2 D = 2

B = 1 by A (column)

	1	2	3
Observed	63.00	66.00	68.00
Fit	69.58	68.32	62.37
Root chi-square	-0.79	-0.28	0.71
Likelihood	-12.51	-4.57	11.75
Freeman-Tukey	-0.78	-0.25	0.73
Residual	-6.58	-2.32	5.63

B = 2 by A (column)

	1	2	3
Observed	53.00	50.00	42.00
Fit	47.06	46.21	42.18
Root chi-square	0.87	0.56	-0.03
Likelihood	12.61	7.88	-0.36
Freeman-Tukey	0.87	0.58	0.01
Residual	5.94	3.79	-0.18

Asymptotic Coefficient Covariance

	1	2	3	4	5
1	1.1076E-03	9.7132E-05	-3.5887E-05	4.3244E-05	4.3786E-05
2		2.2562E-03	-1.1408E-03	-3.4043E-11	2.6829E-11
3			2.1232E-03	2.5675E-11	-5.1643E-11
4				1.0968E-03	1.4480E-04
5					1.0146E-03
	6	7	8	9	10
1	2.9815E-04	1.3065E-04	-1.6147E-05	1.4480E-04	7.6307E-05
2	1.3065E-04	7.2117E-04	-4.0976E-04	6.2343E-11	-1.0681E-11
3	-1.6147E-05	-4.0976E-04	5.7437E-04	-4.9217E-11	-2.3482E-11
4	7.6307E-05	1.2601E-11	-4.1730E-11	4.3786E-05	2.8917E-04
5	-1.4272E-11	-5.5301E-11	4.2801E-11	4.5231E-06	-4.6962E-11
6	1.0851E-03	9.7132E-05	-3.5887E-05	-4.9749E-11	3.0847E-05
7		2.2562E-03	-1.1408E-03	5.9300E-11	-1.0361E-10
8			2.1232E-03	-2.4481E-11	2.9160E-11
9				1.0146E-03	1.1201E-11
10					1.0743E-03

Comments

1. Workspace may be explicitly provided, if desired, by use of C2LLN/DC2LLN. The reference is:

```
CALL C2LLN (NCLVAR, NCLVAL, TABLE, NEF, NVEF, INDEF, EPS, MAXIT,  
TOL, IPRINT, FIT, NCOEF, COEF, LDCOEF, COV, LDDEV, RESID,  
LDRESI, STAT, AMAR, INDEX, NCVF, IXEF, IINDEF, IA, INDCL,  
CLVAL, REG, X, D, XMIN, XMAX, COVWK, WK, IWK)
```

The additional arguments are as follows.

AMAR — Vector of length equal to the sum over all effects in the model ($J = 1$ to NEF) of the length of the marginal table required for the effect. The length of each marginal table is computed as the product of the number of class values for each classification variable in the effect (the product of the nonzero elements of $NCLVAL(INDEF(I))$ where I ranges from $K(J)$ through $K(J) + NVEF(J) - 1$. Here, $K(1) = 1$ and $K(J + 1) = K(J) + NVEF(J)$.)

INDEX — Vector of length NEF .

NCVCF — Vector of length $2^{NCLVAR} - 1$.

IXEF — Vector of length $NCLVAR * 2^{NCLVAR-1}$.

IINDEF — Vector of length $NVEF(1) + \dots + NVEF(NEF)$.

IA — Vector of length $NCLVAR$.

INDCL — Vector of length $NCLVAR$.

CLVAL — Vector of length $NCLVAL(1) + \dots + NCLVAL(NCLVAR)$.

REG — Vector of length $NCOEF + 1$.

X — Vector of length $NCOEF$ if there exists both structural and sampling zeros in **TABLE**; otherwise, it is of length $NCLVAR$.

D — Vector of length $NCOEF + 1$.

XMIN — Vector of length $NCOEF$.

XMAX — Vector of length $NCOEF$.

COVWK — Vector of length $NCOEF^2$ if there exists both structural and sampling zeros in **TABLE**. Otherwise, **COVWK** is not referenced and can be dimensioned of length one.

WK — Vector of length $\max(g, \text{NCOEF} + 1)$ if `IPRINT = 0`; otherwise, `WK` is of length $\max(g, 6m, n)$ where m is the maximal element in `NCLVAL`, n is the length of `TABLE`, and g equals the maximum over all effects in the model ($J = 1, \text{NEF}$) of the length of the marginal table required for the effect. The length of the marginal table is computed as the product of the number of class values for each classification variable in the effect (the product of the nonzero elements of `NCLVAL(INDEF(I))` where I ranges from $K(J)$ through $K(J) + \text{NVEF}(J) - 1$, where $K(1) = 1$ and $K(J + 1) = K(J) + \text{NVEF}(J)$).

IWK — Vector of length $2 * \text{NCLVAR} + z + 1$ where z is the number of structural zeros in `TABLE`.

2. Informational errors

Type Code

- | | | |
|---|----|---|
| 3 | 1 | The optimization algorithm did not converge to the desired accuracy within <code>MAXIT</code> iterations. Some of the estimated statistics may not be accurate. |
| 3 | 5 | The label for one or more of the tables exceeds the buffer limit. |
| 3 | 11 | The label for one or more effects exceeds the buffer limit. |
| 4 | 2 | <code>LDCOEF</code> or <code>LDCOV</code> is less than <code>NCOEF</code> . |

3. The cells of the vectors `TABLE` and `ZERO` are sequenced so that the first variable cycles from 1 to `NCLVAL(1)` the slowest, the second variable cycles from 1 to `NCLVAL(2)` the next slowest, etc., up to the `NCLVAR`-th variable, which cycles from 1 to `NCLVAL(NCLVAR)` the fastest.

Example: For `NCLVAR = 3`, `NCLVAL(1) = 2`, `NCLVAL(2) = 3`, and `NCLVAL(3) = 2`, the cells of table `X(I, J, K)` are entered into `TABLE(1)` through `TABLE(12)` in the following order:

`X(1, 1, 1)`, `X(1, 1, 2)`, `X(1, 2, 1)`, `X(1, 2, 2)`, `X(1, 3, 1)`, `X(1, 3, 2)`, `X(2, 1, 1)`, `X(2, 1, 2)`, `X(2, 2, 1)`, `X(2, 2, 2)`, `X(2, 3, 1)`, `X(2, 3, 2)`. The elements of `FIT` are similarly sequenced.

4. `INDEF` is used to describe the marginal tables to be fit. For example, if `NCLVAR = 3` and the first effect is to fit the marginal table for variables 1 and 3 and the second effect is to fit the marginal table for variable 2, then: `NEF = 2`, `NVEF(1) = 2`, and `NVEF(2) = 1`. Since the sum of the `NVEF(I)` is 3, then `INDEF` is a vector of length 3 with values: `INDEF(1) = 1`, `INDEF(2) = 3`, and `INDEF(3) = 2`.

Description

Routine `CTLLN` computes statistics of interest for a hierarchical model in a log-linear analysis of a multidimensional contingency table. Among the statistics computed are the expected cell values, cell residuals, the log-linear parameters and their estimated variances and covariances, the log-likelihood for the model (plus a constant), and a likelihood-ratio test of the model (versus the alternative that the cell probabilities are free to vary, subject only to the marginal constraints). In addition, `CTLLN` can print and label all statistics that it computes.

Routine `PRPFT` (page 509) is used to find the maximum likelihood estimates of the expected cell counts (`FIT`). These expected values are then used as input to routine `CTPAR` (page 522) in order to compute estimates of the parameters in the model and their estimated covariances.

The matrix `RESID` contains various residuals that may be used in analyzing the model. These residuals are discussed in detail by Bishop, Feinberg, and Holland (1975, pages 136-137), among others. Each is computed from the cell observed (o_i) and expected (fitted, f_i) values according to the following methods:

1. The signed square root of the contributions to χ^2 are computed as

$$(o_i - f_i) / \sqrt{f_i}$$

2. The contributions to the likelihood ratio (G^2) are computed as $2o_i \log(o_i/f_i)$
3. Freeman-Tukey deviates are computed as

$$\sqrt{o_i} + \sqrt{o_i + 1} - \sqrt{4f_i + 1}$$

4. The residual differences are computed as $o_i - f_i$

The log-likelihood `STAT(1)` is computed as

$$\sum_{i=1}^n -o_i \log(f_i)$$

where n is the number of cells in the table. The likelihood ratio statistic for testing the fit of the model is computed as

$$G^2 = \sum_{i=1}^n 2o_i \log\left(\frac{o_i}{f_i}\right)$$

which for large samples follows a chi-squared distribution.

The number of degrees of freedom in G^2 is computed as the number of cells in the table, excluding structural zeros, minus the number of parameters that could be estimated if there were no sampling zeros. When there are either structural or sampling zeros in the model, some parameters may not be estimable because they are infinite. Parameters that cannot be estimated due to structural zeros are not counted in the number of parameters estimated when computing the degrees of freedom for χ^2 . Parameters that cannot be estimated because of sampling zeros are counted as estimated parameters when computing the degrees of freedom for χ^2 .

To explain the calculation of degrees of freedom, note that extended maximum likelihood estimates may be written as

$$\hat{\beta} = \hat{\beta}_F + \rho \hat{\beta}_\infty$$

where

$$\hat{\beta}, \hat{\beta}_F \text{ and } \rho \hat{\beta}_\infty$$

are coefficient vectors, and $\rho \rightarrow \infty$. Routine `CTLLN` estimates the finite portion of the estimates, $\hat{\beta}_F$

The infinite portion,

$$\hat{\beta}_\infty$$

ensures that the fitted values for zero marginal cells corresponding to a term in the hierarchical model have estimated expectation of zero. Thus, `CTLLN` fits the finite portion of extended maximum likelihood estimates where the extension is to $\pm\infty$. Because the Hessian elements corresponding to infinite parameters are zero, the Hessian is computed from a reduced likelihood in which cells leading to infinite estimates have been eliminated. The user is referred to Clarkson and Jennrich (1991) for details.

CTPAR

Computes model estimates and covariances in a fitted log-linear model.

Required Arguments

NCLVAL — Vector of length `NCLVAR` containing, in its i -th element, the number of levels or categories of the i -th classification variable. (Input)

NVEF — Vector of length `NEF` containing the number of classification variables associated with each effect. (Input)

INDEF — Vector of length $NVEF(1) + \dots + NVEF(NEF)$ containing, in consecutive positions, the indices of the variables that are included in each effect. (Input)

The entries in `INDEF` are sequenced so that the first $NVEF(1)$ elements contain the indices of the variables in effect 1, the next $NVEF(2)$ elements of `INDEF` contain the indices of the variables in effect 2, etc. See Comment 4 for an example.

FIT — Vector of length $NCLVAL(1) * NCLVAL(2) * \dots * NCLVAL(NCLVAR)$ containing the model estimates of the cell counts. (Input)

See Comment 3 for the ordering of the elements of `FIT`. To obtain a first iteration approximation to the optimal parameter values, the observed counts may be input in `FIT`, in which case a least-squares model is fit. In all cases, values of zero in `FIT` are assumed to correspond to structural zeros in the table. See the “Algorithm” section for details.

NCOEF — Number of regression coefficients in the model. (Output)

COEF — $NCOEF$ by 4 matrix containing the estimated coefficients and associated statistics. (Output)

Col. Statistic

- 1 Coefficient estimate
- 2 Estimated standard error of the estimated coefficient
- 3 Asymptotic normal score for testing that the coefficient is zero
- 4 p -value associated with the normal score in column 3 (two-sided alternative)

COV — NCOEF by NCOEF covariance matrix of the estimated coefficients. (Output)

Optional Arguments

NCLVAR — Number of classification variables. (Input)

A variable specifying a margin in the table is a classification variable. The first classification variable is named A , the second classification variable is named B , etc.

Default: NCLVAR = size (NCLVAL,1).

NEF — Number of effects in the model. (Input)

A marginal table is implied by each effect in the model. Lower-order effects should not be included since their inclusion is automatic in the hierarchical models fit here (e.g., do not include effects A or B if effect AB is in the model).

Default: NEF = size (NVEF,1).

TOL — Tolerance used in determining linear dependence in COV. (Input)

TOL = 100.0 * AMACH(4) is a common choice. See the documentation for routine AMACH (Reference Material).

Default: TOL = 1.19e-5 for single precision and 2.d -14 for double precision.

IPRINT — Printing option. (Input)

Default: IPRINT = 0.

IPRINT Action

- 0 No printing is performed.
- 1 Printing of COEF and COV is performed.
- 2 COEF, COV, and FIT are printed.

In the printing, $A * B(2)$ denotes the second variable in the AB interaction effect.

LDCOEF — Leading dimension of COEF exactly as specified in the dimension statement in the calling program. (Input)

Default: LDCOEF = size (COEF,1).

LDCOV— Leading dimension of COV exactly as specified in the dimension statement in the calling program. (Input)

Default: LDCOV = size (COV,1).

FORTRAN 90 Interface

Generic: CALL CTPAR (NCLVAL, NVEF, INDEF, FIT, NCOEF, COEF, COV [,...])

Specific: The specific interface names are S_CTPAR and D_CTPAR.

FORTRAN 77 Interface

Single: CALL CTPAR (NCLVAR, NCLVAL, NEF, NVEF, INDEF, FIT, TOL, IPRINT, NCOEF, COEF, LDcoef, COV, LDCOV)

Double: The double precision name is DCTPAR.

Example

The example illustrates the use of CTPAR in a simple four-way table in which the first three factors have two levels, and the fourth factor has three levels. The data, which is taken from Lee (1977), involve the brand preference in different situations.

```
USE PRPFT_INT
USE CTPAR_INT

INTEGER IPRINT, LDcoef, LDCOV, LTAB, NCLVAR
PARAMETER (IPRINT=2, LDcoef=13, LDCOV=13, LTAB=24, NCLVAR=4)
!
INTEGER INDEF(6), NCLVAL(NCLVAR), NCOEF, NVEF(3)
REAL COEF(LDcoef,4), COV(LDCOV,LDCOV), FIT(LTAB), &
TABLE(LTAB)
!
DATA TABLE/19, 57, 29, 63, 29, 49, 27, 53, 23, 47, 33, 66, 47, &
55, 23, 50, 24, 37, 42, 68, 43, 52, 30, 42/
DATA NVEF/2, 2, 2/, INDEF/2, 4, 1, 4, 2, 3/
DATA NCLVAL/3, 2, 2, 2/, FIT/24*1.0/
!
CALL PRPFT (NCLVAL, TABLE, NVEF, INDEF, FIT)
!
CALL CTPAR (NCLVAL, NVEF, INDEF, FIT, NCOEF, COEF, COV, IPRINT=IPRINT)
!
END
```

Output

Variable	Number of Levels
1 A	3
2 B	2
3 C	2
4 D	2

Table 1: B = 1 C = 1
D (row) by A (column)

	1	2	3
1	19.52	23.65	26.09
2	47.85	46.99	42.89

Table 2: B = 1 C = 2
D (row) by A (column)

	1	2	3
1	28.39	34.40	37.94
2	69.58	68.32	62.37

Table 3: B = 2 C = 1
D (row) by A (column)

	1	2	3
1	30.85	37.37	41.23
2	57.52	56.48	51.56

Table 4: B = 2 C = 2
D (row) by A (column)

	1	2	3
1	25.24	30.58	33.73
2	47.06	46.21	42.18

Coefficient Statistics

	Coefficient	Standard Error	Asymptotic Z-statistic	P-value
1 intercept	3.6827	0.0333	110.66	0.0000
2 A(1)	-0.0591	0.0475	-1.24	0.2341
3 A(2)	0.0278	0.0461	0.60	0.5562
4 B	-0.0166	0.0331	-0.50	0.6242
5 C	-0.0434	0.0319	-1.36	0.1943
6 D	-0.2783	0.0329	-8.45	0.0000
7 A*D(1)	-0.1016	0.0475	-2.14	0.0506
8 A*D(2)	0.0034	0.0461	0.07	0.9414
9 B*C	-0.1438	0.0319	-4.51	0.0005
10 B*D	-0.0684	0.0328	-2.09	0.0558

Asymptotic Coefficient Covariance

	1	2	3	4	5
1	1.1076E-03	9.7132E-05	-3.5887E-05	4.3244E-05	4.3786E-05
2		2.2562E-03	-1.1408E-03	-3.4043E-11	2.6829E-11
3			2.1232E-03	2.5675E-11	-5.1643E-11
4				1.0968E-03	1.4480E-04
5					1.0146E-03
	6	7	8	9	10
1	2.9815E-04	1.3065E-04	-1.6147E-05	1.4480E-04	7.6307E-05
2	1.3065E-04	7.2117E-04	-4.0976E-04	6.2343E-11	-1.0681E-11
3	-1.6147E-05	-4.0976E-04	5.7437E-04	-4.9217E-11	-2.3482E-11
4	7.6307E-05	1.2601E-11	-4.1730E-11	4.3786E-05	2.8917E-04
5	-1.4272E-11	-5.5301E-11	4.2801E-11	4.5231E-06	-4.6962E-11
6	1.0851E-03	9.7132E-05	-3.5887E-05	-4.9749E-11	3.0847E-05

7	2.2562E-03	-1.1408E-03	5.9300E-11	-1.0361E-10
8		2.1232E-03	-2.4481E-11	2.9160E-11
9			1.0146E-03	1.1201E-11
10				1.0743E-03

Comments

1. Workspace may be explicitly provided, if desired, by use of `C2PAR/DC2PAR`. The reference is:

```
CALL C2PAR (NCLVAR, NCLVAL, NEF, NVEF, INDEF, FIT, TOL, IPRINT,
NCOEF, COEF, LDcoef, COV, LDcov, IRANK, NCVef, IXEF, IINDEF, IA,
INDCL, CLVAL, REG, X, D, XMIN, XMAX, WK)
```

The additional arguments are as follows:

IRANK — Rank of COV.

NCVEF — Vector of length $2^{NCLVAR} - 1$.

IXEF — Vector of length $NCLVAR * 2^{NCLVAR-1}$.

IINDEF — Vector of length $NVEF(1) + \dots + NVEF(NEF)$.

IA — Vector of length $NCLVAR$.

INDCL — Vector of length $NCLVAR$.

CLVAL — Vector of length $NCLVAL(1) + \dots + NCLVAL(NCLVAR)$.

REG — Vector of length $NCOEF + 1$.

X — Vector of length $NCLVAR$.

D — Vector of length $NCOEF$.

XMIN — Vector of length $NCOEF$.

XMAX — Vector of length $NCOEF$.

WK — Vector of length $NCOEF + 1$ if `IPRINT` $\neq 2$. Otherwise, its length is the maximum of $NCOEF + 1$ and the product of the two largest elements of `NCLVAL`.

2. Informational errors

Type	Code	
3	5	The label for one or more of the tables exceeds the buffer limit.
3	11	The label for one or more effects exceeds the buffer limit.

4 1 LDCOEF or LDCOV is less than NCOEF.

3. The cells of the vector FIT are sequenced so that the first variable cycles from 1 to NCLVAL(1) the slowest, the second variable cycles from 1 to NCLVAL(2) the next slowest, etc., up to the NCLVAR-th variable, which cycles from 1 to NCLVAL(NCLVAR) the fastest.
Example: For NCLVAR = 3, NCLVAL(1) = 2, NCLVAL(2) = 3, and NCLVAL(3) = 2, the cells of table X(I, J, K) are entered into FIT(1) through FIT(12) in the following order: X(1, 1, 1), X(1, 1, 2), X(1, 2, 1), X(1, 2, 2), X(1, 3, 1), X(1, 3, 2), X(2, 1, 1), X(2, 1, 2), X(2, 2, 1), X(2, 2, 2), X(2, 3, 1), X(2, 3, 2).
4. INDEF is used to describe the marginal tables to be fit. For example, if NCLVAR = 3 and the first effect is to fit the marginal table for variables 1 and 3 and the second effect is to fit the marginal table for variable 2, then: NEF = 2, NVEF(1) = 2, and NVEF(2) = 1. Since the sum of the NVEF(I) is 3, then INDEF is a vector of length 3 with values: INDEF(1) = 1, INDEF(2) = 3, and INDEF(3) = 2.

Description

Routine CTPAR computes estimates of parameters and associated variances and covariances in hierarchical loglinear models. A weighted least-squares algorithm is used.

A hierarchical analysis of variance model is a factorial analysis of variance model in which a lower-order effect is included in a model whenever a higher-order effect containing it is in the model. Thus, if the effect *ADF* is in the model, then effects *A*, *D*, *F*, *AD*, *AF*, and *DF* are automatically in the model.

Input to CTPAR may be either the expected table values for the given hierarchical model as output, for example, by routine PRPFT (page 509), or the observed table values. When the fitted values are input, the estimates computed are the maximum likelihood estimates. When observed values are input, weighted least-squares estimates of the parameters in the log-linear model are computed. (Least-squares estimates and maximum likelihood estimates can also be computed via routines CTWLS (page 574) and CTGLM (page 557), respectively.)

When an expected count (as input in FIT) is zero, the cell is taken to be a structural zero. Such cells are not included in the weighted least-squares analysis. Estimates corresponding to structural zeros are set to the missing value indicator (NaN). To avoid this (and to determine the total degrees of freedom for each effect), add a positive constant such as 0.5 to each of the observed cell counts of zero, the “sampling” zeros. When structural zeros are present in the data the estimates may be written as

$$\hat{\beta} = \hat{\beta}_0 + \rho \hat{\beta}_1$$

where

$$\hat{\beta}, \hat{\beta}_0, \text{ and } \hat{\beta}_1$$

are vectors, and $\rho \rightarrow \infty$. Routine CTPAR estimates the finite portion of the estimate, $\hat{\beta}_0$. The infinite portion, $\hat{\beta}_1$ ensures that the fitted values for cells corresponding to structural zeros are

zero (sampling zeros are considered to be structural zeros in CTPAR). If there are no structural zeros

$$\hat{\beta}_i = 0$$

Let f_i denote the i -th element of the vector FIT. The asymptotic variance-covariance matrix of the cell counts is estimated by a diagonal matrix $S = \text{diag}(f)$ where $\text{diag}(f)$ denotes the diagonal matrix in which $s_{ij} = 0$ for $i \neq j$ and $s_{ii} = f_i$ along the diagonal. If X denotes the design matrix for the hierarchical model (with rows in X corresponding to structural zeros omitted), and $y_i = \log f_i$, then the weighted least-squares estimates are

$$\hat{\beta}_o = (X^T S^{-1} X)^{-1} X^T S^{-1} y$$

and the estimated variance-covariance matrix is

$$(X^T S^{-1} X)^{-1}$$

(see Grizzle, Starmer, and Koch [1969]).

If main effect A has, for example, four levels, then the design matrix X contains three dummy variables corresponding to this effect. Main effect dummy variables are generated as follows: For an observation f_i corresponding to level j of the effect, if $j < 3$, then the j -th dummy variable is set to 1 with the remaining dummy variables set to 0. If $j = 4$, then all three dummy variables are set to -1 . Dummy variables for interactions are generated as the product of the corresponding dummy variables in the usual manner with the smallest index in the specification of the interaction dummy varying fastest. The indices of the classification variables for each effect are always sorted from smallest to largest when computing the columns of X .

CTASC

Computes partial association statistics for log-linear models in a multidimensional contingency table.

Required Arguments

NCLVAL — Vector of length NCLVAR containing, in its i -th element, the number of levels or categories of the i -th classification variable. (Input)

TABLE — Vector of length NCLVAL(1) * NCLVAL(2) * ... * NCLVAL(NCLVAR) containing the entries in the cells of the table to be fit. (Input)

See Comment 3 for comments on the ordering of the elements of TABLE.

ZERO — Vector of length NCLVAL(1) * NCLVAL(2) * ... * NCLVAL(NCLVAR) indicating structural zeros in TABLE. (Input)

ZERO has the same structure as TABLE. Structural zeros in the TABLE are specified by setting the corresponding element of ZERO to 0.0. All other elements of zero must be positive. If structural zeros do not exist in TABLE, TABLE and ZERO can share the same storage locations. See Comment 3 for the ordering of the elements of ZERO.

ASSOC — $2^{\text{NCLVAR}} - 1$ by 4 matrix containing the partial association statistics for each effect in the model. (Output)

Column Statistic

- 1 Likelihood ratio partial association chi-squared for testing that all parameters in the effect are zero against a model containing all interactions of the same order
- 2 Degrees of freedom in chi-squared in columns 1 and 4
- 3 p -value for the chi-squared statistic in column 1
- 4 Number of zeros (structural and sampling) in the marginal table of the effect

The rows of **ASSOC** are ordered with main effects first, followed by two-way interactions, followed by the three-way interactions, etc., until the last row, which contains the single **NCLVAR**-way interaction. Thus, if there are 3 classification variables, there would be 8 rows in **ASSOC** and the rows would contain the A , B , C , AB , AC , BC , and the ABC effects where A represents the first (in **INDCL**) classification variable, B represents the second classification variable, etc.

CHHI — **NCLVAR** by 5 matrix containing chi-squared statistics testing that all k and higher interactions are zero where $k = 1, 2, \dots, \text{NCLVAR}$. (Output)
In the following, k is the row number of the statistic where the row numbers are 1, 2, ..., **NCLVAR**.

Col. Statistic

- 1 Likelihood ratio chi-squared statistic for testing that all interactions higher than k are zero against a model including all interactions of order k
- 2 p -value for the chi-squared value in column 1
- 3 Degrees of freedom for chi-squared in columns 1 and 4
- 4 Pearson chi-squared corresponding to column 1
- 5 p -value for the chi-squared value in column 4

CHSIM — **NCLVAR** by 5 matrix containing chi-squared statistics for testing that all k -factor interactions are simultaneously zero where $k = 1, \dots, \text{NCLVAR}$. (Output)
In the following, k is the row number of the statistic where the row numbers are 1, 2, ..., **NCLVAR**.

Col. Statistic

- 1 Likelihood ratio chi-squared statistic for testing that all k -factor interactions are all simultaneously zero given the model in which all k -way interactions are present
- 2 p -value for the chi-squared value in column 1
- 3 Degrees of freedom for chi-squared in columns 1 and 4
- 4 Pearson chi-squared corresponding to column 1
- 5 p -value for the chi-squared value in column 4

Optional Arguments

NCLVAR — Number of classification variables. (Input)

A variable specifying a margin in the table is a classification variable. The first classification variable is named A , the second classification variable is named B , etc.
Default: $NCLVAR = \text{size}(NCLVAL, 1)$.

EPS — Convergence criterion. (Input)

Convergence is assumed when the maximum deviation between an observed and a fitted marginal total is less than EPS . $EPS = 0.10$ is a typical value.
Default: $EPS = 0.10$.

MAXIT — Maximum number of iterations. (Input)

$MAXIT = 15$ is a typical value. When there are structural zeros a larger value, say $MAXIT = 100$, should be used.
Default: $MAXIT = 100$.

IPRINT — Printing option. (Input)

Default: $IPRINT = 0$.

IPRINT Action

- 0 No printing is performed.
- 1 Printing of $ASSOC$, $CHIH1$, and $CHISIM$ is performed.
- 2 $ASSOC$, $CHIH1$, $CHISIM$, and $TABLE$ are printed.

LDASSO — Leading dimension of $ASSOC$ exactly as specified in the dimension statement in the calling program. (Input)

Default: $LDASSO = \text{size}(ASSOC, 1)$.

LDCHIH — Leading dimension of **CHIHI** exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDCHIH = size (CHIHI,1)`.

LDCHIS — Leading dimension of **CHISIM** exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDCHIS = size (CHISIM,1)`.

FORTRAN 90 Interface

Generic: `CALL CTASC (NCLVAL, TABLE, ZERO, ASSOC, CHIHI, CHISIM [,...])`

Specific: The specific interface names are `S_CTASC` and `D_CTASC`.

FORTRAN 77 Interface

Single: `CALL CTASC (NCLVAR, NCLVAL, TABLE, ZERO, EPS, MAXIT, IPRINT, ASSOC, LDASSO, CHIHI, LDCHIH, CHISIM, LDCHIS)`

Double: The double precision name is `DCTASC`.

Example

The following example illustrates the use of `CTASC` for model building in a four-way table involving brand preference. The first three factors each have 2 levels, while the fourth factor has 3 levels. The data are originally from Lee (1977) and are printed in the output. A model with two-way interaction effects AD, BC, and BD looks promising.

```
USE CTASC_INT
INTEGER   IPRINT, LDASSO, LDCHIH, LDCHIS, LTAB
REAL      EPS
PARAMETER (EPS=0.01, IPRINT=2, LDASSO=15, LDCHIH=4, LDCHIS=4, &
           LTAB=24)
!
INTEGER   NCLVAL(4)
REAL      ASSOC(LDASSO,4), CHIHI(LDCHIH,5), CHISIM(LDCHIS,5), &
           TABLE(LTAB)
!
DATA TABLE/19, 57, 29, 63, 29, 49, 27, 53, 23, 47, 33, 66, 47, &
          55, 23, 50, 24, 37, 42, 68, 43, 52, 30, 42/
DATA NCLVAL/3, 2, 2, 2/
!
CALL CTASC (NCLVAL, TABLE, TABLE, ASSOC, CHIHI, CHISIM, &
           EPS=EPS, IPRINT=IPRINT)
!
END
```

Output

Variable	Number of Levels
1 A	3

2 B 2
 3 C 2
 4 D 2

 Table 1: B = 1 C = 1
 D (row) by A (column)
 1 2 3
 1 19.00 23.00 24.00
 2 57.00 47.00 37.00

 Table 2: B = 1 C = 2
 D (row) by A (column)
 1 2 3
 1 29.00 33.00 42.00
 2 63.00 66.00 68.00

 Table 3: B = 2 C = 1
 D (row) by A (column)
 1 2 3
 1 29.00 47.00 43.00
 2 49.00 55.00 52.00

 Table 4: B = 2 C = 2
 D (row) by A (column)
 1 2 3
 1 27.00 23.00 30.00
 2 53.00 50.00 42.00

Omitted Effect	Partial Association Statistics			Marginal Zeros
	Chi-Square	Degrees of Freedom	P-value	
A	0.50	2.0	0.7782	0.0
B	0.06	1.0	0.8010	0.0
C	1.92	1.0	0.1657	0.0
D	73.21	1.0	0.0000	0.0
A*B	0.22	2.0	0.8978	0.0
A*C	1.01	2.0	0.6050	0.0
A*D	6.10	2.0	0.0475	0.0
B*C	19.89	1.0	0.0000	0.0
B*D	3.74	1.0	0.0532	0.0
C*D	0.74	1.0	0.3898	0.0
A*B*C	4.57	2.0	0.1017	0.0
A*B*D	0.16	2.0	0.9223	0.0
A*C*D	1.38	2.0	0.5022	0.0
B*C*D	2.22	1.0	0.1361	0.0
A*B*C*D	0.74	2.0	0.6917	0.0

Chi-square statistics for testing that all k and higher interactions are zero.

k	Likelihood Ratio	P-Value	Degrees of Freedom	Pearson	P-Value
---	------------------	---------	--------------------	---------	---------

1	118.63	0.0000	23.0	115.71	0.0000
2	42.93	0.0008	18.0	43.90	0.0006
3	9.85	0.3631	9.0	9.87	0.3611
4	0.74	0.6917	2.0	0.74	0.6915

Chi-square statistics for testing that all k-factor interactions are simultaneously zero.

k	Likelihood Ratio	P-Value	Degrees of Freedom	Pearson	P-Value
1	75.70	0.0000	5.0	71.81	0.0000
2	33.08	0.0001	9.0	34.03	0.0001
3	9.11	0.2449	7.0	9.13	0.2433
4	0.74	0.6917	2.0	0.74	0.6915

Comments

1. Workspace may be explicitly provided, if desired, by use of C2ASC/DC2ASC. The reference is:

CALL C2ASC (NCLVAR, NCLVAL, TABLE, ZERO, EPS, MAXIT, IPRINT, ASSOC, LDASSO, CHIH, LDCHIH, CHISIM, LDCHIS, FITWK, NCVEF, IXEF, AMAR, INDX, WK, IWK, COVWK)

The additional arguments are as follows:

FITWK — Work vector of length $3 * NCLVAL(1) * \dots * NCLVAL(NCLVAR)$.

NCVEF — Work vector of length $2^{NCLVAR} - 1$.

IXEF — Work vector of length $NCLVAR * 2^{(NCLVAR * NCLVAR - 1)}$

AMAR — Work vector of length n . In defining n , let $q(k)$ be the sum of the sizes of all possible marginal tables with k effects. For example, $q(2)$ is the sum over all possible two-way interactions I and J of $NCLVAL(I) * NCLVAL(J)$ and $q(NCLVAR)$ is the product $NCLVAL(1) * \dots * NCLVAL(NCLVAR)$. Then, $n = \max(q(k), k = 1, \dots, NCLVAR)$.

INDX — Work vector of length m where m is the maximum number of interactions at any level. That is, $m = \max(\text{BINOM}(NCLVAR, I)), I = 1, \dots, NCLVAR$, where $\text{BINOM}(NCLVAR, I)$ is the binomial coefficient (see routine BINOM (IMSL MATH/LIBRARY Special Functions)).

WK — Work vector of length $3 * NCLVAL(1) * \dots * NCLVAL(NCLVAR)$ if there exists more than one structural zero in TABLE, and of length $NCLVAL(1) * \dots * NCLVAL(NCLVAR)$ otherwise.

IWK — Work vector of length $2 * NCLVAR$.

COVWK — Work vector of length $(NCLVAL(1) * \dots * NCLVAL(NCLVAR))^2$ if there exists more than one structural zero in TABLE. Otherwise, COVWK is not

referenced and can be dimensioned of length one in the calling program. On output, `COVWK` contains the upper triangular matrix containing the R matrix from a QR decomposition of the matrix of regressors for the full log-linear model.

2. Informational errors

Type Code

- 3 1 The optimization algorithm did not converge to the desired accuracy, `EPS`, within `MAXIT` iterations.
- 3 5 The label for one or more of the tables exceeds the buffer limit.
- 3 11 The label for one or more effects exceeds the buffer limit.

3. The cells of the vectors `TABLE` and `ZERO` are sequenced so that the first variable cycles from 1 to `NCLVAL(1)` the slowest, the second variable cycles from 1 to `NCLVAL(2)` the next slowest, etc., up to the `NCLVAR`-th variable, which cycles from 1 to `NCLVAL(NCLVAR)` the fastest.

Example: For `NCLVAR = 3`, `NCLVAL(1) = 2`, `NCLVAL(2) = 3`, and `NCLVAL(3) = 2`, the cells of table `X(I, J, K)` are entered into `TABLE(1)` through `TABLE(12)` in the following order:

`x(1, 1, 1)`, `x(1, 1, 2)`, `x(1, 2, 1)`, `x(1, 2, 2)`, `x(1, 3, 1)`, `x(1, 3, 2)`, `x(2, 1, 1)`, `x(2, 1, 2)`, `x(2, 2, 1)`, `x(2, 2, 2)`, `x(2, 3, 1)`, `x(2, 3, 2)`. The elements of `FIT` are similarly sequenced.

Description

Routine `CTASC` computes likelihood-ratio and Pearson χ^2 tests of partial-association for each effect in a hierarchical log-linear model. Also computed are likelihood ratio and Pearson chi-squared tests that all interactions above a given level are simultaneously zero. All of these tests are asymptotic in nature. All models are hierarchical so that all lower order interactions that may be composed from a higher order effect in the model are automatically included in the model. All models are fit via the iterative proportional fitting algorithm, which is implemented in routine `PRPFT` (page 509). The algorithm proceeds as follows:

1. The hierarchical model including all k -factor interactions is fit with $k = 0, \dots, m$ and $m = \text{NCLVAR}$. The $k = 0$ model corresponds to a constant probability in each cell in the table while the $k = m$ model is the full model. For each value of k , the likelihood ratio chi-squared statistic for testing that all interactions not included in the fitted model are all simultaneously zero (against the alternative that this is not the case) is computed as

$$2 \sum_i o_i \ln(o_i / f_i)$$

where o_i is the observed count in the i -th cell, f_i is the fitted count for the given model, and the summation is over all cells in the table. Also computed (for comparison, the two statistics are asymptotically equivalent) is the usual Pearson chi-squared statistic,

$$\sum_i (o_i - f_i)^2 / f_i$$

2. Let $g_i = \text{NCLVAL}(i)$, and let

$$t = \prod_{i=1}^m g_i$$

and assume that there are no structural zeros in the table. Then, the number of degrees of freedom in the chi-squared statistic for testing that all k -order interactions are simultaneously zero is the sum over all k -th order interaction effects of the degrees of freedom for the effect. In the no structural zero case, the degrees of freedom for an effect may be computed as

$$\prod_j (g_j - 1)$$

where j indexes the factors in the effect. Denote the sum of these degrees of freedom at level k by s_k , and let $s_0 = 1$. Then, the degrees of freedom in the k -th test is given by s_k .

When more than one structural zero is present, the degrees of freedom in the chi-squared statistics are computed by fitting a least-squares model for the full hierarchical model in which all interactions are included. Routine RGIVN (see Chapter 2, Regression) is used in fitting the model. Cells with sampling (as opposed to structural) zeros are included (but only when degrees of freedom are computed) by using a cell count of 0.5. Observations corresponding to structural zeros are not included. (Note that a structural zero is a model restriction that requires that the estimated count for a cell be zero. A sampling zero occurs by chance.) The degrees of freedom for each effect are found by summing over the estimated parameters for the effect. Parameters that are linearly related to previous parameters in the model (as determined through RGIVN via input argument TOL where TOL is 100 * AMACH(4)) are not estimated. When there is only one structural zero, degrees of freedom are computed as if there were no structural zeros except for the highest level interaction term, which is given one fewer degree of freedom.

Chi-squared statistics for testing that all effects at a given level k are simultaneously zero (given a hierarchical model in which all effects above level k are absent) are computed as the difference between the chi-squared statistics testing that all k and higher interactions are zero and that of $k + 1$. That is, for $J = 1$ and 4, and $I = 1, 2, \dots, NCLVAR - 1$, then $CHISIM(I, J) = CHIHI(I, J) - CHIHI(I + 1, J)$, and $CHISIM(NCLVAR, J) = CHIHI(NCLVAR, J)$.

3. For each effect, a “partial association” likelihood ratio chi-squared statistic may be used to test the hypothesis that all parameters in the effect are simultaneously zero, given a model in which all interactions at the same level (or lower) are present, and all higher level interactions are absent. The degrees of freedom for the effect are computed as in Step 2.

Programming Notes

1. When sampling zeros are present, the likelihood ratio test statistics may not follow the appropriate chi-squared distribution closely. A common (but not necessarily the best) practice in this case is to add a small positive constant, often 0.5, to each cell in the table. This addition is easily accomplished via routine SADD (IMSL MATH/LIBRARY). The addition of such a constant should not effect the computed degrees of freedom.

- When marginal totals of zero are obtained, the optimization algorithm may be slow to converge. In this case, increase the value of argument `MAXIT`.

CTSTP

Builds hierarchical log-linear models using forward selection, backward selection, or stepwise selection.

Required Arguments

NCLVAL — Vector of length `NCLVAR` containing, in its i -th element, the number of levels or categories of the i -th classification variable. (Input)

TABLE — Vector of length `NCLVAL(1) * NCLVAL(2) * ... * NCLVAL(NCLVAR)` containing the entries in the cells of the table to be fit. (Input)

See Comment 3 for comments on the ordering of the elements of `TABLE`.

ISTEP — Stepping option. (Input)

ISTEP	Action
-1	An attempt is made to remove an effect from the model (a backward step). An effect is removed if it has the largest p -value among all effects considered for removal with p -value exceeding <code>POUT</code> .
0	A backward step is attempted. If a variable is not removed, a forward step is attempted. This is a stepwise step.
1	An attempt is made to add an effect to the model (a forward step). An effect is added if it has the smallest p -value among all effects with p -value less than <code>PIN</code> .

NSTEP — Step length option. (Input)

For nonnegative `NSTEP`, `NSTEP` steps are taken. Less than `NSTEP` are taken if no effect that can enter or leave the model meets the `PIN` or `POUT` criterion. Use `NSTEP = -1` to indicate that stepping is to continue until no effect meets the `PIN` or `POUT` criterion to enter or leave the model.

NFORCE — The number of initial effects in the model that must be included in any model considered. (Input)

For `NFORCE = k`, the first k effects specified by `NEF`, `NVEF`, and `INDEF` will be included in all models considered.

NEF — Number of effects in the model. (Input/Output)

A marginal table is implied by each effect in the model. Lower order effects should not be included in the model specification since their inclusion is automatic (e.g., do not include effects A or B if effect AB is in the model). On input, `NEF` gives the number of

effects in the initial model. On output, *NEF* gives the number of effects in the final model.

NVEF — Vector of length *MAXNVF* containing the number of classification variables associated with each effect. (Input/Output)
On input, *NVEF* contains the number of classification variables for each effect in the initial model. The final values are returned on output.

INDEF — Vector of length *MAXIND* containing, in consecutive positions, the indices of the variables that are included in each effect. (Input/Output)
The entries in *INDEF* are sequenced so that the first *NVEF*(1) elements contain the indices of the variables in effect 1, the next *NVEF*(2) elements of *INDEF* contain the indices of the variables in effect 2, etc. Each element of *INDEF* must be greater than zero. See Comment 4 for an example.

FIT — Vector of length *NCLVAL*(1) * *NCLVAL*(2) * ... * *NCLVAL*(*NCLVAR*) containing the model estimates of the cell counts. (Input/Output)
On input, *FIT* contains the initial estimates of the cell counts. Structural zeros in the model are specified by setting the corresponding element of *FIT* to 0.0. All other elements of *FIT* may be set to 1.0 if no other estimate of the expected cell counts is available. On output, *FIT* contains the fitted table. See Comment 3 for the ordering of the elements of *FIT*. If an element of *FIT* is positive but the corresponding element in *TABLE* is zero, the the element is called a sampling zero. Sampling zeros may effect the number of parameters that can be estimated, but they will not effect the degrees of freedom in chi-squared tests. See the “Algorithm” section of the manual document.

STAT — Vector of length 3 containing some output statistics for the final model fit during this invocation. (Output)

I **STAT(I)**

- 1 Asymptotic chi-squared statistic based upon likelihood ratios for testing that the current model fits the observed data.
- 2 Degrees of freedom in chi-squared. This is the number of cells in the table minus the number of structural zeros minus the degrees of freedom for the model.
- 3 Probability of a greater chi-squared.

IEND — Completion indicator. (Output)

IEND **Meaning**

- 0 Additional steps may be possible.
- 1 No additional steps are possible for the values of *PIN* and *POUT*.

Optional Arguments

IDO — Processing option. (Input)

Default: `IDO = 0`.

IDO Action

- 0 This is the only invocation of `CTSTP` for this table. If there are sampling zeros, set up for computing the degrees of freedom for each effect. Perform `NSTEP` steps (if `ISTEP`, `POUT`, and `PIN` allow it) and then release all workspace.
- 1 This is the first invocation, and additional calls to `CTSTP` will be made. Set up for computing the degrees of freedom for each effect and then perform `NSTEP` steps (if `ISTEP`, `POUT`, and `PIN` allow it).
- 2 This is an intermediate invocation of `CTSTP`. Perform `NSTEP` steps (if `ISTEP`, `POUT`, and `PIN` allow it).
- 3 This is the final invocation of this routine. Perform `NSTEP` steps (if `ISTEP`, `POUT`, and `PIN` allow it). Release all workspace.

NCLVAR — Number of classification variables. (Input)

A variable specifying a margin in the table is a classification variable. The first classification variable is named *A*, the second classification variable is named *B*, etc. Default: `NCLVAR = size(NCLVAL,1)`.

PIN — Largest *p*-value for entering variables. (Input)

Variables with *p*-values less than `PIN` may enter the model. The choice 0.05 is common.

Default: `PIN = .05`.

POUT — Smallest *p*-value for removing variables. (Input)

Variables with *p*-values greater than `POUT` may leave the model. `POUT` must be greater than or equal to `PIN`. The choice 0.10 is common.

Default: `POUT = .10`.

IPRINT — Printing option. (Input)

Default: `IPRINT = 0`.

IPRINT Action

- 0 No printing is performed.
- 1 Printing of the initial and final model summary statistics and step summaries.
- 2 Printing of the input table is performed followed by printing of the initial and final model summary statistics and of the step summaries.

MAXNVF — The maximum length of `NVEF` as specified in the dimension statement in the calling program. (Input)
 If the required length of `NVEF` becomes greater than `MAXNVF`, a type 4 error message is issued and the final model chosen is returned in `NEF`, `NVEF`, and `INDEF`. See Comment 2.
 Default: `MAXNVF = size (NVEF,1)`.

MAXIND — The maximum possible length of `INDEF` as specified in the dimension statement in the calling program. (Input)
 If the required length of `INDEF` becomes greater than `MAXIND`, a type 4 error message is issued and the final model chosen is returned in `NEF`, `NVEF`, and `INDEF`. See Comment 2.
 Default: `MAXIND = size (INDEF,1)`.

FORTRAN 90 Interface

Generic: `CALL CTSTP (NCLVAL, TABLE, ISTEP, NSTEP, NFORCE, NEF, NVEF, INDEF, FIT, STAT, IEND [,...])`

Specific: The specific interface names are `S_CTSTP` and `D_CTSTP`.

FORTRAN 77 Interface

Single: `CALL CTSTP (IDO, NCLVAR, NCLVAL, TABLE, PIN, POUT, ISTEP, NSTEP, NFORCE, IPRINT, NEF, NVEF, MAXNVF, INDEF, MAXIND, FIT, STAT, IEND)`

Double: The double precision name is `DCTSTP`.

Example 1

The following example is taken from Lee (1977). It involves a simple four-way table in which the first three factors have 2 levels, and the fourth factor has 3 levels. The data involves brand preference in different situations. In the example, the three-way interaction is removed, leaving 3 two-way interactions. In the new model, the three-way interaction is omitted.

```

USE UMACH_INT
USE CTSTP_INT
USE WRIRN_INT
USE WRRRN_INT
USE ISUM_INT

INTEGER IPRINT, LTAB, MAXIND, MAXNVF, NCLVAR
PARAMETER (IPRINT=2, LTAB=24, MAXIND=20, MAXNVF=10, NCLVAR=4)
!
INTEGER IEND, INDEF(MAXIND), ISTEP, LIND, NCLVAL(NCLVAR), &
NEF, NFORCE, NOUT, NSTEP, NVEF(MAXNVF)
REAL FIT(LTAB), STAT(3), TABLE(LTAB)
!
DATA TABLE/19.0, 57.0, 29.0, 63.0, 29.0, 49.0, 27.0, 53.0, 23.0, &

```



```

          47.0, 33.0, 66.0, 47.0, 55.0, 23.0, 50.0, 24.0, 37.0, 42.0, &
          68.0, 43.0, 52.0, 30.0, 42.0/
DATA NCLVAL/3, 2, 2, 2/, FIT/24*1.0/
DATA NEF/1/
!
CALL UMACH (2, NOUT)
!
ISTEP      = 0
NSTEP      = 1
NFORCE     = 0
NVEF(1)    = 3
INDEF(1)   = 1
INDEF(2)   = 2
INDEF(3)   = 4
!
CALL CTSTP (NCLVAL, TABLE, ISTEP, NSTEP, NFORCE, NEF, &
           NVEF, INDEF, FIT, STAT, IEND, IPRINT=IPRINT)
!
WRITE (NOUT,99999) IEND, NEF
CALL WRIRN ('NVEF', NVEF, 1, NEF, 1, 0)
LIND = ISUM(NEF,NVEF,1)
CALL WRIRN ('INDEF', INDEF, 1, LIND, 1, 0)
CALL WRRRN ('FIT', FIT, 1, LTAB, 1, 0)
!
99999 FORMAT (/, ' IEND = ', I3, '   NEF = ', I3)
END

```

Output

Variable	Number of Levels
1 A	3
2 B	2
3 C	2
4 D	2

```

-----
Table 1: B = 1 C = 1
D (row) by A (column)
      1      2      3
1  19.00  23.00  24.00
2  57.00  47.00  37.00

```

```

-----
Table 2: B = 1 C = 2
D (row) by A (column)
      1      2      3
1  29.00  33.00  42.00
2  63.00  66.00  68.00

```

```

-----
Table 3: B = 2 C = 1
D (row) by A (column)
      1      2      3
1  29.00  47.00  43.00
2  49.00  55.00  52.00

```

```

-----
Table 4: B = 2 C = 2
D (row) by A (column)
      1      2      3
1  27.00  23.00  30.00
2  53.00  50.00  42.00

```

```

----- Step: 0 -----
Input Model: (A*B*D)
Smallest p-value for removing effects      0.100
Largest p-value for entering effects      0.050
Chi-squared                               33.92
Degrees of Freedom                        12.
p-value                                   0.0007

Effect Tested      Chi-squared      Degrees of      P-value
A*B*D              0.12              2              0.9408
Effect Removed: A*B*D

```

```

----- Step: 1 -----
Model: (A*B, A*D, B*D)
Chi-squared                34.05
Degrees of Freedom          14.
p-value                    0.0020

IEND = 0   NEF = 3

```

```

NVEF
1  2  3
2  2  2

```

```

INDEF
1  2  3  4  5  6
1  2  1  4  2  4

```

```

FIT
      1      2      3      4      5      6      7      8      9     10
24.39  59.61  24.39  59.61  27.61  51.39  27.61  51.39  28.24  56.26

     11     12     13     14     15     16     17     18     19     20
28.24  56.26  34.76  52.74  34.76  52.74  32.38  53.12  32.38  53.12

     21     22     23     24
37.12  46.38  37.12  46.38

```

Comments

1. Workspace may be explicitly provided, if desired, by use of C2STP/DC2STP. The reference is:

```

CALL C2STP (IDO, NCLVAR, NCLVAL, TABLE, PIN, POUT, ISTEP, NSTEP,
NFORCE, NEF, IPRINT, NVEF, MAXNVF, INDEF, MAXIND, FIT, STAT,
IEND, MAXMAR, AMAR, INVEF, IINDEF, IDF, ZWK, RWK, IWK)

```

The additional arguments are as follows:

MAXMAR — The length of *AMAR*. (Input)

When workspace is allocated by *CTSTP*, *MAXMAR* is equal to the number of workspace elements remaining after all other workspace is allocated. *MAXMAR* should be chosen as the maximum over all models considered of the sum over all marginal tables in the model of the number of elements in each marginal table.

AMAR — Work vector of length *MAXMAR* used to store marginal means in the proportional fitting algorithm. (Output)

INVEF — Work vector whose length is dependent on *ISTEP*, *IPRINT*, and z = the number of structural zeros in *TABLE*.

ISTEP	IPRINT	z	Length of INVEF
-1, 0, 1	0, 1, 2	$z > 1$	$3v$
0, 1	0, 1, 2	$z \leq 1$	$3v$
-1	0	$z \leq 1$	$2v$

Here, $v = 2^{\text{NCLVAR}} - 1$.

IINDEF — Work vector whose length is dependent on *ISTEP*, *IPRINT*, and z = the number of structural zeros in *TABLE*.

ISTEP	IPRINT	z	Length of IINDEF
-1, 0, 1	0, 1, 2	$z > 1$	$3d$
0, 1	0, 1, 2	$z \leq 1$	$3d$
-1	0	$z \leq 1$	$2d$

Here, $d = \text{NCLVAR} * 2^{\text{NCLVAR}-1}$.

IDF — Vector of length $n + z$. (Output, for *IDO* = 0 or 1; input/output otherwise)

Here, $n = \text{NCLVAL}(1) * \text{NCLVAL}(2) * \dots * \text{NCLVAL}(\text{NCLVAR})$. If there are no structural zeros in *TABLE*, *IDF* is not referenced and may be dimensioned of length 1 in the calling program. When using the *IDO* = 1, 2, ..., 2, 3 option, the values stored in *IDF* should not be altered between calls to *C2STP*.

ZWK — Vector of length $n(n + 2)$. (Output, for *IDO* = 0 or 1; input/output otherwise)

Here, $n = \text{NCLVAL}(1) * \text{NCLVAL}(2) * \dots * \text{NCLVAL}(\text{NCLVAR})$. If there are no structural zeros in *TABLE*, *ZWK* is not referenced and may be dimensioned of length 1 in the calling program. When using the *IDO* = 1, 2, ..., 2, 3 option, the values stored in *ZWK* should not be altered between calls to *C2STP*.

RWK — Work vector whose length is dependent on *IDO* and z , the number of structural zeros in *TABLE*.

IDO	Z	Length of RWK
0, 1	$z > 1$	$2n + m$
0, 1	$z \leq 1$	n
2, 3	$z > 1$	n
2, 3	$z \leq 1$	n

Here, $n = \text{NCLVAL}(1) * \text{NCLVAL}(2) * \dots * \text{NCLVAL}(\text{NCLVAR})$ and
 $m = \text{NCLVAL}(1) + \text{NCLVAL}(2) + \dots + \text{NCLVAL}(\text{NCLVAR})$.

IRK — Work vector whose length is dependent on **ISTEP** and **NSTEP**.

ISTEP	NSTEP	Length of IRK
-1, 0	$\text{NSTEP} = 0$	$3 * \text{NCLVAR} + \text{NEF}$
-1, 0	$\text{NSTEP} \neq 0$	$3 * \text{NCLVAR} + \nu$
1	$\text{NSTEP} = 0$	$2 * \text{NCLVAR} + \text{NEF}$
1	$\text{NSTEP} \neq 0$	$2 * \text{NCLVAR} + \nu$

Here, $\nu = 2^{\text{NCLVAR}-1}$.

2. Informational errors

Type Code

- | | | |
|---|---|---|
| 3 | 1 | The proportional fitting algorithm did not converge. |
| 4 | 2 | There is not enough workspace allocated for storing the marginal means. |
| 4 | 3 | The required length of NVEF to store the effects of the new model exceeds MAXNVF . |
| 4 | 4 | The required length of INDEF to store the effects of the new model exceeds MAXIND . |

3. The cells of the vectors **TABLE**, and **FIT** are sequenced so that the first variable cycles from 1 to **NCLVAL**(1) the slowest, the second variable cycles from 1 to **NCLVAL**(2) the next slowest, etc., up to the **NCLVAR**-th variable, which cycles from 1 to **NCLVAL**(**NCLVAR**) the fastest.

Example: For **NCLVAR** = 3, **NCLVAL**(1) = 2, **NCLVAL**(2) = 3, and **NCLVAL**(3) = 2, the cells of table **X**(**I**, **J**, **K**) are entered into **TABLE**(1) through **TABLE**(12) in the following order:

X(1, 1, 1), **X**(1, 1, 2), **X**(1, 2, 1), **X**(1, 2, 2), **X**(1, 3, 1), **X**(1, 3, 2), **X**(2, 1, 1), **X**(2, 1, 2), **X**(2, 2, 1), **X**(2, 2, 2), **X**(2, 3, 1), **X**(2, 3, 2). The elements of **FIT** are similarly sequenced.

4. **INDEF** is used to describe the marginal tables to be fit. For example, if **NCLVAR** = 3 and the first effect is to fit the marginal table for variables 1 and 3 and the second effect is to fit the marginal table for variable 2, then: **NEF** = 2, **NVEF**(1) = 2, and **NVEF**(2) = 1. Since the sum of the **NVEF**(**I**) is 3, then **INDEF** is a vector of length 3 with values: **INDEF**(1) = 1, **INDEF**(2) = 3, and **INDEF**(3) = 2.

Description

Routine CTSTP performs stepwise model building in hierarchical log-linear models. CTSTP handles structural and sampling zeros, and allows “downward,” “upward,” or “stepwise” stepping. For $NFORCE > 0$, the leading $NFORCE$ effects in the initial model specified in NEF, NVEF, and INDEF are forced to remain in the model. A variable number (NSTEP) of steps from the input model are performed during a single invocation of CTSTP. Printing of the input table and intermediate results is performed if requested.

In hierarchical models, lower order effects are automatically included whenever a higher order effect containing the lower order effect is in the model. That is, the model (AB) automatically includes the mean and the main effects A and B , and the model (AB, ACD) automatically includes the lower order effects A, B, C, D, AC, AD , and CD .

The algorithm proceeds through the following steps during a single invocation when $IDO = 0$. For $IDO > 0$, these steps are still followed, but they may require more than one invocation of the routine.

1. The input model is fit. The current model is set to the input model.
2. If downward stepping is to be performed ($ISTEP = -1$ or $ISTEP = 0$), then each effect in the model is examined to determine if it can be deleted from the current model. An effect may be deleted from the current model if it is not a “forced effect” and if it must be included in the hierarchical specification of the model (in which lower order terms are not specified). Thus, for example, the effect ABC can be deleted from the model (ABC, BCD) , yielding a model (AB, AC, BCD) , but not from the model $(ABCD)$ since ABC is not included in the hierarchical specification.

For each effect that can be deleted in a downward step, the usual chi-squared likelihood-ratio test statistic is computed as twice the difference of the log-likelihoods between the current model and the model in which the effect has been deleted. The degrees of freedom for the effect are determined (see below), and an asymptotic p -value is computed via the chi-squared distribution. After the p -values for all deleted models have been determined, the maximum p -value is selected. If it is greater than the p -value for deletion, POUT, the effect is deleted from the model, and the resulting model is fit.

3. If a downward step is not possible, either because all computed p -values are too small or because downward stepping is not to be performed, an upward step is attempted if requested ($ISTEP = 0$ or $ISTEP = 1$). For upward stepping, each effect in the full factorial analysis of variance specification of the table is examined to determine if the effect differs from the current model by exactly one term. For example, (ABC) differs by one term from the model (AB, AC, BC) and from the model (ABD, ACD, BCD) , but it differs by more than one term from the model (AB, BC) .

For each effect that may be added to the model, a chi-squared likelihood-ratio test statistic is computed comparing the current model to the model with the added effect, its degrees of freedom are determined (see below), and an asymptotic p -value based upon the chi-squared distribution is computed. After all p -values for models with additive effects have been computed, the model with the minimum p -value is

determined. If the minimum p -value is less than the p -value for addition, PIN, then the effect is added to the model, and the resulting model is fit.

4. If neither a step down, nor a step up can be performed, then CTSTP sets IEND = 1 and returns the original model to the user. Otherwise, if additional steps are to be made, execution continues at Step 2 above.

Degrees of Freedom

In CTSTP, structural zeros are considered to be a restriction of the parameter space. As such, they subtract from the degrees of freedom for an effect. Alternatively, sampling zeros are a result of sampling, and thus, they do not subtract from the degrees of freedom or restrict the parameter space. When computing degrees of freedom, sampling zeros are treated as if they were positive counts. If there are no structural zeros, then the degrees of freedom are computed as the product of the degrees of freedom for each variable in the effect where the degrees of freedom for the variable is the number of levels for the variable minus one. When structural zeros are present, there are restrictions on the parameter space, and the degrees of freedom for an effect are computed as the number of non-zero diagonal elements corresponding to the effect along the Cholesky factorization of the $X^T X$ matrix where X is the “design matrix” for the model. That is, each row of X contains the indicator variables for a cell in the table, with the indicator variables for the current model preceding the indicator variables for the effect for which degrees of freedom are desired. Because the degrees of freedom for an effect must be relative to the model, when there are structural zeros, it is possible for the degrees of freedom for an effect to change from one step to the next.

Example 2

Example two illustrates the use of CTSTP when sampling zeros are present. In this example, which is taken from Brown and Fuchs (1983), there are thirteen sampling zeros so that thirteen parameter estimates are infinite when the full model is fit. Here, we begin with the model fit by Brown and Fuchs, which, in CTSTP notation, is given as

$(AC, AD, ABE, BCDE)$

When this model is fit, there are five parameter estimates that are infinite. Note that these estimates have no effect on the degrees of freedom used in the tests computed here.

```

USE UMACH_INT
USE CTSTP_INT
USE WRIRN_INT
USE WRRRN_INT
USE ISUM_INT

INTEGER   IPRINT, LTAB, MAXIND, MAXNVF, NCLVAR
PARAMETER (IPRINT=2, LTAB=32, MAXIND=30, MAXNVF=10, NCLVAR=5)
!
INTEGER   IDO, IEND, INDEF(MAXIND), ISTEP, LIND, NCLVAL(NCLVAR), &
          NEF, NFORCE, NOUT, NSTEP, NVEF(MAXNVF)
REAL      FIT(LTAB), STAT(3), TABLE(LTAB)
!
DATA TABLE/33.0, 32.0, 8.0, 8.0, 0.0, 1.0, 1.0, 0.0, 0.0, 1.0, &
          0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 2.0, 10.0, 3.0, 6.0, 1.0, &

```

```

      2.0, 0.0, 2.0, 0.0, 1.0, 0.0, 4.0, 0.0, 1.0, 0.0, 2.0/
DATA NCLVAL/2, 2, 2, 2, 2/, FIT/32*1.0/, NEF/4/
DATA (NVEF(I),I=1,4)/2, 2, 3, 4/
DATA (INDEF(I),I=1,11)/1, 3, 1, 4, 1, 2, 5, 2, 3, 4, 5/
!
CALL UMACH (2, NOUT)
!
ISTEP = -1
NSTEP = 2
NFORCE = 0
!
CALL CTSTP (NCLVAL, TABLE, ISTEP, NSTEP, NFORCE, NEF, &
           NVEF, INDEF, FIT, STAT, IEND, IPRINT=IPRINT)
!
WRITE (NOUT,99999) IEND, NEF
CALL WRIRN ('NVEF', NVEF, 1, NEF, 1, 0)
LIND = ISUM(NEF,NVEF,1)
CALL WRIRN ('INDEF', INDEF, 1, LIND, 1, 0)
CALL WRRRN ('FIT', FIT, 1, LTAB, 1, 0)
!
99999 FORMAT (/, ' IEND = ', I3, '   NEF = ', I3)
END

```

Output

Variable	Number of Levels
1 A	2
2 B	2
3 C	2
4 D	2
5 E	2

Table 1: A = 1 B = 1 C = 1

D (row) by E (column)		
	1	2
1	33.00	32.00
2	8.00	8.00

Table 2: A = 1 B = 1 C = 2

D (row) by E (column)		
	1	2
1	0.000	1.000
2	1.000	0.000

Table 3: A = 1 B = 2 C = 1

D (row) by E (column)		
	1	2
1	0.000	1.000
2	0.000	0.000

Table 4: A = 1 B = 2 C = 2

D (row) by E (column)		
-----------------------	--	--

		1	2
1	0.000	1.000	
2	0.000	0.000	

Table 5: A = 2 B = 1 C = 1
D (row) by E (column)

		1	2
1	2.00	10.00	
2	3.00	6.00	

Table 6: A = 2 B = 1 C = 2
D (row) by E (column)

		1	2
1	1.000	2.000	
2	0.000	2.000	

Table 7: A = 2 B = 2 C = 1
D (row) by E (column)

		1	2
1	0.000	1.000	
2	0.000	4.000	

Table 8: A = 2 B = 2 C = 2
D (row) by E (column)

		1	2
1	0.000	1.000	
2	0.000	2.000	

----- Step: 0 -----

Input Model: (A*C, A*D, A*B*E, B*C*D*E)
Smallest p-value for removing effects 0.100
Chi-squared 9.07
Degrees of Freedom 10.
p-value 0.5251

Effect Tested	Chi-squared	Degrees of Freedom	P-value
A*C	4.41	1	0.0358
A*D	6.56	1	0.0104
A*B*E	0.00	1	0.9912
B*C*D*E	0.00	1	0.9912

Effect Removed: B*C*D*E

----- Step: 1 -----

Model: (A*C, A*D, A*B*E, B*C*D, B*C*E, B*D*E, C*D*E)
Chi-squared 9.07
Degrees of Freedom 11.
p-value 0.6151

Effect Tested	Chi-squared	Degrees of Freedom	P-value
---------------	-------------	--------------------	---------

A*C	4.41	1	0.0358
A*D	6.56	1	0.0104
A*B*E	0.00	1	1.0000
B*C*D	0.53	1	0.4673
B*C*E	0.00	1	1.0000
B*D*E	0.00	1	1.0000
C*D*E	0.10	1	0.7522

Effect Removed: B*C*E

```

----- Step: 2 -----
Model: (A*C, A*D, A*B*E, B*C*D, B*D*E, C*D*E)
Chi-squared          9.07
Degrees of Freedom   12.
p-value              0.6966
IEND = 0   NEF = 6

```

NVEF					
1	2	3	4	5	6
2	2	3	3	3	3

INDEF															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	3	1	4	1	2	5	2	3	4	2	4	5	3	4	5

FIT									
1	2	3	4	5	6	7	8	9	10
32.36	32.56	8.53	6.91	0.71	1.21	0.40	0.32	0.00	0.90
11	12	13	14	15	16	17	18	19	20
0.00	0.75	0.00	0.27	0.00	0.09	2.64	9.44	2.47	7.09
21	22	23	24	25	26	27	28	29	30
0.29	1.79	0.60	1.68	0.00	1.10	0.00	3.25	0.00	1.73
31	32								
0.00	1.91								

CTRAN

Performs generalized Mantel-Haenszel tests in a stratified contingency table.

Required Arguments

NCLVAL — Vector of length NCLVAR containing, in its *i*-th element, the number of levels (categories) of the *i*-th classification variable. (Input)

TABLE — Vector of length NCLVAL(1) * NCLVAL(2) * ... * NCLVAL(NCLVAR) containing the entries in the cells of the table to be fit. (Input)

See Comment 3 for comments on the ordering of the elements in TABLE. For the classification variables specified in INDROW and INDCOL, a series of two-dimensional contingency tables are obtained from the elements in TABLE. All other classification variables are stratification variables.

INDROW — Index of the classification variable to be used for the row variable in the stratified two-dimensional table. (Input)

INDCOL — Index of the classification variable to be used for the column variable in the stratified two-dimensional table. (Input)

ROWSCR — Vector of length $NCLVAL(INDCOL)$ containing the scores associated with the column and used in each row. (Input, if $IROWSC = 0$; output, otherwise)
ROWSCR is not used and can be dimensioned of length 1 in the calling program if $ITYPE = 1$. If $IROWSC$ is 3, 4, or 5, then ROWSCR contains the scores used in the last contingency table analyzed.

COLSCR — Vector of length $NCLVAL(INDROW)$ containing the scores associated with each row and used in each column. (Input, if $ICOLSC = 0$; output, otherwise)
COLSCR is not used and can be dimensioned of length 1 in the calling program if $ITYPE$ is not 3. If $ICOLSC$ is 3, 4, or 5, then COLSCR contains the scores used in the last contingency table analyzed.

STAT — Table of size m by 3 containing the Mantel-Haenszel statistics. (Output)
Where m is one plus the number of stratified tables, i.e., $m = 1 + NCLVAL(1) * NCLVAL(2) * \dots * NCLVAL(NCLVAR)/(NCLVAL(INDROW) * NCLVAL(INDCOL))$. The first column of **STAT** contains the chi-squared statistic for a test of partial association, the second column contains its degrees of freedom, and the third column contains the probability of a greater chi-squared. The first $m - 1$ rows of **STAT** contain the statistics computed for each of the stratified tables. The first row corresponds to the classification stratification variable levels (1, 1, ..., 1). The second row corresponds to levels (1, 1, ..., 2), etc., so that in row $m - 1$ all stratification variables are at their highest levels. The last row of **STAT** contains the same statistics pooled over all of the stratified tables.

Optional Arguments

NCLVAR — Number of classification variables. (Input)
Default: $NCLVAR = \text{size}(NCLVAL, 1)$.

ITYPE — The type of statistic to compute. (Input)
Default: $ITYPE = 1$.

ITYPE	Statistic
1	Generalized Mantel-Haenszel based upon the two-dimensional contingency tables.
2	Generalized Mantel-Haenszel based upon the row mean score in the two-dimensional table.

- 3 Generalized Mantel-Haenszel based upon the correlation score for the two-dimensional tables.

IROWSC — Option parameter giving the scores associated with the column index to be used when computing statistics in each row. (Input)
Default: `IROWSC = 0`.

***IROWSC* Weights**

- 0 User specified (or no) weights.
- 1 The digits 1, 2, ..., `NCLVAL(INDCOL)`.
- 2 Combined (over all tables) ridity-type scores.
- 3 Rank scores computed separately for each table.
- 4 Ridity-type scores computed separately for each table.
- 5 Logrank scores computed separately for each table.

`IROWSC` is not used if `ITYPE = 1`.

ICOLSC — Option parameter giving the scores associated with the row index to be used when computing statistics in each column. (Input)
Default: `ICOLSC = 0`.

***ICOLSC* Weights**

- 0 User specified (or no) weights.
- 1 The digits 1, 2, ..., `NCLVAL(INDROW)`.
- 2 Combined (over all tables) ridity-type scores.
- 3 Rank scores computed separately for each table.
- 4 Ridity-type scores computed separately for each table.
- 5 Logrank scores computed separately for each table.

`ICOLSC` is not used if `ITYPE` is not 3.

IPRINT — Print option. (Input)
Default: `IPRINT = 0`.

IPRINT	Action
0	No printing.
1	Print the contents of the <code>STAT</code> array.
2	Print each stratified table followed by the contents of the <code>STAT</code> array.

LDSTAT— Leading dimension of `STAT` exactly as specified in the dimension statement of the calling program. (Input)
 Default: `LDSTAT = size (STAT,1)`.

FORTRAN 90 Interface

Generic: `CALL CTRAN (NCLVAL, TABLE, INDROW, INDCOL, ROWSCR, COLSCR, STAT [,...])`

Specific: The specific interface names are `S_CTRAN` and `D_CTRAN`.

FORTRAN 77 Interface

Single: `CALL CTRAN (NCLVAR, NCLVAL, TABLE, INDROW, INDCOL, ITYPE, IROWSC, ICOLSC, IPRINT, ROWSCR, COLSCR, STAT, LDSTAT)`

Double: The double precision name is `DCTRAN`.

Example

In the following example, all three values of `ITYPE` are used in computing the partial association statistics. This is accomplished via three calls to `CTRAN`. The value of `ITYPE` changes on each call. The example is taken from Landis, Cooper, Kennedy, and Koch (1979, page 241). Uniform scores are used in both the rows and column as required by the tests type. The results indicate the presence of association between the row and column variables.

```

USE CTRAN_INT
INTEGER   ICOLSC, INDCOL, INDROW, IROWSC, LDSTAT, NCLVAR
PARAMETER (ICOLSC=1, INDCOL=1, INDROW=3, IROWSC=1, LDSTAT=5, &
           NCLVAR=3)
!
INTEGER   IPRINT, ITYPE, NCLVAL(NCLVAR)
REAL      COLSCR(4), ROWSCR(3), STAT(LDSTAT,3), TABLE(48)
!
DATA TABLE/23, 23, 20, 24, 18, 18, 13, 9, 8, 12, 11, 7, 12, 15, &
           14, 13, 7, 10, 13, 10, 6, 6, 13, 15, 6, 4, 6, 7, 9, 3, 8, &
           6, 2, 5, 5, 6, 1, 2, 2, 2, 3, 4, 2, 4, 1, 2, 3, 4/
DATA NCLVAL/3, 4, 4/
!
IPRINT = 2
DO 10 ITYPE=1, 3
    CALL CTRAN (NCLVAL, TABLE, INDROW, INDCOL, &

```

```

        ROWSCR, COLSCR, STAT, ITYPE=ITYPE, &
        IROWSC=IROWSC, ICOLSC=ICOLSC, IPRINT=IPRINT)
    IPRINT = 1
!
10 CONTINUE
    END

```

Output

Values for the class variables are defined to be:

Variable	Number of Levels
1 A	3
2 B	4
3 C	4

```

-----
Strata 1: B = 1
C (row) by A (column)
  1      2      3
1 23.00  7.00  2.00
2 23.00 10.00  5.00
3 20.00 13.00  5.00
4 24.00 10.00  6.00

```

```

-----
Strata 2: B = 2
C (row) by A (column)
  1      2      3
1 18.00  6.00  1.00
2 18.00  6.00  2.00
3 13.00 13.00  2.00
4  9.00 15.00  2.00

```

```

-----
Strata 3: B = 3
C (row) by A (column)
  1      2      3
1  8.00  6.00  3.00
2 12.00  4.00  4.00
3 11.00  6.00  2.00
4  7.00  7.00  4.00

```

```

-----
Strata 4: B = 4
C (row) by A (column)
  1      2      3
1 12.00  9.00  1.00
2 15.00  3.00  2.00
3 14.00  8.00  3.00
4 13.00  6.00  4.00

```

Test of independence between row and column variables

Strata	Chi-Squared	Degrees of Freedom	Probability
1	3.4	6	0.7575
2	10.8	6	0.0942

3	3.1	6	0.7987
4	5.2	6	0.5177

	Chi-Squared	Degrees of Freedom	Probability
Mantel-Haenszel	10.6	6	0.1016

Test of equality of location for rows given column scores

Strata	Chi-Squared	Degrees of Freedom	Probability
1	2.62	3	0.4536
2	7.34	3	0.0617
3	1.69	3	0.6381
4	1.68	3	0.6420

	Chi-Squared	Degrees of Freedom	Probability
Mantel-Haenszel	6.59	3	0.08618

Row Scores

1	2	3
1.000	2.000	3.000

Test of correlation given row and column scores

Strata	Chi-Squared	Degrees of Freedom	Probability
1	1.57	1	0.2105
2	7.06	1	0.0079
3	0.16	1	0.6927
4	0.66	1	0.4161

	Chi-Squared	Degrees of Freedom	Probability
Mantel-Haenszel	6.34	1	0.0118

Row Scores

1	2	3
1.000	2.000	3.000

Column Scores

1	2	3	4
1.000	2.000	3.000	4.000

Comments

1. Workspace may be explicitly provided, if desired, by use of C2RAN/DC2RAN. The reference is:

```
CALL C2RAN (NCLVAR, NCLVAL, TABLE, INDROW, INDCOL, ITYPE,
IROWSC, ICOLSC, IPRINT, ROWSCR, COLSCR, STAT, LDSTAT, IX, F,
COLSUM, ROWSUM, DIFVEC, DIFSUM, COV, COVSUM, AWK, BWK)
```

The additional arguments are as follows:

IX — Work array of length NCLVAR.

F — Work array of length $NCLVAL(INDROW) * NCLVAL(INDCOL)$.

COLSUM — Work array of length $NCLVAL(INDCOL)$.

ROWSUM — Work array of length $NCLVAL(INDROW)$.

DIFVEC — Work array. If $ITYPE = 1$, the length is $(NCLVAL(INDROW) - 1) * (NCLVAL(INDCOL) - 1)$. For $ITYPE = 2$, the length is $NCLVAL(INDROW)$. For $ITYPE = 3$, **DIFVEC** is not used and may be of length 1.

DIFSUM — Work array. If $ITYPE = 1$, the length is $(NCLVAL(INDROW) - 1) * (NCLVAL(INDCOL) - 1)$. **DIFSUM** contains the sum of the tables containing the observed minus expected frequencies (excluding the last row and column of each table). For $ITYPE = 2$, the length is $NCLVAL(INDROW)$. **DIFSUM** contains the sum of the table row mean scores minus their expected value. For $ITYPE = 3$, the length is 1. **DIFSUM** contains the sum of the table correlations between the row and column mean scores. (Output)

COV — Work array. If $ITYPE = 1$, the length is $(NCLVAL(INDROW) - 1)^2 * (INCLVA(INDCOL) - 1)^2$. For $ITYPE = 2$, the length is $NCLVAL(INDROW)^2$. For $ITYPE = 3$, **COV** is not used and may be of length 1.

COVSUM — Work array. If $ITYPE = 1$, the length is $(NCLVAL(INDROW) - 1)^2 * (INCLVA(INDCOL) - 1)^2$. For $ITYPE = 2$, the length is $NCLVAL(INDROW)^2$. For $ITYPE = 3$, the length is 1.

AWK — Work array. If $ITYPE = 1$, the length is $(NCLVAL(INDROW) - 1)^2$. For $ITYPE = 2$, the length is $NCLVAL(INDROW)$. For $ITYPE = 3$, **AWK** is not used and may be of length 1.

BWK — Work array. If $ITYPE = 1$, the length is $(NCLVAL(INDCOL) - 1)^2$. For $ITYPE = 2$ or 3, **BWK** is not used and may be of length 1.

2. Informational errors

Type	Code	
3	1	All frequencies of stratified table κ are zero. This table will be excluded from the Mantel-Haenszel test statistic.
3	2	The elements of stratified table κ sum to one. This table will be excluded from the Mantel-Haenszel test statistic.
3	3	The variance of the response variable for stratified table κ is zero.
3	4	The variance of either the sub-population or the response variable is zero for stratified table κ .
3	5	The label for table κ exceeds the buffer limit of 72.

Here, κ is an integer that is greater than or equal to one and less than or equal to the number of stratified contingency tables.

3. The cells of the vectors `TABLE` are sequenced so that the first variable cycles from 1 to `NCLVAL(1)` the slowest, the second variable cycles from 1 to `NCLVAL(2)` the next most slowly, and so on, up to the `NCLVAR`-th variable, which cycles from 1 to `NCLVAL(NCLVAR)` the fastest.

Example: For `NCLVAR = 3`, `NCLVAL(1) = 2`, `NCLVAL(2) = 3`, and `NCLVAL(3) = 2` the cells of table `X(I, J, K)` are entered into `TABLE(1)` through `TABLE(12)` in the following order:

`X(1, 1, 1)`, `X(1, 1, 2)`, `X(1, 2, 1)`, `X(1, 2, 2)`, `X(1, 3, 1)`, `X(1, 3, 2)`, `X(2, 1, 1)`, `X(2, 1, 2)`,
`X(2, 2, 1)`, `X(2, 2, 2)`, `X(2, 3, 1)`, `X(2, 3, 2)`.

Description

Routine `CTRAN` computes tests of partial association (a test of homogeneity, a test on means, and a test on correlations) in stratified two-dimensional contingency tables. The type of test computed depends upon parameter `ITYPE`. All tests are generalizations of the Mantel-Haenszel stratified 2×2 contingency table test statistic in the sense that information is “pooled” over all tables without increasing the total degrees of freedom in the test. Like the Mantel-Haenszel test, if all tables violate the null hypothesis in the same direction, the tests computed here are more powerful than most other tests of the same null hypothesis.

While `CTRAN` allows for an arbitrary number of classification variables, only three are required to describe the test statistics since all stratification variables could be (if desired) lumped into a single classification variable. Because of this, only three classification variables are discussed here. Let f_{ijk} denote the frequency in cell ij of stratum k where $k = 1, \dots, m$, $i = 1, \dots, r$, and $j = 1, \dots, c$. Then, the input data can be described as a series of contingency tables. For example, if $r = c = 2$, so that 2×2 tables are used, then we would have:

f_{111}	f_{121}		f_{112}	f_{122}	...	f_{11m}	f_{12m}
f_{211}	f_{221}		f_{212}	f_{222}		f_{21m}	f_{22m}

All tests are computed as follows: For each table, a test statistic vector x_k with estimated covariance matrix

$$\hat{\Sigma}_k$$

is computed. The test statistic vector x_k represents the mean difference (from the null hypothesis) for the test being computed. Thus, if `ITYPE = 1`, x_k is a vector of cell frequencies minus their expected value under the hypothesis of homogeneity while if `ITYPE = 2`, x_k is a vector containing the row means (based upon the row scores) for the elements in a row of a table minus the estimated mean for the table (estimated under the assumption that all means are equal). Finally, if `ITYPE = 3`, x_k is a vector of length 1 containing an estimated correlation coefficient computed between the row and column scores.

Note that for nominal data in both the rows and columns, one would generally use `ITYPE = 1` while if an ordering (and scores) make sense for each row of a table, `ITYPE = 2` would be used. If an ordering (and scores) make sense for both the rows and the columns of a table, then a correlation measure (`ITYPE = 3`) is appropriate.

Test statistics for each table are computed as

$$\chi_k^2 = x_k^T \hat{\Sigma}_k^{-1} x_k$$

which has degrees of freedom $(r-1)(c-1)$ when `ITYPE = 1`, $r-1$ when `ITYPE = 2`, and 1 for `ITYPE = 3`. While these test statistics could be combined by summing them over all tables (yielding a χ^2 test with m times the degrees of freedom in a single table), the Mantel-Haenszel test combines the scores in a different way. Let

$$x = \sum_k x_k, \text{ and let } \hat{\Sigma} = \sum_k \hat{\Sigma}_k$$

Then, an overall χ^2 may be computed as

$$x^T \hat{\Sigma}^{-1} x$$

This test statistic has the same degrees of freedom as the test statistic computed for a single stratum of the three-way table and is reported in the last row of `STAT`. Routine `CTRAN` uses simplified computational methods. See Landis, Cooper, Kennedy, and Koch (1979) for details.

Landis, Cooper, Kennedy, and Koch (1979, page 225) give the null hypothesis for a test of partial association as follows (paraphrased):

H_0 : For each of the separate tables, the data in the respective rows of the table can be regarded as a successive set of simple random samples from a fixed population corresponding to the column marginal totals for the table.

All three tests above are tests of partial association.

For `ITYPE = 2` and 3, different row and column (`ITYPE = 3`) scores are used in computing measures of location and association. The scores used by `CTRAN` for the rows are

1. For `IROWSC = 0`, the user supplies the scores to be used in each row of the table.
2. For `IROWSC = 1`, uniform scores are used. These scores consist of the digits 1, 2, ..., c where c is the number of columns in each table.
3. For `IROWSC = 2`, combined ridit scores are used. A combined ridit score is computed by summing the column marginals over all tables. The combined row score for the j -th column is then computed as the sum of the initial $j-1$ column marginals plus half of the j -th column marginal. The result is divided by the number of observations in all tables to yield the ridit score.
4. For `IROWSC = 3`, marginal rank scores are used. The j -th marginal rank score is computed for each table from the column marginals for that table as the sum of the initial $j-1$ column marginals plus half the j -th column marginal.
5. For `IROWSC = 4`, marginal ridit scores are used. These are computed as the marginal rank scores divided by the total frequency in the table.
6. For `IROWSC = 5`, logrank scores are used. These are computed as

$$c_{jk} = 1 - \sum_{l=1}^j \frac{f_{+lk}}{\sum_{i=1}^c f_{+ik}}$$

where f_{+lk} is the column marginal for column l in table k .

Column scores are computed in a similar manner.

CTGLM

Analyzes categorical data using logistic, Probit, Poisson, and other generalized linear models.

Required Arguments

X — NOBS by NCOL matrix containing the data. (Input)

MODEL — Model option parameter. (Input)

MODEL specifies the distribution of the response variable and the function used to model the distribution parameter. The lower-bound given in the following table is the minimum possible value of the response variable:

MODEL	Distribution	Function	Lower-bound
0	Poisson	Exponential	0
1	Neg. Binomial	Logistic	0
2	Logarithmic	Logistic	1
3	Binomial	Logistic	0
4	Binomial	Probit	0
5	Binomial	Log-log	0

Let γ be the dot product of a row in the design matrix with the parameters (plus the fixed parameter, if used). Then, the functions used to model the distribution parameter are given by:

Name	Function
Exponential	$\exp(\gamma)$
Logistic	$\exp(\gamma)/(1 + \exp(\gamma))$
Probit	Normal(γ) (normal cdf)
Log-log	$1 - \exp(-\gamma)$

NCOEF — Number of estimated coefficients in the model. (Output)

COEF — NCOEF by 4 matrix containing the parameter estimates and associated statistics. (Output, if INIT = 0; input, if INIT = 1 and MAXIT = 0; input/output, if INIT = 1 and MAXIT > 0)

Col. Statistic

- 1 Coefficient estimate.
- 2 Estimated standard deviation of the estimated coefficient.
- 3 Asymptotic normal score for testing that the coefficient is zero.
- 4 p -value associated with the normal score in column 3.

When `INIT = 1`, only the first column needs to be specified on input.

COV — `NCOEF` by `NCOEF` matrix containing the estimated asymptotic covariance matrix of the coefficients. (Output)

For `MAXIT = 0`, this is the Hessian computed at the initial parameter estimates.

XMEAN — Vector of length `NCOEF` containing the means of the design variables. (Output)

GR — Vector of length `NCOEF` containing the last parameter updates (excluding step halvings). (Output)

For `MAXIT = 0`, `GR` contains the inverse of the Hessian times the gradient vector, all computed at the initial parameter estimates.

Optional Arguments

NOBS — Number of observations. (Input)

Default: `NOBS = size(x,1)`.

NCOL — Number of columns in `x`. (Input)

Default: `NCOL = size(x,2)`.

LDX — Leading dimension of `x` exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDX = size(x,1)`.

ILT — For full-interval and left-interval observations, the column number in `x` that contains the upper endpoint of the observation interval. (Input)

See argument `ICEN`. If `ILT = 0`, left-interval and full-interval observations cannot be input.

Default: `ILT = 0`.

IRT — For full-interval and right-interval observations, the column number in `x` that contains the lower endpoint of the interval. (Input)

For point observations, `x(i, IRT)` contains the observation point. `IRT` must not be zero. See argument `ICEN`. In the usual case, all observations are “point” observations (see argument `ICEN`).

Default: `IRT = size(x,2)`.

IFRQ— Column number in X containing the frequency of response for each observation. (Input)
 If $IFRQ = 0$, a response frequency of 1 for each observation is assumed.
 Default: $IFRQ = 0$.

IFIX— Column number in X containing a fixed parameter for each observation that is added to the linear response prior to computing the model parameter. (Input)
 The “fixed” parameter allows one to test hypothesis about the parameters via the log-likelihoods. If $IFIX = 0$, the fixed parameter is assumed to be 0.
 Default: $IFIX = 0$.

IPAR— Column number in X containing an optional distribution parameter for each observation. (Input)
 Default: $IPAR = 0$.
 If $IPAR = 0$, the distribution parameter is assumed to be 1. The meaning of the distributional parameter depends upon $MODEL$ as follows:

MODEL	Meaning of $x(i, IPAR)$
0	The Poisson parameter is given by $x(i, IPAR) * \exp(\gamma)$.
1	The number of successes required in the negative binomial is given by $x(i, IPAR)$.
2	$x(i, IPAR)$ is not used.
3–5	The number of trials in the binomial distribution is given by $x(i, IPAR)$.

ICEN— Column number in X containing the interval-type for each observation. (Input)
 Default: $ICEN = 0$.
 If $ICEN = 0$, a code of 0 is assumed. Valid codes are

$x(i, ICEN)$ Censoring	
0	Point observation. The response is unique and is given by $x(i, IRT)$.
1	Right-interval. The response is greater than or equal to $x(i, IRT)$ and less than or equal to the upper bound, if any, of the distribution.
2	Left-interval. The response is less than or equal to $x(i, ILT)$ and greater than or equal to the lower bound of the distribution.
3	Full-interval. The response is greater than or equal to $x(i, IRT)$, but less than or equal to $x(i, ILT)$.

INFIN— Method to be used for handling infinite estimates. (Input)
 Default: $INFIN = 1$.

INFIN Method

- 0 Remove a right or left-censored observation from the log-likelihood whenever the probability of the observation exceeds 0.995. At convergence, use linear programming to check that all removed observations actually have an estimated linear response that is infinite. Set `IADD(i)` for observation *i* to 2 if the linear response is infinite. If not all removed observations have infinite linear response, recompute the estimates based upon the observations with estimated linear response that is finite. This option is valid only for censoring codes (see `ICEN`) 1 and 2.
- 1 Iterate without checking for infinite estimates.

See the “Description” section for more discussion.

MAXIT — Maximum number of iterations. (Input)

`MAXIT = 30` is usually sufficient. Use `MAXIT = 0` to compute the Hessian, stored in `COV`, and the Newton step, stored in `GR`, at the initial estimates.
Default: `MAXIT = 30`.

EPS — Convergence criterion. (Input)

Convergence is assumed when the maximum relative change in any coefficient estimate is less than `EPS` from one iteration to the next or when the relative change in the log-likelihood, `ALGL`, from one iteration to the next is less than `EPS/100`. If `EPS` is negative, `EPS = 0.001` is assumed.
Default: `EPS = .001`.

INTCEP — Intercept option. (Input)

Default: `INTCEP = 1`.

INTCEP Action

- 0 No intercept is in the model (unless otherwise provided for by the user).
- 1 Intercept is automatically included in the model.

NCLVAR — Number of classification variables. (Input)

Dummy or indicator variables are generated for classification variables using the `IDUMMY = 2` option of IMSL routine `GRGLM` (see Chapter 2, Regression). See Comment 3.
Default: `NCLVAR = 0`.

INDCL — Index vector of length `NCLVAR` containing the column numbers of `X` that are classification variables. (Input, if `NCLVAR` is positive; not used otherwise)
If `NCLVAR` is 0, `INDCL` is not referenced and can be dimensioned of length 1 in the calling program.

NEF — Number of effects in the model. (Input)
In addition to effects involving classification variables, simple covariates and the product of simple covariates are also considered effects.
Default: $NEF = 0$.

NVEF — Vector of length NEF containing the number of variables associated with each effect in the model. (Input, if NEF is positive; not used otherwise)
If NEF is zero, $NVEF$ is not used and can be dimensioned of length 1 in the calling program.

INDEF — Index vector of length $NVEF(1) + NVEF(2) + \dots + NVEF(NEF)$ containing the column numbers in X associated with each effect. (Input, if NEF is positive, not used otherwise) The first $NVEF(1)$ elements of $INDEF$ give the column numbers of the variables in the first effect. The next $NVEF(2)$ elements give the column numbers for the second effect, etc. If NEF is zero, $INDEF$ is not used and can be dimensioned of length 1 in the calling program.

INIT — Initialization option. (Input)
Default: $INIT = 0$.

INIT Action

- 0 Unweighted linear regression is used to obtain initial estimates.
- 1 The $NCOEF$ elements in the first column of $COEF$ contain initial estimates of the parameters on input to $SVGLM$ (requiring that the user know $NCOEF$ prior to calling $SVGLM$).

IPRINT — Printing option. (Input)
Default: $IPRINT = 0$.

IPRINT Action

- 0 No printing is performed.
- 1 Printing is performed, but observational statistics are not printed.
- 2 All output statistics are printed.

MAXCL — An upper bound on the sum of the number distinct values taken on by each classification variable. (Input)
Default: $MAXCL = 1$.

NCLVAL — Vector of length $NCLVAR$ containing the number of values taken by each classification variable. (Output, if $NCLVAR$ is positive; not used otherwise) $NCLVAL(i)$ is the number of distinct values for the i -th classification variable. If $NCLVAR$ is zero, $NCLVAL$ is not used and can be dimensioned of length 1 in the calling program.

CLVAL — Vector of length $NCLVAL(1) + NCLVAL(2) + \dots + NCLVAL(NCLVAR)$ containing the distinct values of the classification variables in ascending order. (Output, if *NCLVAR* is positive; not used otherwise)

Since in general the length of *CLVAL* will not be known in advance, *MAXCL* (an upper bound for this length) should be used for purposes of dimensioning *CLVAL*. The first *NCLVAL(1)* elements of *CLVAL* contain the values for the first classification variables, the next *NCLVAL(2)* elements contain the values for the second classification variable, etc. If *NCLVAR* is zero, then *CLVAL* is not referenced and can be dimensioned of length 1 in the calling program.

LDCOEF — Leading dimension of *COEF* exactly as specified in the dimension statement in the calling program. (Input)

Default: $LDCOEF = \text{size}(\text{COEF}, 1)$.

ALGL — Optimized criterion. (Output)

The criterion to be maximized is a constant plus the log-likelihood.

LDCOV — Leading dimension of *COV* exactly as specified in the dimension statement in the calling program. (Input)

Default: $LDCOV = \text{size}(\text{COV}, 1)$.

CASE — *NOBS* by 5 vector containing the case analysis. (Output)

Col. Statistic

- 1 Predicted parameter.
- 2 The residual.
- 3 The estimated standard error of the residual.
- 4 The estimated influence of the observation.
- 5 The standardized residual.

Case statistics are computed for all observations except where missing values prevent their computation.

The predicted parameter in column 1 depends upon *MODEL* as follows.

MODEL Parameter

- 0 The predicted mean for the observation
- 1–5 The probability of a success on a single trial

LDCASE — Leading dimension of *CASE* exactly as specified in the dimension statement in the calling program. (Input)

IADDS — Vector of length `NOBS` indicating which observations are included in the extended likelihood. (Output, if `MAXIT > 0`; input/output, if `MAXIT = 0`)

Value	Status of observation
0	Observation <i>i</i> is in the likelihood.
1	Observation <i>i</i> cannot be in the likelihood because it contains at least one missing value in <i>x</i> .
2	Observation <i>i</i> is not in the likelihood. Its estimated parameter is infinite. For <code>MAXIT = 0</code> , the <code>IADDS</code> array must be initialized prior to calling <code>CTGLM</code> .

In this case, some elements of `IADDS` may be set to 1, by `CTGLM`, but no check for infinite estimates performed.

NRMIS — Number of rows of data in *x* that contain missing values in one or more columns `ILT`, `IRT`, `IFRQ`, `IFIX`, `IPAR`, `ICEN`, `INDCL`, or `INDEF` of *x*. (Output)

FORTRAN 90 Interface

Generic: `CALL CTGLM (X, MODEL, NCOEF, COEF, COV, XMEAN, GR [, ...])`

Specific: The specific interface names are `S_CTGLM` and `D_CTGLM`.

FORTRAN 77 Interface

Single: `CALL CTGLM (NOBS, NCOL, X, LDX, MODEL, ILT, IRT, IFRQ, IFIX, IPAR, ICEN, INFIN, MAXIT, EPS, INTCEP, NCLVAR, INDCL, NEF, NVEF, INDEF, INIT, IPRINT, MAXCL, NCLVAL, CLVAL, NCOEF, COEF, LDCOEF, ALGL, COV, LDCOV, XMEAN, CASE, LDCASE, GR, IADDS, NRMIS)`

Double: The double precision name is `DCTGLM`.

Example

The first example is from Prentice (1976) and involves the mortality of beetles after exposure to various concentrations of carbon disulphide. Both a logit and a probit fit are produced for linear model

$$\mu + \beta x$$

The data is given as:

Covariate(x)	N	y
1.690	59	6
1.724	60	13

Covariate(x)	N	y
1.755	62	18
1.784	56	28
1.811	63	52
1.836	59	53
1.861	62	61
1.883	60	60

```

USE UMACH_INT
USE CTGLM_INT

INTEGER      IPAR, IRT, LDCASE, LDCOEF, LDCOV, LDX, NCOL, &
             NEF, NOBS
REAL        EPS
PARAMETER   (EPS=0.0001, IPAR=2, IRT=3, LDCASE=8, LDCOEF=2, LDCOV=2, &
             LDX=8, MAXCL=1, NCOL=3, NEF=1, NOBS=8)
!
INTEGER      INDCL(MAXCL), INDEF(1), IPRINT, MODEL, NCLVAL(1), &
             NCOEF, NRMISS, NVEF(1)
REAL        ALGL, CASE(LDCASE,5), CLVAL(1), COEF(LDCOEF,4), &
             COV(LDCOV,4), GR(2), X(LDX,NCOL), XMEAN(2)
!
DATA NVEF/1/, INDEF/1/
DATA X/1.690, 1.724, 1.755, 1.784, 1.811, 1.836, 1.861, 1.883, &
      59, 60, 62, 56, 63, 59, 62, 60, 6, 13, 18, 28, 52, 53, 61, &
      60/
!
IPRINT = 2
CALL UMACH(2, NOUT)
DO 10 MODEL=3, 4
  WRITE(NOUT, *) `MODEL=' , MODEL
  CALL CTGLM(X, MODEL, NCOEF, COEF, COV, XMEAN, GR, IRT=IRT, &
            IPAR=IPAR, EPS=EPS, NEF=NEF, NVEF=NVEF, &
            INDEF=INDEF, IPRINT=IPRINT)
  IPRINT = 1
10 CONTINUE
!
END

```

Output

Model = 3

Initial Estimates

1	2
-63.27	35.84

Method	Iteration	Step size	Maximum scaled coef. update	Log likelihood
Q-N	0			-20.31
Q-N	1	1.0000	0.1387	-19.25
N-R	2	1.0000	0.6112E-01	-18.89
N-R	3	1.0000	0.7221E-01	-18.78
N-R	4	1.0000	0.6362E-03	-18.78

N-R 5 1.0000 0.3044E-06 -18.78

Log-likelihood -18.77818

Coefficient Statistics				
	Coefficient	Standard Error	Asymptotic Z-statistic	Asymptotic P-value
1	-60.76	5.19	-11.66	0.00
2	34.30	2.92	11.76	0.00

Asymptotic Coefficient Covariance

	1	2
1	0.2691E+02	-0.1512E+02
2		0.8505E+01

Case Analysis						
	Predicted	Residual	Std. Residual	Leverage	Standardized Residual	
1	0.058	2.593	1.792	0.267	1.448	
2	0.164	3.139	2.871	0.347	1.093	
3	0.363	-4.498	3.786	0.311	-1.188	
4	0.606	-5.952	3.656	0.232	-1.628	
5	0.795	1.890	3.202	0.269	0.590	
6	0.902	-0.195	2.288	0.238	-0.085	
7	0.956	1.743	1.619	0.198	1.077	
8	0.979	1.278	1.119	0.138	1.143	

Last Coefficient Update

	1	2
	1.104E-07	-2.295E-07

Covariate Means

1.793

Observation Codes

1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0

Number of Missing Values 0

Model = 4

Log-likelihood -18.23232

Coefficient Statistics				
	Coefficient	Standard Error	Asymptotic Z-statistic	Asymptotic P-value
1	-34.94	2.64	-13.23	0.00
2	19.74	1.49	13.29	0.00

Note that the probit model yields a slightly smaller absolute log-likelihood and, thus, is preferred. For this data, a model based upon the log-log transformation function is even better. See Prentice (1976) for details.

Comments

1. Workspace may be explicitly provided, if desired, by use of C2GLM/DC2GLM. The reference is:

```
CALL C2GLM (NOBS, NCOL, X, LDX, MODEL, ILT, IRT, IFRQ, IFIX,
IPAR, ICEN, INFIN, MAXIT, EPS, INTCEP, NCLVAR, INDCL, NEF, NVEF,
INDEF, INIT, IPRINT, MAXCL, NCLVAL, CLVAL, NCOEF, COEF, LDCOEF,
ALGL, COV, LDCOV, XMEAN, CASE, LDCASE, GR, IADD, NRMISS, NMAX,
OBS, ADDX, XD, WK, KBASIS)
```

The additional arguments are as follows:

NMAX — Maximum number of observations that can be handled in the linear programming. (Input)

If workspace is not explicitly provided, NMAX is set to

$NMAX = (n - 8)/(7 + NCOEF)$ in S_CTGLM and $NMAX = (n - 16)/(11 + 2 * NCOEF)$ in D_CTGLM where n is the maximum number of units of workspace available after allocating space for OBS. In the typical problem, no linear programming is performed so that $NMAX = 1$ is sufficient. $NMAX = NOBS$ is always sufficient. Even when extended maximum likelihood estimates are computed, $NMAX = 30$ will usually suffice. If $INFIN = 1$, set $NMAX = 0$.

OBS — Work vector of length $NCOEF + 1$.

ADDX — Logical work vector of length NMAX. ADDX is not needed and can be a array of length 1 in the calling program if $NMAX = 0$.

XD — Work vector of length $NMAX * NCOEF$. XD is not needed and can be a array of length 1 in the calling program if $NMAX = 0$.

WK — Work vector of length $4 * NMAX$. WK is not needed and can be a array of length 1 in the calling program if $NMAX = 0$.

KBASIS — Work vector of length $2 * NMAX$. KBASIS is not needed and can be a array of length 1 in the calling program if $NMAX = 0$.

2. Informational errors

Type	Code	
3	1	There were too many iterations required. Convergence is assumed.
3	2	There were too many step halvings. Convergence is assumed.
4	3	The number of distinct values of the classification variables exceeds MAXCL.
4	4	The number of distinct values of a classification must be greater than one.
4	5	LDCOEF or LDCOV must be greater than or equal to NCOEF.
4	6	The number of observations to be deleted has exceeded NMAX. Rerun with a different model or increase the workspace.

3. Dummy variables are generated for the classification variables as follows: An ascending list of all distinct values of each classification variable is obtained and stored in `CLVAL`. Dummy variables are then generated for each but the last of these distinct values. Each dummy variable is zero unless the classification variable equals the list value corresponding to the dummy variable, in which case the dummy variable is one. See argument `IDUMMY` for `IDUMMY = 2` in routine `GRGLM` in Chapter 2.
4. The “product” of a classification variable with a covariate yields dummy variables equal to the product of the covariate with each of the dummy variables associated with the classification variable.
5. The “product” of two classification variables yields dummy variables in the usual manner. Each dummy variable associated with the first classification variable multiplies each dummy variable associated with the second classification variable. The resulting dummy variables are such that the index of the second classification variable varies fastest.

Description

Routine `CTGLM` uses iteratively reweighted least squares to compute (extended) maximum likelihood estimates in some generalized linear models involving categorized data. One of several models, including the probit, logistic, Poisson, logarithmic, and negative binomial models, may be fit for input point or interval observations. (In the usual case, only point observations are observed.)

Let

$$\gamma_i = w_i + x_i^T \beta = w_i + \eta_i$$

be the linear response where x_i is a design column vector obtained from a row of X , β is the column vector of coefficients to be estimated, and w_i is a fixed parameter that may be input in `x`. When some of the γ_i are infinite at the supremum of the likelihood, then *extended maximum likelihood estimates* are computed. Extended maximum likelihood are computed as the finite (but nonunique) estimates $\hat{\beta}$ that optimize the likelihood containing only the observations with finite $\hat{\gamma}_i$. These estimates, when combined with the set of indices of the observations such that $\hat{\gamma}_i$ is infinite at the supremum of the likelihood, are called extended maximum estimates. When none of the optimal $\hat{\gamma}_i$ are infinite, extended maximum likelihood estimates are identical to maximum likelihood estimates. Extended maximum likelihood estimation is discussed in more detail by Clarkson and Jennrich (1991). In `CTGLM`, observations with potentially infinite

$$\hat{\eta}_i = x_i^T \hat{\beta}$$

are detected and removed from the likelihood if `INFIN = 0`. See below.

The models available in CTGLM are :

MODEL	Name	Parameterization	PDF
0	Poisson	$\lambda = N * \exp(w + \eta)$	$f(y) = \lambda^y \exp(-\lambda) / y!$
1	Neg. Bin.	$\theta = \frac{\exp\{w + \eta\}}{1 + \exp\{w + \eta\}}$	$f(y) = \binom{S + y - 1}{y - 1} \theta^S (1 - \theta)^y$
2	Logarith.	$\theta = \frac{\exp\{w + \eta\}}{1 + \exp\{w + \eta\}}$	$f(y) = (1 - \theta)^y / (y \ln \theta)$
3	Logistic	$\theta = \frac{\exp\{w + \eta\}}{1 + \exp\{w + \eta\}}$	$f(y) = \binom{N}{y} \theta^y (1 - \theta)^{N - y}$
4	Probit	$\theta = \Phi(w + \eta)$	$f(y) = \binom{N}{y} \theta^y (1 - \theta)^{N - y}$
5	Log-log	$\theta = 1 - \exp\{-\exp(w + \eta)\}$	$f(y) = \binom{N}{y} \theta^y (1 - \theta)^{N - y}$

Here, Φ denotes the cumulative normal distribution, N and S are known parameters specified for each observation via column `IPAR` of `X`, and w is an optional fixed parameter specified for each observation via column `IFIX` of `X`. (If `IPAR` = 0, then N is taken to be 1 for `MODEL` = 0, 3, 4 and 5 and S is taken to be 1 for `MODEL` = 1. If `IFIX` = 0, then w is taken to be 0.) Since the log-log model (`MODEL` = 5) probabilities are not symmetric with respect to 0.5, quantitatively, as well as qualitatively, different models result when the definitions of “success” and “failure” are interchanged in this distribution. In this model and all other models involving θ , θ is taken to be the probability of a “success.”

Note that each row vector in the data matrix can represent a single observation; or, through the use of column `IFRQ` of the matrix `X`, each vector can represent several observations. Also note that classification variables and their products are easily incorporated into the models via the usual regression-type specifications.

Computational Details

For interval observations, the probability of the observation is computed by summing the probability distribution function over the range of values in the observation interval. For right-interval observations, $\Pr(Y \geq y)$ is computed as a sum based upon the equality $\Pr(Y \geq y) = 1 - \Pr(Y < y)$. Derivatives are computed similarly. CTGLM allows three types of interval observations. In full interval observations, both the lower and the upper endpoints of the interval must be specified. For right-interval observations, only the lower endpoint need be given while for left-interval observations, only the upper endpoint is given.

The computations proceed as follows:

1. The input parameters are checked for consistency and validity.
2. Estimates of the means of the “independent” or design variables are computed. The frequency of the observation in all but binomial distribution models is taken from

column `IFRQ` of the data matrix X . In binomial distribution models, the frequency is taken as the product of $n = X(I, \text{IPAR})$ and $X(I, \text{IFRQ})$. In all cases, if `IFRQ` = 0, or `IPAR` = 0, these values default to 1. Means are computed as

$$\bar{x} = \frac{\sum_i f_i x_i}{\sum_i f_i}$$

3. If `INIT` = 0, initial estimates of the coefficients are obtained (based upon the observation intervals) as multiple regression estimates relating transformed observation probabilities to the observation design vector. For example, in the binomial distribution models, θ for point observations may be estimated as

$$\hat{\theta} = X(I, \text{IRT}) / X(I, \text{IPAR})$$

and, when `MODEL` = 3, the linear relationship is given by

$$\left(\ln \left(\hat{\theta} / (1 - \hat{\theta}) \right) \approx X\beta \right)$$

while if `MODEL` = 4,

$$\left(\Phi^{-1}(\hat{\theta}) = X\beta \right)$$

For bounded interval observations, the midpoint of the interval is used for $X(I, \text{IRT})$. Right-interval observations are not used in obtaining initial estimates when the distribution has unbounded support (since the midpoint of the interval is not defined). When computing initial estimates, standard modifications are made to prevent illegal operations such as division by zero.

Regression estimates are obtained at this point, as well as later, by use of routine `RGIVN` (see Chapter 2, Regression).

4. Newton-Raphson iteration for the maximum likelihood estimates is implemented via iteratively reweighted least squares. Let

$$\Psi(x_i^T \beta)$$

denote the log of the probability of the i -th observation for coefficients β . In the least-squares model, the weight of the i -th observation is taken as the absolute value of the second derivative of

$$\Psi(x_i^T \beta)$$

with respect to

$$\gamma_i = x_i^T \beta$$

(times the frequency of the observation), and the dependent variable is taken as the first derivative Ψ with respect to γ_i , divided by the square root of the weight times the frequency. The Newton step is given by

$$\Delta\beta = \left(\sum_i |\Psi''(\gamma_i)| x_i x_i^T \right)^{-1} \sum_i \Psi'(\gamma_i) x_i$$

where all derivatives are evaluated at the current estimate of γ , and

$\beta_{n+1} = \beta_n - \Delta\beta$. This step is computed as the estimated regression coefficients in the least-squares model. Step halving is used when necessary to ensure a decrease in the criterion.

5. Convergence is assumed when the maximum relative change in any coefficient update from one iteration to the next is less than `EPS` or when the relative change in the log-likelihood from one iteration to the next is less than `EPS/100`. Convergence is also assumed after `MAXIT` iterations or when step halving leads to a step size of less than .0001 with no increase in the log-likelihood.
6. For interval observations, the contribution to the log-likelihood is the log of the sum of the probabilities of each possible outcome in the interval. Because the distributions are discrete, the sum may involve many terms. The user should be aware that data with wide intervals can lead to expensive (in terms of computer time) computations.
7. If requested (`INFIN = 0`), then the methods of Clarkson and Jennrich (1991) are used to check for the existence of infinite estimates in

$$\eta_i = x_i^T \beta$$

As an example of a situation in which infinite estimates can occur, suppose that observation j is right censored with $t_j > 15$ in a logistic model. If design matrix X is such that $x_{jm} = 1$ and $x_{im} = 0$ for all $i \neq j$, then the optimal estimate of β_m occurs at

$$\hat{\beta}_m = \infty$$

leading to an infinite estimate of both β_m and η_j . In `CTGLM`, such estimates may be “computed.”

In all models fit by `CTGLM`, infinite estimates can only occur when the optimal estimated probability associated with the left- or right-censored observation is 1. If `INFIN = 0`, left- or right- censored observations that have estimated probability greater than 0.995 at some point during the iterations are excluded from the log-likelihood, and the iterations proceed with a log-likelihood based upon the remaining observations. This allows convergence of the algorithm when the maximum relative change in the estimated coefficients is small and also allows for the determination of observations with infinite

$$\eta_i = x_i^T \beta$$

At convergence, linear programming is used to ensure that the eliminated observations have infinite η_i . If some (or all) of the removed observations should not have been removed (because their estimated η_i 's must be finite), then the iterations are restarted with a log-likelihood based upon the finite η_i observations. See Clarkson and Jennrich (1991) for more details.

When `INFIN = 1`, no observations are eliminated during the iterations. In this case, when infinite estimates occur, some (or all) of the coefficient estimates $\hat{\beta}$ will become large, and it is likely that the Hessian will become (numerically) singular prior to convergence.

When infinite estimates for the $\hat{\eta}_j$ are detected, routine `RGIVN` (see Chapter 2, Regression) is used at the convergence of the algorithm to obtain unique estimates $\hat{\beta}$. This is accomplished by regressing the optimal $\hat{\eta}_j$ or the observations with finite η against $X\beta$, yielding a unique $\hat{\beta}$ (by setting coefficients $\hat{\beta}$ that are linearly related to previous coefficients in the model to zero). All of the final statistics relating to $\hat{\beta}$ are based upon these estimates.

8. Residuals are computed according to methods discussed by Pregibon (1981). Let $l_i(\gamma_i)$ denote the log-likelihood of the i -th observation evaluated at γ_i . Then, the standardized residual is computed as

$$r_i = \frac{\ell'_i(\hat{\gamma}_i)}{\sqrt{\ell''_i(\hat{\gamma}_i)}}$$

where $\hat{\gamma}_i$ is the value of γ_i when evaluated at the optimal $\hat{\beta}$ and the derivatives here (and only here) are with respect to γ rather than with respect to β . The denominator of this expression is used as the “standard error of the residual” while the numerator is the “raw” residual.

Following Cook and Weisberg (1982), we take the influence of the i -th observation to be

$$\ell'_i(\hat{\gamma}_i)^T \ell''(\hat{\gamma})^{-1} \ell'(\hat{\gamma}_i)$$

This quantity is a one-step approximation to the change in the estimates when the i -th observation is deleted. Here, the partial derivatives are with respect to β .

Programming Notes

1. Classification variables are specified via arguments `NCLVAR` and `INDCL`. Indicator or dummy variables are created for the classification variables using routine `GRGLM` (see Chapter 2, Regression) with `IDUMMY = 2`.
2. To enhance precision “centering” of covariates is performed if `INTCEP = 1` and `NOBS - NRMISS > 1`. In doing so, the sample means of the design variables are subtracted from each observation prior to its inclusion in the model. On convergence the intercept, its variance and its covariance with the remaining estimates are transformed to the uncentered estimate values.
3. Two methods for specifying a binomial distribution model are possible. In the first method, `X(I, IFRQ)` contains the frequency of the observation while `X(I, IRT)` is 0 or 1

depending upon whether the observation is a success or failure. In this case, $N = x(I, I\text{PAR})$ is always 1. The model is treated as repeated Bernoulli trials, and interval observations are not possible.

A second method for specifying binomial models is to use $x(I, I\text{RT})$ to represent the number of successes in the $x(I, I\text{PAR})$ trials. In this case, $x(I, I\text{FRQ})$ will usually be 1, but it may be greater than 1, in which case interval observations are possible.

Additional Example

As a second example, the following data illustrate the Poisson model when all types of interval data are present. The example also illustrates the use of classification variables and the detection of potentially infinite estimates (which turn out here to be finite). These potential estimates lead to the two iteration summaries. The input data is

Column				
ILT	IRT	ICEN	Class 1	Class 2
0	5	0	1	0
9	4	3	0	0
0	4	1	0	0
9	0	2	1	1
0	1	0	0	1

A linear model

$$\mu + \beta_1 x_1 + \beta_2 x_2$$

is fit where $x_1 = 1$ if the Class 1 variable is 0, $x_1 = 0$, otherwise, and the x_2 variable is similarly defined.

```

USE CTGLM_INT
INTEGER ICEN, IFIX, ILT, INFIN, IPAR, IPRINT, IRT, &
LDCASE, LDcoef, LDCOV, LDX, MAXCL, MAXIT, MODEL, &
NCLVAR, NCOL, NEF, NOBS
REAL EPS
PARAMETER (ICEN=3, ILT=1, INFIN=0, IPAR=2, IPRINT=2, &
IRT=2, LDCASE=5, LDcoef=4, LDCOV=4, LDX=5, MAXCL=4, &
MODEL=0, NCLVAR=2, NCOL=5, NEF=2, NOBS=5)
!
INTEGER IADD(NOBS), INDCL(NCLVAR), INDEF(2), NCLVAL(MAXCL), &
NCOEF, NRMISS, NVEF(NEF)
REAL ALGL, CASE(LDCASE,5), CLVAL(4), COEF(LDcoef,4), &
COV(LDCOV,4), GR(5), X(LDX,NCOL), XMEAN(3)
!
DATA INDCL/4, 5/, NVEF/1, 1/, INDEF/4, 5/
DATA X/0, 9, 0, 9, 0, 5, 4, 4, 0, 1, 0, 3, 1, 2, 0, 1, 0, 0, 1, &
0, 0, 0, 0, 1, 1/
!
CALL CTGLM (X, MODEL, NCOEF, COEF, COV, XMEAN, GR, ILT=ILT, &
IRT=IRT, IPAR=IPAR, ICEN=ICEN, INFIN=INFIN, &
NCLVAR=NCLVAR, NCLVAL=NCLVAL, CLVAL=CLVAL, INDCL=INDCL, &

```

NEF=NEF, NVEF=NVEF, INDEF=INDEF, IPRINT=IPRINT, &
 MAXCL=MAXCL)

!

END

Output

Initial Estimates

	1	2	3
0.2469	0.4463	-0.0645	

Method	Iteration	Step size	Maximum scaled coef. update	Log likelihood
Q-N	0			-3.529
Q-N	1	0.2500	5.168	-3.262
N-R	2	0.0625	183.4	-3.134
N-R	3	1.0000	0.7438	-3.006
N-R	4	1.0000	0.2108	-3.005
N-R	5	1.0000	0.5559E-02	-3.005

Method	Iteration	Step size	Maximum scaled coef. update	Log likelihood
Q-N	0			-3.529
Q-N	1	0.2500	5.168	-3.262
N-R	2	0.0625	183.4	-3.217
N-R	3	1.0000	1.128	-3.116
N-R	4	1.0000	0.1673	-3.115
N-R	5	1.0000	0.4418E-02	-3.115

Log-likelihood -3.114638

Coefficient Statistics

	Coefficient	Standard Error	Asymptotic Z-statistic	Asymptotic P-value
1	-0.549	1.171	-0.517	0.605
2	0.549	0.610	0.900	0.368
3	0.549	1.083	0.507	0.612

Asymptotic Coefficient Covariance

	1	2	3
1	0.1372E+01	-0.3719E+00	-0.1172E+01
2		0.3719E+00	0.1719E+00
3			0.1172E+01

Case Analysis

	Predicted	Residual	Residual Std. Error	Leverage	Standardized Residual
1	5.000	0.000	2.236	1.000	0.000
2	6.925	-0.412	2.108	0.764	-0.196
3	6.925	0.412	1.173	0.236	0.351
4	0.000	0.000	0.000	0.000	NaN
5	1.000	0.000	1.000	1.000	0.000

Last Coefficient Update

	1	2	3

-2.924E-07 -1.131E-08 7.075E-07

Covariate Means

	1	2
0.6000	0.6000	

Distinct Values For Each Class Variable

Variable 1:	0.	1.0
Variable 2:	0.	1.0

Observation Codes

1	2	3	4	5
0	0	0	0	0

Number of Missing Values 0

CTWLS

Performs a generalized linear least-squares analysis of transformed probabilities in a two-dimensional contingency table.

Required Arguments

TABLE — NRESP by NPOP matrix containing the frequency count in each cell of each population. (Input)

The i -th column of TABLE contains the counts for the i -th population.

ITRAN — Vector of length NTRAN containing the transformation code for each of the NTRAN transformations to be applied. (Input)

ITRAN is not referenced and can be a vector of length 1 in the calling program if NTRAN = 0. Let a “response” denote a transformed cell probability. Then, ITRAN(1) contains the first transformation to be applied to the cell probabilities, ITRAN(2) contains the second transformation, which is to be applied to the responses obtained after ITRAN(1) is performed, etc. Note that the k -th transformation takes the ISIZE($k - 1$) responses at step k into ISIZE(k) responses, where ISIZE(0) is taken to be NPOP * NRESP. Let y denote the vector result of a transformation, x denote the responses before the transformation is applied, A denote a matrix of constants, and v denote a vector of constants. Then, the possible transformations are

ITRAN Transformation

- | | |
|---|---|
| 1 | Linear, defined over all populations ($y = Ax$) |
| 2 | Logarithmic ($y(i, j) = \ln(x(i, j))$) |
| 3 | Exponential ($y(i, j) = \exp(x(i, j))$) |
| 4 | Additive ($y(i, j) = y(i, j) + v(i, j)$) |

ITRAN Transformation

5 Linear, defined for one population and, identically, applied over all populations ($y(i) = Ax(i)$)

where $y(i)$ and $x(i)$ are the subvectors for the i -th population, $y(i, j)$ and $x(i, j)$ denote the j -th response in the i -th population, and $v(i, j)$ denotes the corresponding element of the vector “ v ”. Transformation type 5 is the same as transformation type 1 when the same linear transformation is applied in each population (i.e., the type 1 matrix is block diagonal with identical blocks). Because the size of the type 5 transformation matrix A is $NPOP^2$ times smaller than the type 1 transformation matrix, the type 5 transformation is usually preferred where it can be used.

ISIZE — Vector of length $NTRAN$ containing the number of response functions defined by the k -th transformation. (Input)
Transformation types 2, 3, and 4 have the same number of output responses as are input, and elements of **ISIZE** corresponding to transformations of these types should reflect this fact. Transformation types 1 and 5 can either increase or, more commonly, decrease the number of responses. For transformation type 5, if m linear transformations are defined for each population, the corresponding element of **ISIZE** should be $m * NPOP$.

AMATS — Vector containing the transformation constants. (Input)
AMATS contains the transformation matrices and vectors needed in the type 1, 4 and 5 transformations. While **AMATS** is a vector, its elements may be treated as a number of matrices or vectors where the number of structures depends upon the transformation types as follows:

ITRAN	Type	Dimension	Length
1	Matrix	m by n	$M * n$
2, 3	Not referenced		0
4	Vector	m	M
5	Matrix	$m/NPOP$ by $n/NPOP$	$M * n/(NPOP * NPOP)$

Here, $m = ISIZE(i)$ and $n = ISIZE(i - 1)$, and $ISIZE(0)$ is not input (in **ISIZE**) but is taken to be $NPOP * NRESP$. Matrices and vectors are stored consecutively in **AMATS** with column elements for matrices stored consecutively as is standard in FORTRAN. Thus, if $ITRAN(1) = 5$ and $ITRAN(2) = 4$, with $NREP = 3$, $NPOP = 2$, and $ISIZE(1) = ISIZE(2) = 2$, then the vector **AMATS** would contain in consecutive positions $A(1, 1)$, $A(2, 1)$, $A(1, 2)$, $A(2, 2)$, $A(1, 3)$, $A(2, 3)$, $v(1)$, $v(2)$, $v(3)$, $v(4)$ where A is the matrix for transformation type 5 and v is the vector for transformation type 4.

X — Design matrix of size $ISIZE(NTRAN)$ by $NCOEF$. (Input, if $NCOEF > 0$)
X contains the design matrix for predicting the transformed cell probabilities **F** from the covariates stored in **X**. If $NCOEF = 0$, **X** is not referenced and can be a 1 by 1 matrix in the calling program.

NH — Vector of length NUMH. (Input, if NCOEF > 0)
NH(*i*) contains the number of consecutive rows in H used to specify hypothesis *i*. If NCOEF = 0, NH is not referenced and can be a vector of length 1 in the calling program.

H — Matrix of size *m* by NCOEF containing the constants to be used in the multivariate hypothesis tests. (Input, if NCOEF > 0)
Here, *m* is the sum of the elements in NH. Each hypothesis is of the form $H_0 : C * COEF = 0$, where C for the *i*-th hypothesis is NH(*i*) rows of H, and COEF is estimated in the linear model. The first NH(1) rows of H make up the first hypothesis, the next NH(2) rows make up the second hypothesis, etc. If NCOEF = 0, H is not referenced and can be a 1 by 1 matrix in the calling program.

CHSQ — NUMH + 1 by 3 matrix containing the results of the hypothesis tests. (Output, if NCOEF > 0)
The first row of CHSQ contains the results for test 1, the next row contains the results for test 2, etc. The last row of CHSQ contains a test of the adequacy of the model. Within each row, the first column contains the chi-squared statistic, the second column contains its degrees of freedom, and the last column contains the probability of a larger chi-squared. If NCOEF = 0, CHSQ is not referenced and can be a 1 by 1 matrix in the calling program.

COEF — NCOEF by 4 matrix containing the coefficient estimates and related statistics. (Output, if NCOEF > 0)
The columns of coefficient are as follows:

Col. Statistic

- 1 Coefficient estimate
- 2 Estimated standard error of the coefficient
- 3 *z*-statistic for a test that the coefficient equals 0 versus the Two-sided alternative
- 4 *p*-value corresponding to the *z*-statistic

If NCOEF = 0, COEF is not referenced and can be a 1 by 1 matrix in the calling program.

COVCF — NCOEF by NCOEF matrix containing the estimated variances and covariances of COEF. (Output, if NCOEF > 0)
If NCOEF = 0, COVCF is not referenced and can be a 1 by 1 matrix in the calling program.

F — Vector of length ISIZE(NTRAN) containing the transformed probabilities, the responses. (Output)

COVF — Matrix of size ISIZE(NTRAN) by ISIZE(NTRAN) containing the estimated variances and covariances of F. (Output)

RESID — $\text{ISIZE}(\text{NTRAN})$ by 4 matrix containing a case analysis for the transformed probabilities as estimated by the linear model. (Output, if $\text{NCOEF} > 0$)

The linear model gives $F = X * \text{BETA}$. The columns of **RESID** are as follows:

Col. Description

1	Residual
2	Standard error
3	Leverage
4	Standardized residual

If $\text{NCOEF} = 0$, **RESID** is not referenced and can be a 1 by 1 matrix in the calling program.

Optional Arguments

NRESP — Number of cells in each population. (Input)

Default: $\text{NRESP} = \text{size}(\text{TABLE}, 1)$.

NPOP — Number of populations. (Input)

Default: $\text{NPOP} = \text{size}(\text{TABLE}, 2)$.

LDTABL — Leading dimension of **TABLE** exactly as specified in the dimension statement in the calling program. (Input)

Default: $\text{LDTABL} = \text{size}(\text{TABLE}, 1)$.

NTRAN — Number of transformations to be applied to the cell probabilities. (Input)

Cell probabilities are computed as the frequency count for the cell divided by the population sample size. Set $\text{NTRAN} = 0$ if a linear model predicting the cell probabilities is to be used.

Default: $\text{NTRAN} = \text{size}(\text{ITRAN}, 1)$.

NCOEF — Number of coefficients in the linear model relating the transformed probabilities F to the design matrix X . (Input)

Let F denote the vector result of applying the NTRAN transformations, and assume that the model gives $F = X * \text{COEF}$. Then, NCOEF is the length of COEF .

Default: $\text{NCOEF} = \text{size}(X, 2)$.

LDX — Leading dimension of X exactly as specified in the dimension statement in the calling program. (Input)

Default: $\text{LDX} = \text{size}(X, 1)$.

NUMH — Number of multivariate hypotheses to be tested on the coefficients in COEF .

(Input, if $\text{NCOEF} > 0$)

If `NCOEF = 0`, `NUMH` is not referenced.
Default: `NUMH = size (NH,1)`.

LDH — Leading dimension of `H` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDH = size(H,1)`.

IPRINT — Printing option. (Input)
Default: `IPRINT = 0`.

IPRINT Action

0 No printing is performed.

1 Print all output arrays and vectors.

2 Print all output arrays and vectors as well as the matrices and vectors in `AMATS`.

LDCHSQ — Leading dimension of `CHSQ` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDCHSQ = size (CHSQ,1)`.

LDCOEF — Leading dimension of `COEF` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDCOEF = size (COEF,1)`.

LDCOVC — Leading dimension of `COVCF` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDCOVC = size (COVCF,1)`

LDCOVF — Leading dimension of `COVF` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDCOVF = size (COVF,1)`.

LDRESI — Leading dimension of `RESID` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDRESI = size (RESID,1)`.

FORTRAN 90 Interface

Generic: `CALL CTWLS (TABLE, ITRAN, ISIZE, AMATS, X, NH, H, CHSQ, COEF, COVCF, F, COVF, RESID [,...])`

Specific: The specific interface names are `S_CTWLS` and `D_CTWLS`.

FORTRAN 77 Interface

Single: CALL CTWLS (NRESP, NPOP, TABLE, LDTABL, NTRAN, ITRAN,
 ISIZE, AMATS, NCOEF, X, LDX, NUMH, NH, H, LDH, IPRINT,
 CHSQ, LDCHSQ, COEF, LDCOEF, COVCF, LDCOVC, F, COVF, LDCOVF,
 RESID, LDRESI)

Double: The double precision name is DCTWLS.

Example 1

This example is taken from Landis, Stanish, Freeman, and Koch (1976), pages 213-217.
Generalized kappa statistics are computed via vector functions of the form:

$$F(p) = \exp(A_4 \ln(A_3 \exp(A_2 \ln(A_1 p))))$$

where p is the cell probabilities. The raw frequencies are given as two 4×4 contingency tables. These tables are reorganized as a single 16×2 table for input into CTWLS. The input tables are

$$\begin{pmatrix} 38 & 5 & 0 & 1 \\ 33 & 11 & 3 & 0 \\ 10 & 14 & 5 & 6 \\ 3 & 7 & 3 & 10 \end{pmatrix} \begin{pmatrix} 5 & 3 & 0 & 0 \\ 3 & 11 & 4 & 0 \\ 2 & 13 & 3 & 4 \\ 1 & 2 & 4 & 14 \end{pmatrix}$$

Two generalized kappa statistics using two different sets of weights are computed for each population. Hypothesis tests are then performed on the four resulting generalized kappa statistics. In this example, the matrix of covariates is an identity matrix so that tests on the responses are performed.

```
USE CTWLS_INT
INTEGER IPRINT, LDCHSQ, LDCOEF, LDCOVC, LDCOVF, LDH, LDRESI, &
LDTABL, LDX, NCOEF, NPOP, NTRAN
PARAMETER (IPRINT=2, LDCHSQ=10, LDCOEF=4, LDCOVC=4, LDCOVF=4, &
LDH=10, LDRESI=4, LDTABL=16, LDX=4, NCOEF=4, NPOP=2, &
NTRAN=8)
!
INTEGER ISIZE(NTRAN), ITRAN(NTRAN), NH(9)
REAL A1(10,16), A2(18,10), A3(4,18), A4(2,4), AMATS(420), &
CHSQ(LDCHSQ,3), COEF(LDCOEF,4), COVCF(LDCOVC,NCOEF), &
COVF(LDCOVF,LDCOVF), F(LDX), H(LDH,4), &
RESID(LDRESI,4), TABLE(LDTABL,NPOP), X(LDX,NCOEF)
!
EQUIVALENCE (A1, AMATS(1)), (A2, AMATS(161)), (A3, AMATS(341)), &
(A4, AMATS(413))
!
DATA TABLE/38, 5, 0, 1, 33, 11, 3, 0, 10, 14, 5, 6, 3, 7, 3, 10, &
5, 3, 0, 0, 3, 11, 4, 0, 2, 13, 3, 4, 1, 2, 4, 14/
DATA X/1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1/
DATA NH/1, 1, 1, 1, 1, 1, 2, 1, 1/
DATA H/1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, -1, 0, 0, 0, 0, 1, 0, &
1, 0, 0, 0, 1, 0, 1, -1, 0, -1, 0, 0, 0, 0, 0, 1, -1, 0, &
-1, 0, -1/
```



```

DATA ITRAN/5, 2, 5, 3, 5, 2, 5, 3/
DATA ISIZE/20, 20, 36, 36, 8, 8, 4, 4/
DATA A1/1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, &
      .5, 1, 0, 0, 0, 0, 0, 1, 0, 0, .25, 1, 0, 0, 0, 0, 0, 0, 1, &
      0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, .5, 0, 1, 0, 0, 0, 1, 0, &
      0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, .5, 0, 1, 0, 0, 0, 0, &
      0, 1, 0, .25, 0, 0, 1, 0, 1, 0, 0, 0, 0, .25, 0, 0, 1, 0, &
      0, 1, 0, 0, 0, .5, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, &
      0, 0, 0, 0, 1, 0, .5, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, &
      0, 1, 0, 1, 0, 0, 0, .25, 0, 0, 0, 1, 0, 0, 1, 0, 0, .5, 0, &
      0, 0, 1, 0, 0, 0, 1, 1, 1/
DATA A2/1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, &
      0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, &
      0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, &
      0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, &
      0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, &
      1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, &
      0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, &
      0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, &
      0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, &
      1/
DATA A3/-1, -1, 0, 0, 0, -.5, 1, .5, 0, -.25, 1, .75, 0, 0, 1, &
      1, 0, -.5, 1, .5, -1, -1, 0, 0, 0, -.5, 1, .5, 0, -.25, 1, &
      .75, 0, -.25, 1, .75, 0, -.5, 1, .5, -1, -1, 0, 0, 0, -.5, &
      1, .5, 0, 0, 1, 1, 0, -.25, 1, .75, 0, -.5, 1, .5, -1, -1, &
      0, 0, 1, 0, 0, 0, 0, 1, 0, 0/
DATA A4/1, 0, 0, 1, -1, 0, 0, -1/
!
CALL CTWLS (TABLE, ITRAN, ISIZE, AMATS, X, NH, H, &
           CHSQ, COEF, COVCF, F, COVF, RESID, IPRINT=IPRINT)
!
END

```

Output

Hypothesis Tests on Coefficients

H-1	1	0	0	0
H-2	0	1	0	0
H-3	1	-1	0	0
H-4	0	0	1	0
H-5	0	0	0	1
H-6	0	0	1	-1
H-7	1	0	-1	0
	0	1	0	-1
H-8	1	0	-1	0
H-9	0	1	0	-1

Hypothesis Chi-Squared Statistics				
Hypothesis	Chi-Squared	Degrees of freedom		p-value
1	16.99	1	1	0.0000
2	39.70	1	1	0.0000
3	39.54	1	1	0.0000
4	14.27	1	1	0.0002
5	30.07	1	1	0.0000
6	28.76	1	1	0.0000
7	1.07	2	2	0.5850
8	0.90	1	1	0.3425
9	1.06	1	1	0.3040

Model Test	Chi-Squared	Degrees of freedom	p-value
	0.00	0	NaN

Coefficient Statistics				
	Coefficient	Standard Error	Statistic	p-value
1	0.2079	0.05	4.12	0.0000
2	0.3150	0.05	6.30	0.0000
3	0.2965	0.08	3.78	0.0002
4	0.4069	0.07	5.48	0.0000

Asymptotic Coefficient Covariance				
	1	2	3	4
1	2.5457E-03	2.3774E-03	0.	0.
2		2.4988E-03	0.	0.
3			6.1629E-03	5.6229E-03
4				5.5069E-03

Residual Analysis				
	Residual	Standard Error	Leverage	Standardized Residual
1	0.0000	0.0000	1.0000	NaN
2	0.0000	0.0000	1.0000	NaN
3	0.0000	0.0000	1.0000	NaN
4	0.0000	0.0000	1.0000	NaN

Transformed Probabilities	
1	0.2079
2	0.3150
3	0.2965
4	0.4069

Asymptotic Covariance of the Transformed Probabilities				
	1	2	3	4
1	2.5457E-03	2.3774E-03	0.	0.
2		2.4988E-03	0.	0.
3			6.1629E-03	5.6229E-03
4				5.5069E-03

Linear transformation matrix, by population, for transformation 5

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

1	1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000
2	0.000	0.000	0.000	0.000	1.000	1.000	1.000	1.000	0.000
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	1.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000
6	0.000	1.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000
7	0.000	0.000	1.000	0.000	0.000	0.000	1.000	0.000	0.000
8	0.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000	0.000
9	1.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000
10	1.000	0.500	0.250	0.000	0.500	1.000	0.500	0.250	0.250

	10	11	12	13	14	15	16		
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000		
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000		
3	1.000	1.000	1.000	0.000	0.000	0.000	0.000		
4	0.000	0.000	0.000	1.000	1.000	1.000	1.000		
5	0.000	0.000	0.000	1.000	0.000	0.000	0.000		
6	1.000	0.000	0.000	0.000	1.000	0.000	0.000		
7	0.000	1.000	0.000	0.000	0.000	1.000	0.000		
8	0.000	0.000	1.000	0.000	0.000	0.000	1.000		
9	0.000	1.000	0.000	0.000	0.000	0.000	1.000		
10	0.500	1.000	0.500	0.000	0.250	0.500	1.000		

Linear transformation matrix, by population, for transformation 5

	1	2	3	4	5	6	7	8	9
1	1.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000
2	1.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000
3	1.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000
4	1.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000
5	0.000	1.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000
6	0.000	1.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000
7	0.000	1.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000
8	0.000	1.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000
9	0.000	0.000	1.000	0.000	1.000	0.000	0.000	0.000	0.000
10	0.000	0.000	1.000	0.000	0.000	1.000	0.000	0.000	0.000
11	0.000	0.000	1.000	0.000	0.000	0.000	1.000	0.000	0.000
12	0.000	0.000	1.000	0.000	0.000	0.000	0.000	1.000	0.000
13	0.000	0.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000
14	0.000	0.000	0.000	1.000	0.000	1.000	0.000	0.000	0.000
15	0.000	0.000	0.000	1.000	0.000	0.000	1.000	0.000	0.000
16	0.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000	0.000
17	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
18	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

	10
1	0.000
2	0.000
3	0.000
4	0.000
5	0.000
6	0.000
7	0.000
8	0.000
9	0.000
10	0.000

```

11  0.000
12  0.000
13  0.000
14  0.000
15  0.000
16  0.000
17  0.000
18  1.000

```

```

      Linear transformation matrix, by population, for transformation 5
      1      2      3      4      5      6      7      8      9
1 -1.000  0.000  0.000  0.000  0.000 -1.000  0.000  0.000  0.000
2 -1.000 -0.500 -0.250  0.000 -0.500 -1.000 -0.500 -0.250 -0.250
3  0.000  1.000  1.000  1.000  1.000  0.000  1.000  1.000  1.000
4  0.000  0.500  0.750  1.000  0.500  0.000  0.500  0.750  0.750

      10      11      12      13      14      15      16      17      18
1  0.000 -1.000  0.000  0.000  0.000  0.000 -1.000  1.000  0.000
2 -0.500 -1.000 -0.500  0.000 -0.250 -0.500 -1.000  0.000  1.000
3  1.000  0.000  1.000  1.000  1.000  1.000  0.000  0.000  0.000
4  0.500  0.000  0.500  1.000  0.750  0.500  0.000  0.000  0.000

```

```

Linear transformation matrix, by population, for transformation 5
      1      2      3      4
      1  1.000  0.000 -1.000  0.000
      2  0.000  1.000  0.000 -1.000

```

Comments

1. Workspace may be explicitly provided, if desired, by use of C2WLS/DC2WLS. The reference is:

```

CALL C2WLS (NRESP, NPOP, TABLE, LDTABL, NTRAN,
ITRAN, ISIZE, AMATS, NCOEF, X, LDX, NUMH, NH, H, LDH, IPRINT,
CHSQ, LDCHSQ, COEF, LDCOEF, COVCF, LDCOVC, F, COVF, LDCOVF,
RESID, LDRESI, PDER, FRQ, EST, XX, WK, IWK, WWK)

```

The additional arguments are as follows:

PDER — Work vector of length $ISIZE(NTRAN) * \max(NPOP * NRESP, ISIZE(i))$ if $NTRAN$ is greater than zero. **PDER** is not used and can be dimensioned of length 1 if $NTRAN = 0$.

FRQ — Work vector of length $NPOP$.

EST — Work vector of length $NPOP * NRESP + ISIZE(1) + \dots + ISIZE(NTRAN)$.

XX — Work vector of length $(NCOEF + 1) * ISIZE(NTRAN)$ if $NCOEF$ is greater than zero. If $NCOEF = 0$, **XX** is not referenced and can be a vector of length 1 in the calling program.

WK — Work vector of length $3(\max(\text{NPOP} * \text{NRESP}, \text{ISIZE}(i))) + \text{NCOEF} + 1$.

IWK — Work vector of length $\max(\text{NH}(i))$ if NUMH is greater than 0. If $\text{NCOEF} = 0$, **IWK** is not referenced and can be a vector of length 1 in the calling program.

WWK — Work vector of length $\max(\text{NH}(i)) * (4 + \text{NCOEF} + \max(\text{NCOEF}, \max(\text{NH}(i))))$ if NUMH is greater than 0. If $\text{NUMH} = 0$, **WWK** is not referenced and can be a vector of length 1 in the calling program.

2. Informational error

Type Code

4 1 A negative response occurred while performing a logarithmic transformation. The logarithm of a negative number is not allowed.

Description

Routine **CTWLS** performs weighted least-squares analysis of a general $p = \text{NPOP}$ population by $r = \text{NRESP}$ response categories per population contingency table. After division by the sample size, there are $n = pr$ cell probabilities.

Define $s = \text{ISIZE}(\text{NTRAN})$ responses f_i such that each response is obtained from the cell probabilities as $f_i = g_i(p_1, p_2, \dots, p_n)$, for $i = 1, \dots, s$. Call the functions g_i the response functions". Then, if

$$\hat{\Sigma}_f$$

is the asymptotic covariance matrix of the responses, and X is a design matrix for a linear model predicting $f = X\beta$ with $q = \text{NCOEF}$ coefficients $\beta = \text{COEF}$, then **CTWLS** performs a weighted least-squares analysis of the model $f = X\beta$ where the generalized weights are given by

$$\hat{\Sigma}_f = \text{COVF}$$

Estimates obtained in this way are best asymptotic normal estimates of β .

Let

$$\hat{\Sigma}_p$$

denote the estimated variance-covariance matrix of the estimated cell probabilities, and let $(\partial g_i / \partial p_j)$ denote the matrix of partial derivatives of g_i with respect to p_j . Then,

$$\hat{\Sigma}_f$$

is given by

$$\hat{\Sigma}_f = \left(\frac{\partial g_i}{\partial p_j} \right) \hat{\Sigma}_p \left(\frac{\partial g_i}{\partial p_j} \right)^T$$

where the (i, j) -th element in

$$\hat{\Sigma}_p$$

is computed as

$$p_i(\delta_{ij} - p_j)$$

Here, $\delta_{ij} = 1$ if $i = j$ and is zero otherwise.

In CTWLS, the transformations g_i are defined by successive application of one of five types of simpler transformations. Let $p_i = h_{0,j}$ for $j = 1, \dots, n$ denote the n cell probabilities, and let $h_{i,j}$ denote the `ISIZE(i)` responses obtained after i simple transformations have been performed with h_i denoting the corresponding vector of estimates. Then, the simple transformations are defined by:

1. Linear: $h_{i+1} = A_i h_i$ where A_i is a matrix of coefficients specified via the vector `AMATS` in CTWLS.
2. Logarithmic: $h_{i+1,j} = \ln(h_{i,j})$ where $j = 1, \dots, \text{ISIZE}(i)$. That is, take the logarithm of each of the responses.
3. Exponential: $h_{i+1,j} = \exp(h_{i,j})$ where $j = 1, \dots, \text{ISIZE}(i)$. That is, take the exponential of each of the responses.
4. Additive: $h_{i+1,j} = h_{i,j} + v_j$, where $j = 1, \dots, \text{ISIZE}(i)$, and v_j is specified via the vector `AMATS` in CTWLS. Additive transformations are generally used to adjust for zero cells or to apply a continuity correction to the cell probabilities.
5. Linear (by population):

$$h_{i+1}^j = A_i h_i^j \text{ where } h_i^j$$

is the vector of responses at stage i in the j -th population, and A_i is a matrix of coefficients specified via `AMATS`.

Given the responses f_i and their covariances

$$\hat{\Sigma}_f$$

estimates for β are computed via generalized least squares as

$$\hat{\beta} = (X^T \hat{\Sigma}_f^{-1} X)^{-1} X^T \hat{\Sigma}_f^{-1} f$$

Let Σ_β denote the asymptotic covariance matrix of β . Then, Σ_β is estimated by

$$\hat{\Sigma}_\beta = (X^T \hat{\Sigma}_f^{-1} X)^{-1}$$

Hypothesis tests of the form $H_o : C_i \beta = 0$ are performed when requested. Here, C_i is a matrix of coefficients specified via a submatrix of the matrix `H`. Results are returned in the vector `CHSQ`. The asymptotic chi-squared test for testing the null hypothesis is given by

$$\chi^2 = (C_i \beta)^T (C_i \hat{\Sigma} \beta)^{-1} C_i \beta$$

This test has $q_i = \text{rank}(C_i)$ degrees of freedom. If zero degrees of freedom are returned, the hypothesis cannot be tested in the original parameterization.

A test of the model checks that the residuals obtained from the model $f = X\beta$ are not too large. This test, which has $s - q$ degrees of freedom, is an asymptotic chi-squared test and is computed as

$$Q = (f - X\hat{\beta})^T (\hat{\Sigma}_f)^{-1} (f - X\hat{\beta})$$

Residuals from the generalized linear model are easily computed as

$$r_i = f_i - x_i \hat{\beta}$$

where x_i is the row of the design matrix X corresponding to the i -th observation. This residual has the asymptotic variance

$$\hat{\sigma}_i^2 = (\hat{\Sigma}_f)_{ii} \left(1 - \left(X (X^T \hat{\Sigma}_f X)^{-1} X^T \right)_{ii} \right)$$

where $(A)_{ii}$ denotes the i -th diagonal element of matrix A . A standardized residual is then computed as

$$z = r_i / \hat{\sigma}_i$$

which has an asymptotic standard normal distribution if the model is correct.

The leverage of observation i , v_i , is computed as

$$v_i = \left(X (X^T \hat{\Sigma}_f^{-1} X)^{-1} X^T \hat{\Sigma}_f^{-1} \right)_{ii}$$

It is a measure of the importance of the observation in the predicted values. Values greater than $2q/s$ are large.

Because the tests performed by CTWLS are asymptotic ones, the user should treat the results with caution. The reported asymptotic p -values are most likely to be exact when the number of counts in each cell is large (say 5 or more), and less exact for smaller cell counts. Care should also be taken to avoid illegal operations. For example, the routine returns an error message when the log of a negative or zero value is attempted. When this occurs, the user should either use a continuity correction (i.e. modify the transformations used by adding a constant to all cells or to the cell resulting in the illegal operation) or abandon the model.

Additional Example

Example 2

The second example is taken from Prentice (1976) and involves a logistic fit to the mortality of beetles after exposure to various concentrations of carbon disulphide. Because one of the cells on input has a count of zero and it is not possible to take the logarithm of zero, a constant 0.5 is added to each cell prior to calling CTWLS. The model can be expressed as

$$\ln \frac{p_{i1}}{p_{i2}} = \mu + \beta_1 x$$

where i indexes the 8 populations. The data is given as:

X	f₁	f₂
1.690	6	53
1.724	13	47
1.755	18	44
1.784	28	28
1.811	52	11
1.836	53	6
1.861	61	1
1.883	60	0

For comparison, a maximum fit yields

$$\hat{\mu} = .74 \text{ and } \hat{\beta} = 34.3$$

(see STAT routine CTGLM, page 557).

```

USE CTWLS_INT

INTEGER      IPRINT, LDCHSQ, LDCOEF, LDCOVC, LDCOVF, LDH, LDRESI, &
             LDTABL, LDX, NCOEF, NPOP, NRESP, NTRAN, NUMH

PARAMETER   (IPRINT=2, LDCOVF=8, LDH=1, LDX=8, NCOEF=2, NPOP=8, &
             NRESP=2, NTRAN=2, NUMH=0, LDCHSQ=NUMH+1, &
             LDCOEF=NCOEF, LDCOVC=NCOEF, LDRESI=LDX, LDTABL=NRESP)

!

INTEGER      ISIZE(NTRAN), ITRAN(NTRAN), NH(1)
REAL         AMATS(2), CHSQ(LDCHSQ,3), COEF(LDCOEF,4), &
             COVCF(LDCOVC,NCOEF), COVF(LDCOVF,LDCOVF), F(LDX), &
             H(LDH,4), RESID(LDRESI,4), TABLE(LDTABL,NPOP), &
             X(LDX,NCOEF)

!

DATA TABLE/6, 53, 13, 47, 18, 44, 28, 28, 52, 11, 53, 6, 61, 1, &
           60, 0/, ITRAN/2, 5/, ISIZE/16, 8/, AMATS/1, -1/
DATA X/8*1, 1.690, 1.724, 1.755, 1.784, 1.811, 1.836, 1.861, &
           1.883/

!

TABLE = TABLE + 0.5

!

CALL CTWLS (TABLE, ITRAN, ISIZE, AMATS, X, NH, H, &
           CHSQ, COEF, COVCF, F, COVF, RESID, NUMH=NUMH, &
           IPRINT=IPRINT)

!

END

```


Output

Test of the Model
 Chi-Squared Degrees of freedom p-value
 8.43 6 0.2081

Coefficient Statistics
 Coefficient Standard Error Statistic p-value
 1 -55.6590 5.02 -11.10 0.0000
 2 31.4177 2.83 11.09 0.0000

Asymptotic Coefficient Covariance
 1 2
 1 25.16 -14.20
 2 8.024

Residual Analysis
 Residual Standard Error Leverage Standardized Residual
 1 0.4552 0.3232 0.6052 1.4086
 2 0.2368 0.2480 0.6468 0.9548
 3 -0.3568 0.2413 0.7608 -1.4787
 4 -0.3902 0.2285 0.7440 -1.7076
 5 0.2800 0.2761 0.7192 1.0141
 6 0.0840 0.3484 0.7036 0.2410
 7 0.9042 0.7749 0.8791 1.1670
 8 1.2953 1.3777 0.9413 0.9402

Transformed Probabilities
 1 -2.108
 2 -1.258
 3 -0.878
 4 0.000
 5 1.518
 6 2.108
 7 3.714
 8 4.796

Asymptotic Covariance of the Transformed Probabilities
 1 2 3 4 5
 1 0.1725 0. 0. 0. 0.
 2 9.5127E-02 0. 0. 0.
 3 7.6526E-02 0. 0.
 4 7.0175E-02 0.
 5 0.1060
 6 7 8
 1 0. 0. 0.
 2 0. 0. 0.
 3 0. 0. 0.
 4 0. 0. 0.
 5 0. 0. 0.
 6 0.1725 0. 0.
 7 0.6829 0.
 8 2.017

Linear transformation matrix, by population, for transformation 5

	1	2
	1.000	-1.000

Chapter 6: Nonparametric Statistics

Routines

6.1. One Sample or Matched Samples		
6.1.1 Tests of Location		
Sign test for percentiles	SIGNT	592
Wilcoxon signed rank test	SNRNK	595
6.1.2 Tests for Trend		
Noether test for cyclical trend	NCTRD	599
Cox and Stuart trends test in dispersion and location	SDPLC	602
6.1.3 Ties		
Tie statistics	NTIES	607
6.2. Two Independent Samples		
Wilcoxon rank sum test	RNKSM	609
Includance test	INCLD	613
6.3. More than Two Samples		
6.3.1 One-way Tests of Location		
Kruskal-Wallis test for identical medians	KRSKL	616
Bhaskar V test for identical medians	BHAKV	619
6.3.2 Two-way Tests of Location		
Friedmans test for randomized complete block designs	FRDMN	620
Cochran Q test for related observations	QTEST	625
6.3.3 Tests for Trends		
Trends test against ordered alternatives	KTRND	627

Usage Notes

Other Chapters

Much of what is considered nonparametric statistics is included in other chapters. Topics of possible interest in other chapters are: nonparametric measures of location and scale (Chapter 1, “Basic Statistics”), quantile estimation (Chapter 1, “Basic Statistics”), nonparametric measures in a contingency table (Chapter 5, “Categorical and Discrete Data Analysis”), measures of

correlation in a contingency table (Chapter 3, “Correlation”), tests of goodness of fit and randomness (Chapter 7, “Tests of Goodness of Fit and Randomness”), and nonparametric routines for density and hazard estimation (Chapter 15, “Density and Hazard Estimation”).

Other Methods

Many of the tests described in this chapter may be computed using the routines described in other chapters after first substituting ranks (or some other score) for the observed values. (Routine `RANKS` (see Chapter 1, Basic Statistics) may be used to compute ranks.) This method for computing nonparametric test statistics is recommended for cases such as unbalanced one-way ANOVA designs for which no nonparametric subroutine is provided.

Missing Values

Most routines described in this chapter automatically handle missing values (NaN, “not a number”; see the Reference Material section of this manual). In these routines, observations that are missing are ignored; the variable `NMISS` is incremented by one for each missing observation. The user should be aware, however, that some routines described in this chapter do not handle missing values. Missing values input to such routines may result in erroneous results.

Tied Observations

Many of the routines described in this chapter contain an argument `FUZZ` in the input. Observations that are within `FUZZ` of each other in absolute value are said to be tied. Moreover, in some routines, an observation within `FUZZ` of some value is said to be equal to that value. In routine `SNRNK` (page 595), for example, such observations are eliminated from the analysis. If `FUZZ = 0.0`, observations must be identically equal before they are considered to be tied. Other positive values of `FUZZ` allow for numerical imprecision or roundoff error.

SIGNT

Performs a sign test of the hypothesis that a given value is a specified quantile of a distribution.

Required Arguments

X — Vector of length `NOBS` containing the input data. (Input)

Q — Hypothesized percentile of the population from which *x* was drawn. (Input)

P — Value in the range (0, 1). (Input)
Q is the $100 * P$ percentile of the population.

NPOS — Number of positive differences $x(j) - Q$, for $j = 1, 2, \dots, \text{NOBS}$. (Output)

NTIE — Number of zero differences (ties) $x(j) - Q$, for $j = 1, 2, \dots, \text{NOBS}$. (Output)

PROB — Binomial probability of N_{POS} or more positive differences in $N_{OBS} - N_{TIE} - N_{MISS}$ trials. (Output)

NMISS — Number of missing values in X . (Output)

Optional Arguments

NOBS — Number of observations. (Input)
Default: $NOBS = \text{size}(X,1)$.

FORTRAN 90 Interface

Generic: `CALL SIGNT (X, Q, P, NPOS, NTIE, PROB, NMISS [,...])`

Specific: The specific interface names are `S_SIGNT` and `D_SIGNT`.

FORTRAN 77 Interface

Single: `CALL SIGNT (NOBS, X, Q, P, NPOS, NTIE, PROB, NMISS)`

Double: The double precision name is `DSIGNT`.

Example

We wish to test the hypothesis that at least 75% of a population is negative. Because $0.923 < 0.95$, we fail to reject the null hypothesis at the 5 percent level of significance.

```
USE SIGNT_INT
USE UMACH_INT

INTEGER    NOBS
REAL      P, Q
PARAMETER (NOBS=19, P=0.75, Q=0.0)
!
INTEGER    NMISS, NOUT, NPOS, NTIE
REAL      PROB, X(NOBS)
!
DATA X/92.0, 139.0, -6.0, 10.0, 81.0, -11.0, 45.0, -25.0, -4.0, &
      22.0, 2.0, 41.0, 13.0, 8.0, 33.0, 45.0, -33.0, -45.0, -12.0/
!
                                Perform sign test
CALL SIGNT (X, Q, P, NPOS, NTIE, PROB, NMISS)
!
                                Print output
CALL UMACH (2, NOUT)
WRITE (NOUT,99996) NPOS
WRITE (NOUT,99997) NTIE
WRITE (NOUT,99998) PROB
WRITE (NOUT,99999) NMISS
!
99996 FORMAT (' Number of positive differences = ', I2)
99997 FORMAT (' Number of ties = ', I2)
```

```

99998 FORMAT (' PROB                               = ', F6.3)
99999 FORMAT (' Number of missing values         = ', I2)
      END

```

Output

```

Number of positive differences = 12
Number of ties                 = 0
PROB                          = 0.923
Number of missing values      = 0

```

Comments

Other probabilities that may be of interest can be computed via routine BINDF (see Chapter 17, Probability Distribution Functions and Inverses).

Description

Routine SIGNT tests hypotheses about the proportion P of a population that lies below a value Q . In continuous distributions, this can be a test that Q is the $100P$ -th percentile of the population from which X was obtained. To carry out testing, SIGNT tallies the number of values above Q in NPOS. The binomial probability of NPOS or more values above Q is then computed using the proportion P and the sample size NOBS (adjusted for the missing observations [NMISS] and ties [NTIE]).

Hypothesis testing is performed as follows for the usual null and alternative hypotheses.

- $H_0 : \Pr(X \leq Q) \leq P$ (the P -th quantile is at least Q)
 $H_1 : \Pr(X \leq Q) > P$
 Reject H_0 if PROB is less than or equal to the significance level.
- $H_0 : \Pr(X \leq Q) \geq P$ (the P -th quantile is no greater than Q)
 $H_1 : \Pr(X \leq Q) < P$
 Reject H_0 if PROB is greater than or equal to one minus the significance level.
- $H_0 : \Pr(X = Q) = P$ (the P -th quantile is Q)
 $H_1 : \Pr(X \leq Q) < P$ or $\Pr(X \leq Q) > P$
 Reject H_0 if PROB is less than or equal to half the significance level or greater than or equal to one minus half the significance level.

The assumptions are as follows:

1. The x_i are a random sample; i.e., they are independent and identically distributed.
2. The measurement scale is at least ordinal; i.e, an ordering less than, greater than, and equal to exists in the observations.

Many uses for the sign test are possible with various values of P and Q . For example, to perform a matched sample test that the difference of the medians of Y and Z is 0.0, let $P = 0.5$, $q = 0.0$, and $X_i = Y_i - Z_i$ in matched observations Y and Z . To test that the median difference is C , let $Q = C$.

SNRNK

Performs a Wilcoxon signed rank test.

Required Arguments

Y — Vector of length `NOBS` containing the data. (Input)

FUZZ — Constant used to determine when a value is 0.0 or when two values are tied. (Input)

When $|Y(i)|$ or $|Y(i) - Y(j)|$ is less than or equal to `FUZZ`, then the i -th observation is taken to be zero, or the i -th and j -th observations are said to be tied, respectively.

STAT — Vector of length 10 containing the computed statistics. (Output)

Statistics are computed in two ways. In method 1, the average rank of tied observations is used, and observations equal to zero are not counted. In method 2, ties are randomly broken, and observations equal to zero are randomly assigned to the positive or negative half line.

I **STAT(I)**

- 1 The positive rank sum, W^+ , using method 1.
- 2 The absolute value of the negative rank sum, W^- , using method 1.
- 3 The standardized (to an asymptotic variance of 1.0) minimum of (W^+, W^-) using method 1.
- 4 The asymptotic probability of not exceeding `STAT(3)` under the null hypothesis that the distribution is symmetric about 0.0.
- 5 The positive rank sum, W^+ , using method 2.
- 6 The absolute value of the negative rank sum, W^- , using method 2.
- 7 The standardized (to an asymptotic variance of 1.0) minimum of (W^+, W^-) using method 2.
- 8 The asymptotic probability of not exceeding `STAT(7)` under the null hypothesis that the distribution is symmetric about 0.0.
- 9 The number of zero observations.
- 10 The total number of observations that are tied, and that are not within `FUZZ` of zero.

Optional Arguments

NOBS — Number of observations. (Input)

Default: *NOBS* = size (*Y*,1).

NMISS — Number of missing values in *Y*. (Output)

FORTRAN 90 Interface

Generic: CALL SNRNK (*Y*, *FUZZ*, *STAT* [,...])

Specific: The specific interface names are *S_SNRNK* and *D_SNRNK*.

FORTRAN 77 Interface

Single: CALL SNRNK (*NOBS*, *Y*, *FUZZ*, *STAT*, *NMISS*)

Double: The double precision name is *DSNRNK*.

Example

This example illustrates the application of the Wilcoxon signed rank test to a test on two matched samples (matched pairs). A test that the median difference is 10.0 (rather than 0.0) is performed by subtracting 10.0 from each of the differences prior to calling *SNRNK*. The routine *RNSET* (see Chapter 18, Random Number Generation) is used to set the seed. As can be seen from the output, the null hypothesis is rejected. The warning error will always be printed when the number of observations is 50 or less unless printing is turned off for warning errors. See routine *ERSET* (Reference Material).

```
USE WRRRN_INT
USE RNSET_INT
USE SNRNK_INT
USE UMACH_INT

INTEGER NOBS
REAL FUZZ
PARAMETER (FUZZ=0.0001, NOBS=7)
!
INTEGER I, NMISS, NOUT
REAL STAT(10), W(NOBS), X(NOBS), Y(NOBS)
!
DATA W/223, 216, 211, 212, 209, 205, 201/
DATA X/208, 205, 202, 207, 206, 204, 203/
!
DO 10 I=1, NOBS
    Y(I) = X(I) - W(I) - 10.0
10 CONTINUE
!
                                Print Y prior to calling SNRNK
CALL WRRRN ('Y', Y, 1, NOBS, 1, 0)
!
                                Initialize the seed
CALL RNSET (123457)
!
```

```

CALL SNRNK (Y, FUZZ, STAT, NMISS=NMISS)
!
!                               Print output
CALL UMACH (2, NOUT)
WRITE (NOUT,99999) STAT(1), STAT(5), STAT(2), STAT(6), STAT(3), &
STAT(7), STAT(4), STAT(8), STAT(9), STAT(10), &
NMISS
!
99999 FORMAT (' Statistic                Method 1      Method 2', &
/, ' W+.....', F9.0, 4X, F9.0, /, &
' W-.....', F9.0, 4X, F9.0, /, &
' Standardized Minimum.....', F9.4, 4X, F9.4, /, &
' p-value.....', F9.4, 4X, F9.4, //, &
' Number of zeros.....', F9.0, /, ' Number of ', &
'ties.....', F9.0, /, ' Number of missing.....', &
I5)
!
END

```

Output

	Y						
	1	2	3	4	5	6	7
	-25.00	-21.00	-19.00	-15.00	-13.00	-11.00	-8.00

```

*** WARNING  ERROR 4 from SNRNK.  NOBS = 7.  The number of
***          observations, NOBS, is less than 50, and exact
***          tables should be referenced for probabilities.

```

Statistic	Method 1	Method 2
W+.....	0.	0.
W-.....	28.	28.
Standardized Minimum.....	-2.3664	-2.3664
p-value.....	0.0090	0.0090
Number of zeros.....	0.	
Number of ties.....	0.	
Number of missing.....	0	

Comments

1. Workspace may be explicitly provided, if desired, by use of S2RNK/DS2RNK. The reference is:

```
CALL S2RNK (NOBS, Y, FUZZ, ISEED, STAT, NMISS, IR, YRANK)
```

The additional arguments are as follows:

IR — Work vector of length NOBS.

YRANK — Work vector of length NOBS.

If Y is not needed, Y and YRANK can share the same storage locations.

2. Informational errors

Type	Code	
3	4	NOBS is less than 50 and exact tables should be referenced for probabilities.
3	5	Each element of Y is within FUZZ of 0. STAT(1) through STAT(8) are set to NaN (not a number).
3. The signed rank statistic provides a test of the hypothesis that the population median is equal to zero. To test that the median is equal to some other value, say, 10.0, use the routine SADD (IMSL MATH/LIBRARY) to subtract 10.0 from each observation prior to calling SNRNK.
4. The signed rank test can be used to test that the medians of two matched random variables are equal. This is the nonparametric equivalent of the paired *t*-test. To use SNRNK to perform this test, use the routine SAXPY (IMSL MATH/LIBRARY) prior to calling SNRNK to compute the differences, $Y(i) - X(i)$. Then, call SNRNK with these differences.
5. The routine RNUN (see Chapter 18, Random Number Generation) is used to randomly break ties. The routine RNSET (see Chapter 18, Random Number Generation) can be used to initialize the seed of the random number generator. The routine RNOPT (see Chapter 18, Random Number Generation) can be used to select the form of the generator.

Description

Routine SNRNK performs a Wilcoxon signed rank test of symmetry about zero. In one sample, this test can be viewed as a test that the population median is zero. In matched samples, a test that the medians of the two populations are equal can be computed by first computing difference scores. These difference scores would then be used as input to SNRNK. A general reference for the methods used is Conover (1980).

Routine SNRNK computes statistics for two methods for handling zero and tied observations. In the first method, observations within FUZZ of zero are not counted, and the average rank of tied observations is used. (Observations within FUZZ of each other are said to be tied.) In the second method, observations within FUZZ of zero are randomly assigned a positive or negative sign, and the ranks of tied observations are randomly permuted.

The W^+ and W^- statistics are computed as the sums of the ranks of the positive observations and the sum of the ranks of the negative observations, respectively. Asymptotic probabilities are computed using standard methods (see, e.g., Conover 1980, page 282).

The W^+ and W^- statistics may be used to test the following hypotheses about the median, M . In deciding whether to reject the null hypothesis, use the bracketed statistic if method 2 for handling ties is preferred. Possible null hypotheses and alternatives are given as follows:

- $H_0 : M \leq 0 \quad H_1 : M > 0$
Reject if STAT(1) [or STAT(5)] is too large.

- $H_0 : M \geq 0$ $H_1 : M < 0$
Reject if `STAT(2)` [or `STAT(6)`] is too large.
- $H_0 : M = 0$ $H_1 : M \neq 0$
Reject if `STAT(3)` [or `STAT(7)`] is too small. Alternatively, if an asymptotic test is desired, reject if $2 * \text{STAT}(4)$ [or $2 * \text{STAT}(8)$] is less than the significance level.
- Tabled values of the test statistic can be found in the references. If possible, tabled values should be used. If the number of nonzero observations is too large, then the asymptotic probabilities computed by `SNRNC` can be used.

The assumptions required for the hypothesis tests are as follows:

1. The distribution of each X_i is symmetric.
 2. The X_i are mutually independent.
 3. All X_i 's have the same median.
 4. An ordering of the observations exists (i.e., $X_1 > X_2$ and $X_2 > X_3$ implies that $X_1 > X_3$).
- If other assumptions are made, related hypotheses that are more (or less) restrictive can be tested.

NCTRD

Performs the Noether test for cyclical trend.

Required Arguments

X — Vector of length `NOBS` containing the observations in chronological order. (Input)

FUZZ — Value to be used in determining when consecutive observations in `x` are tied. (Input)
If $|x(i+1) - x(i)|$ is less than or equal to `FUZZ`, then $x(i+1)$ and $x(i)$ are said to be tied.

NSTAT — Vector of length 6 containing output statistics. (Output)

I **NSTAT(I)**

- 1 The number of consecutive sequences of length three used to detect cyclical trend when tying middle elements are eliminated from the sequence, and the next consecutive observation is used.
- 2 The number of monotonic sequences of length three in the set defined by `NSTAT(1)`.

- I** **NSTAT(I)**
- 3 The number of monotonic sequences where tied threesomes are counted as nonmonotonic.
- 4 The number of monotonic sequences where tied threesomes are counted as monotonic.
- 5 The number of middle observations eliminated because they were tied in forming the NSTAT(1) sequences.
- 6 The number of tied sequences found in forming the NSTAT(3) and NSTAT(4) sequences. A sequence is called a tied sequence if the middle element is tied with either of the two other elements.

P — Vector of length 3 containing the probabilities of NSTAT(2) or more, NSTAT(3) or more, or NSTAT(4) or more monotonic sequences. (Output)
 If NSTAT(1) is less than 1, P(1) is set to NaN (not a number).

Optional Arguments

NOBS — Number of observations. (Input)
 NOBS must be greater than or equal to 3.
 Default: NOBS = size (x,1).

NMISS — Number of missing (NaN, not a number) values in x. (Output)

FORTRAN 90 Interface

Generic: CALL NCTRD (X, FUZZ, NSTAT, P [, ...])

Specific: The specific interface names are S_NCTRD and D_NCTRD.

FORTRAN 77 Interface

Single: CALL NCTRD (NOBS, X, FUZZ, NSTAT, P, NMISS)

Double: The double precision name is DNCTRD.

Example

A test for cyclical trend in a sequence of 1000 randomly generated observations is performed. Because of the sample used, there are no ties and all three test statistics yield the same result.

```
!                                     SPECIFICATIONS FOR PARAMETERS
      USE IMSL_LIBRARIES

      INTEGER      NOBS
      REAL         FUZZ
```

```

PARAMETER (FUZZ=0.0, NOBS=1000)
!
INTEGER   ISEED, NSTAT(6)
REAL      P(3), X(NOBS)
!
DATA ISEED/123457/
!
CALL RNSET (ISEED)
CALL RNUN (X)
!
CALL NCTRD (X, FUZZ, NSTAT, P)           Noether test
!
CALL WRIRN ('NSTAT', NSTAT, 1, 6, 1, 0)  Print results
CALL WRRRN ('P', P, 1, 3, 1, 0)
!
END

```

Output

		NSTAT			
1	2	3	4	5	6
333	107	107	107	0	0

		P	
1	2	3	
0.6979	0.6979	0.6979	

Comments

- Informational errors

Type	Code	Description
3	3	NSTAT(1), which is used to determine NSTAT(3) and NSTAT(4), is less than 8. The asymptotic probabilities will not be exact.
3	4	At least one tie was detected in X.
- If NOBS is greater than or equal to 3 but NSTAT(1) is less than one, P(1) will be set to NaN. The remaining statistics and associated probabilities will be determined and returned as described.

Description

Routine NCTRD performs the Noether test for cyclical trend (Noether 1956) for a sequence of measurements. In this test, the observations are first divided into sets of three consecutive observations. Each set is then inspected, and if the set is monotonically increasing or decreasing, the count variable is incremented.

The count variables, NSTAT(2), NSTAT(3), and NSTAT(4), differ in the manner in which ties are handled. A tie can occur in a set (of size three) only if the middle element is tied with either of the two ending elements. Tied ending elements are not considered. In NSTAT(2), tied middle observations are eliminated, and a new set of size 3 is obtained by using the next observation in the sample. In NSTAT(3), the original set of size three is used, and tied middle observations are counted as nonmonotonic. In NSTAT(4), tied middle observations are counted as monotonic.

The probabilities of occurrence of the counts are obtained from the binomial distribution with $p = 1/3$, where p is the probability that a random sample of size three from a continuous distribution is monotonic. The binomial sample size is, of course, the number of sequences of size three found (adjusted for ties).

Hypothesis test:

$$H_0 : q = \Pr(X_i > X_{i-1} > X_{i-2}) + \Pr(X_i < X_{i-1} < X_{i-2}) \leq 1/3 \quad H_1 : q > 1/3$$

Reject if $\mathbb{P}(1)$ (or $\mathbb{P}(2)$ or $\mathbb{P}(3)$ depending on the method used for handling ties) is less than the significance level of the test.

Assumption: The observations are independent and are from a continuous distribution.

SDPLC

Performs the Cox and Stuart sign test for trends in dispersion and location.

Required Arguments

X — Vector of length `NOBS` containing the observations in chronological order. (Input)

K — Number of consecutive `X` elements to be used to measure dispersion. (Input)
Not required if `IOPT` is different from zero.

IDS — Dispersion measure option. (Input)
If `IDS` is zero, the range is used as a measure of dispersion. Otherwise, the centered sum of squares is used. Not required if `IOPT` is different from zero.

FUZZ — Value used to determine when elements in `X` are tied. (Input)
If $|x(i) - x(j)|$ is less than or equal to `FUZZ`, `x(i)` and `x(j)` are said to be tied. `FUZZ` must be nonnegative.

NSTAT — Vector of length 8. (Output)
The first 4 elements of `NSTAT` are the output statistics when the observations are divided into two groups. The last 4 elements are the output statistics when the observations are divided into three groups.

I **NSTAT(I)**

1 Number of negative differences (two groups)

2 Number of positive differences (two groups)

3 Number of zero differences (two groups)

4 Number of differences used to calculate `PSTAT(1)` through `PSTAT(4)` (two groups).

5 Number of negative differences (three groups)

I NSTAT(I)

6 Number of positive differences (three groups)

7 Number of zero differences (three groups)

8 Number of differences used to calculate *PSTAT*(5) through *PSTAT*(8) (three groups).

PSTAT — Vector of length 8 containing probabilities. (Output)

The first four elements of *PSTAT* are computed from two groups of observations.

I PSTAT(I)

1 Probability of *NSTAT*(1) + *NSTAT*(3) or more negative signs (ties are considered negative).

2 Probability of obtaining *NSTAT*(2) or more positive signs (ties are considered negative).

3 Probability of *NSTAT*(1) + *NSTAT*(3) or more negative signs (ties are considered positive).

4 Probability of obtaining *NSTAT*(2) or more positive signs (ties are considered positive).

The last four elements of *PSTAT* are computed from three groups of observations.

I PSTAT(I)

5 Probability of *NSTAT*(1) + *NSTAT*(3) or more negative signs (ties are considered negative).

6 Probability of obtaining *NSTAT*(2) or more positive signs (ties are considered negative).

7 Probability of *NSTAT*(1) + *NSTAT*(3) or more negative signs (ties are considered positive).

8 Probability of obtaining *NSTAT*(2) or more positive signs (ties are considered positive).

Optional Arguments

NOBS — Number of observations. (Input)

Default: *NOBS* = size (x,1).

IOPT — Statistic option parameter. (Input)

If *IOPT* = 0, the Cox and Stuart tests for trends in dispersion are computed. Otherwise, the Cox and Stuart tests for trends in location are computed.

Default: *IOPT* = 0.

NMISS — Number of missing values in *X*. (Output)

FORTRAN 90 Interface

Generic: CALL SDPLC (*X*, *K*, *IDS*, *FUZZ*, *NSTAT*, *PSTAT* [,...])

Specific: The specific interface names are *S_SDPLC* and *D_SDPLC*.

FORTRAN 77 Interface

Single: CALL SDPLC (*NOBS*, *X*, *IOPT*, *K*, *IDS*, *FUZZ*, *NSTAT*, *PSTAT*,
 NMISS)

Double: The double precision name is *DSDPLC*.

Example

This example illustrates both the location and dispersion tests. The data, which are taken from Bradley (1968), page 176, give the closing price of AT&T on the New York stock exchange for 36 days in 1965. Tests for trends in location (*IOPT* = 1), and for trends in dispersion (*IOPT* = 0) are performed. Trends in location are found.

```
USE IMSL_LIBRARIES
INTEGER    IDS, K, NOBS, IS
REAL      FUZZ
PARAMETER  (FUZZ=0.001, IDS=0, K=2, NOBS=36)
!
INTEGER    IOPT, NSTAT(8)
REAL      PSTAT(8), X(NOBS)
!
DATA X/9.5, 9.875, 9.25, 9.5, 9.375, 9.0, 8.75, 8.625, 8.0, &
      8.25, 8.25, 8.375, 8.125, 7.875, 7.5, 7.875, 7.875, 7.75, &
      7.75, 7.75, 8.0, 7.5, 7.5, 7.125, 7.25, 7.25, 7.125, 6.75, &
      6.5, 7.0, 7.0, 6.75, 6.625, 6.625, 7.125, 7.75/
!
                                Tests for trends in location
IOPT = 1
IS = 1
CALL SDPLC (X, K, IDS, FUZZ, NSTAT, PSTAT, IOPT=IOPT)
!
                                Print results
CALL WROPT (-6, IS, 1)
CALL WRIRN ('NSTAT', NSTAT, 1, 8, 1, 0)
CALL WRRRN ('PSTAT', PSTAT, 1, 8, 1, 0)
!
                                Tests for trends in dispersion
IOPT = 0
CALL SDPLC (X, K, IDS, FUZZ, NSTAT, PSTAT)
!
                                Print results
CALL WRIRN ('NSTAT', NSTAT, 1, 8, 1, 0)
CALL WRRRN ('PSTAT', PSTAT, 1, 8, 1, 0)
!
END
```

Output

*** WARNING ERROR 5 from SDPLC. At least one tie is detected in X.

```
          NSTAT
1   2   3   4   5   6   7   8
0  17   1  18   0  12   0  12

          PSTAT
          1           2           3           4           5
1.00000      0.00007      1.00000      0.00000      1.00000

          6           7           8
0.00024      1.00000      0.00024
```

*** WARNING ERROR 5 from SDPLC. At least one tie is detected in X.

```
          NSTAT
1   2   3   4   5   6   7   8
4   3   2   9   4   2   0   6

          PSTAT
          1           2           3           4           5
0.253906      0.910156      0.746094      0.500000      0.343750

          6           7           8
0.890625      0.343750      0.890625
```

Comments

1. Workspace may be explicitly provided, if desired, by use of *S2PLC/DS2PLC*. The reference is:

```
CALL S2PLC (NOBS, X, IOPT, K, IDS, FUZZ, NSTAT, PSTAT, NMISS,
XWK)
```

The additional argument is:

XWK — Work vector of length *NOBS*.

If *X* is not needed, *X* and *XWK* can share the same storage location.

2. Informational errors
Type Code

```
4         4   NSTAT(4) is too small to continue with a dispersion test.
3         5   At least one tie is detected in X.
```

Description

Routine *SDPLC* tests for trends in dispersion or location in a sequence of random variables depending upon the value of the input variable *IOPT*. A derivative of the sign test is used (see Cox and Stuart 1955).

Location Test

For the location test (`IOPT = 1`) with two groups, the observations are first divided into two groups with the middle observation thrown out if there are an odd number of observations. Each observation in group one is then compared with the observation in group two that has the same lexicographical order. A count is made of the number of times a group-one observation is less than (`NSTAT(1)`), greater than (`NSTAT(2)`), or equal to (`NSTAT(3)`), its counterpart in group two. Two observations are counted as equal if they are within `FUZZ` of one another.

In the three-group test, the observations are divided into three groups, with the center group losing observations if the division is not exact. The first and third groups are then compared as in the two-group case, and the counts are stored in `NSTAT(5)` through `NSTAT(7)`.

Probabilities in `PSTAT` are computed using the binomial distribution with sample size equal to the number of observations in the first group (`NSTAT(4)` or `NSTAT(8)`), and binomial probability $p = 0.5$.

Dispersion Test

The dispersion tests proceed exactly as with the tests for location, but using one of two derived dispersion measures. The input value `K` is used to define `NOBS/K` groups of consecutive observations starting with observation 1. The first `K` observations define the first group, the next `K` observations define the second group, etc., with the last observations omitted if `NOBS` is not evenly divisible by `K`. A dispersion score is then computed for each group as either the range (`IDS = 0`), or a multiple of the variance (`IDS ≠ 0`) of the observations in the group. The dispersion scores form a derived sample. The tests proceed on the derived sample as above.

Ties

Ties are defined as occurring when a group one observation is within `FUZZ` of its last group counterpart. Ties imply that the probability distribution of X is not strictly continuous, which means that $\Pr(X_1 > X_2) \neq 0.5$ under the null hypothesis of no trend (and the assumption of independent identically distributed observations). When ties are present, the computed binomial probabilities are not exact, and the hypothesis tests will be conservative.

Hypothesis tests

In the following, i indexes an observation from group 1, while j indexes the corresponding observation in group 2 (two groups) or group 3 (three groups).

- $H_0 : \Pr(X_i > X_j) = \Pr(X_i < X_j) = 0.5$
 $H_1 : \Pr(X_i > X_j) < \Pr(X_i < X_j)$
Hypothesis of upward trend. Reject if `PSTAT(3)` (or `PSTAT(7)`) is less than the significance level.
- $H_0 : \Pr(X_i > X_j) = \Pr(X_i < X_j) = 0.5$
 $H_1 : \Pr(X_i > X_j) > \Pr(X_i < X_j)$
Hypothesis of downward trend. Reject if `PSTAT(2)` (or `PSTAT(6)`) is less than the significance level.

- $H_0 : \Pr(X_i > X_j) = \Pr(X_i < X_j) = 0.5$
 $H_1 : \Pr(X_i > X_j) \neq \Pr(X_i < X_j)$
 Two tailed test. Reject if $2 \max(\text{PSTAT}(2), \text{PSTAT}(3))$ (or $2 \max(\text{PSTAT}(6), \text{PSTAT}(7))$) is less than the significance level.

Assumptions

1. The observations are a random sample; i.e., the observations are independently and identically distributed.
2. The distribution is continuous.

TIES

Computes tie statistics for a sample of observations.

Required Arguments

X — Vector of length `NOBS` containing the observations. (Input)
`X` must be ordered monotonically increasing with all missing values removed.

FUZZ — Value used to determine ties. (Input)
 Observations i and j are tied if the successive differences $x(k+1) - x(k)$ between observations i and j , inclusive, are all less than `FUZZ`. `FUZZ` must be nonnegative.

TIES — Vector of length 4 containing the tie statistics. (Output)
 The tie statistics are returned in `TIES` and are computed as follows:

$$\text{TIES}(1) = \sum_{j=1}^{\tau} [t_j(t_j - 1)]/2$$

$$\text{TIES}(2) = \sum_{j=1}^{\tau} [t_j(t_j - 1)(t_j + 1)]/12$$

$$\text{TIES}(3) = \sum_{j=1}^{\tau} t_j(t_j - 1)(2t_j + 5)$$

$$\text{TIES}(4) = \sum_{j=1}^{\tau} t_j(t_j - 1)(t_j - 2)$$

where t_j is the number of ties in the j -th group (rank) of ties, and τ is the number of tie groups in the sample.

Optional Arguments

NOBS — The number of observations. (Input)
 Default: `NOBS = size(X,1)`.

FORTRAN 90 Interface

Generic: CALL NTIES (X, FUZZ, TIES [,...])

Specific: The specific interface names are S_NTIES and D_NTIES.

FORTRAN 77 Interface

Single: CALL NTIES (NOBS, X, FUZZ, TIES)

Double: The double precision name is DNTIES.

Example

We want to compute tie statistics for a sample of length 7.

```
USE NTIES_INT
USE WRRRN_INT
INTEGER      NOBS
REAL         FUZZ
PARAMETER   (FUZZ=0.001, NOBS=7)
!
!   REAL      TIES(4), X(NOBS)
!
!   DATA X/1.0, 1.0001, 1.0002, 2.0, 3.0, 3.0, 4.0/
!           Compute tie statistics
!   CALL NTIES (X, FUZZ, TIES)
!           Print results
!   CALL WRRRN ('TIES', TIES, 1, 4, 1, 0)
!
END
```

Output

```
          TIES
1         2         3         4
4.00    2.50    84.00    6.00
```

Description

Routine `NTIES` computes tie statistics for a monotonically increasing sample of observations. “Tie statistics” are statistics that may be used to correct a continuous distribution theory nonparametric test for tied observations in the data. Observations i and j are tied if the successive differences $x(k+1) - x(k)$, inclusive, are all less than `FUZZ`. Note that if each of the monotonically increasing observations is equal to its predecessor plus a constant, if that constant is less than `FUZZ`, then all observations are contained in one tie group. For example, if `FUZZ = 0.11`, then the following observations are all in one tie group.

```
0.0, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1.00
```

RNKSM

Performs the Wilcoxon rank sum test.

Required Arguments

X — Vector of length `NOBSX` containing the first sample. (Input)

Y — Vector of length `NOBSY` containing the second sample. (Input)

FUZZ — Constant used to determine ties in *X* and *Y*. (Input)

If $|z_i - z_j| \leq \text{FUZZ}$, then z_i and z_j are said to be tied, where z_i is the i -th element of *X* or *Y*.
FUZZ must be nonnegative.

STAT — Vector of length 10 containing the output statistics. (Output)

I **STAT(I)**

1 Wilcoxon W statistic (the sum of the ranks of the *X* observations) adjusted for ties in such a manner that W is as small as possible.

2 $2 * E(W) - W$, where $E(W)$ is the expected value of W .

3 Probability of obtaining a statistic less than or equal to the minimum of (W , $2E(W) - W$).

4 W statistic adjusted for ties in such a manner that W is as large as is possible.

5 `STAT(2)`; but adjusted for ties as in 4.

6 `STAT(3)`; but adjusted for ties as in 4.

7 w statistic with average ranks used in place of tied ranks.

8 Estimated standard error of `STAT(7)` under the null hypothesis of no difference.

9 Standard normal score associated with `STAT(7)`.

10 Two-sided p -value associated with `STAT(9)`.

Optional Arguments

NOBSX — Number of observations in *X*. (Input)
Default: `NOBSX = size(X,1)`.

NOBSY — Number of observations in *Y*. (Input)
Default: `NOBSY = size(Y,1)`.

NMISSX — Number of missing (NaN, not a number) observations in *X*. (Output)

NMISSY — Number of missing (NaN, not a number) observations in *Y*. (Output)

FORTRAN 90 Interface

Generic: CALL RNKSM (*X*, *Y*, FUZZ, STAT [,...])

Specific: The specific interface names are S_RNKSM and D_RNKSM.

FORTRAN 77 Interface

Single: CALL RNKSM (NOBSX, *X*, NOBSY, *Y*, FUZZ, STAT,
 NMISSX, NMISSY)

Double: The double precision name is DRNKSM.

Example

The following example is taken from Conover (1980, page 224). It involves the mixing time of 2 mixing machines using a total of 10 batches of a certain kind of batter, 5 batches for each machine. The null hypothesis is not rejected at the 5 percent level of significance. The warning error is always printed when one or more ties are detected unless printing for warning errors is turned off. See routine ERSET (Reference Material).

```
USE RNKSM_INT
USE UMACH_INT

INTEGER    NOBSX, NOBSY
REAL      FUZZ
PARAMETER (FUZZ=0.001, NOBSX=5, NOBSY=5)
!
INTEGER    I, NMISSX, NMISSY, NOUT
REAL      STAT(10), X(NOBSX), Y(NOBSY)
!
DATA X/7.3, 6.9, 7.2, 7.8, 7.2/
DATA Y/7.4, 6.8, 6.9, 6.7, 7.1/
!
CALL RNKSM (X, Y, FUZZ, STAT, NMISSX=NMISSX, NMISSY=NMISSY)
!
                                Print the results
CALL UMACH (2, NOUT)
WRITE (NOUT,99999) (STAT(I),I=1,10), REAL(NMISSX), REAL(NMISSY)
!
99999 FORMAT (' Wilcoxon W statistic .....', F5.1, &
/, ' 2*WBAR - W .....', &
F5.1, /, ' p-value .....', &
, F7.3, /, ' Adjusted Wilcoxon statistic ', '.....', &
, '.....', F5.1, /, ' Adjusted 2*WBAR - W ', '.....', &
'...', '.....', F5.1, /, ' Adjusted p-value ', &
'.....', F7.3, /, ' W statistic ', &
'for averaged ranks .....', F5.1, /, ' Standard ' &
, 'error of W (averaged ranks) .....', F7.3, /, &
```

```

      ' Standard normal score of W (averaged ranks) .', F7.3, &
      /, ' Two-sided p-value of W (averaged ranks) .....', &
      F7.3, //, ' Number of missing for X .....', &
      , F5.1, /, ' Number of missing for Y ', '.....', &
      , '.....', F5.1)
!
      END

```

Output

```

*** WARNING  ERROR 5 from RNKSM.  At least one tie is detected between the
***          samples.
Wilcoxon W statistic ..... 34.0
2*WBAR - W ..... 21.0
p-value ..... 0.110
Adjusted Wilcoxon statistic ..... 35.0
Adjusted 2*WBAR - W ..... 20.0
Adjusted p-value ..... 0.075
W statistic for averaged ranks ..... 34.5
Standard error of W (averaged ranks) ..... 4.758
Standard normal score of W (averaged ranks) . 1.471
Two-sided p-value of W (averaged ranks) ..... 0.141

Number of missing for X ..... 0.0
Number of missing for Y ..... 0.0

```

Comments

1. Workspace may be explicitly provided, if desired, by use of R2KSM/DR2KSM. The reference is:

```
CALL R2KSM (NOBSX, X, NOBSY, Y, FUZZ, STAT, NMISSX, NMISSY, IWK,
           YWK)
```

The additional arguments are as follows:

IWK — Integer work vector of length NOBSX + NOBSY

YWK — Work vector of length NOBSX + NOBSY.

2. Informational errors

Type	Code	
3	4	Both NOBSX and NOBSY are less than 25. Tabled critical values for W should be used.
3	5	Tied observations occurred between the samples.
4	6	Each element of X and/or Y is a missing (NaN, not a number) value.
3. The Mann-Whitney U statistic is given in terms of W as $U = W - K * (K + 1)/2$, where $K = \text{NOBSX}$, and $W = \text{STAT}(1)$ (or $\text{STAT}(4)$). Tables of critical values for W are available in the references given in the manual document.
4. For greatest efficiency in computing W , the X sample should be the smallest sample.

Description

Routine `RNKSM` performs the Wilcoxon rank sum test for identical population distribution functions. The Wilcoxon test is a linear transformation of the Mann-Whitney U test. If the difference between the two populations can be attributed solely to a difference in location, then the Wilcoxon test becomes a test of equality of the population means (or medians) and is the nonparametric equivalent of the two-sample t -test.

Routine `RNKSM` obtains ranks in the combined sample after first eliminating missing values from the data. The rank sum statistic is then computed as the sum of the ranks in the x sample. Three methods for handling ties are used. (A tie is counted when two observations are within `FUZZ` of each other.) The first method uses the largest possible rank for tied observations in the smallest sample, while the second method uses the smallest possible rank for these observations. Thus, the range of possible rank sums is obtained. The third, method for handling tied observations between samples uses the average rank of the tied observations.

Asymptotic standard normal scores are computed for the W score (based upon a variance that has been adjusted for ties) when average ranks are used (see Conover 1980, page 217), and the probability associated with the two-sided alternative is computed.

Hypothesis Tests

In each test following, the first line gives the hypothesis (and its alternative) under the assumptions 1 to 3 below, while the second line gives the hypothesis when assumption 4 is also true. The rejection region is the same for both hypotheses and is given in terms of method 3 for handling ties. Another output statistic should be used (`STAT(1)` or `STAT(4)`) if another method for handling ties is desired.

- $H_0 : \Pr(x < y) = 0.5$ $H_1 : \Pr(x < y) \neq 0.5$
 $H_0 : E(x) = E(y)$ $H_1 : E(x) \neq E(y)$
Reject if `STAT(10)` is less than the significance level of the test. Alternatively, reject H_0 if `STAT(7)` is too large or too small.
- $H_0 : \Pr(x < y) \leq 0.5$ $H_1 : \Pr(x < y) > 0.5$
 $H_0 : E(x) \geq E(y)$ $H_1 : E(x) < E(y)$
Reject if `STAT(7)` is too small.
- $H_0 : \Pr(x < y) \geq 0.5$ $H_1 : \Pr(x < y) < 0.5$
 $H_0 : E(x) \leq E(y)$ $H_1 : E(x) > E(y)$
Reject if `STAT(7)` is too large.

Assumptions

1. x and y are a random sample from their respective populations.
2. All observations are mutually independent.
3. The measurement scale is at least ordinal (i.e., an ordering less than, greater than, or equal to exists among the observations).

4. If $F(X)$ and $G(Y)$ are the distribution functions of X and Y , respectively, then $G(Y) = F(X + c)$ for some constant c (i.e., the distribution of Y is at worst a translation of the distribution of X).

Tables of critical values of the W statistic are given in the references for small samples.

INCLD

Performs an inclusion test.

Required Arguments

X— Vector of length `NOBSX` containing the data for the first sample. (Input)

Y— Vector of length `NOBSY` containing the data for the second sample. (Input)

ILX— Index of the element in the ordered first sample to be used as the low endpoint of the range considered. (Input)
ILX must be greater than zero and less than *IHX*.

IHX— Index of the element in the ordered first sample to be used as the high endpoint of the range considered. (Input)
IHX must be greater than *ILX* and less than or equal to `NOBSX`.

FUZZ— Value used to determine ties. (Input)
 If a second sample element is within *FUZZ* of the *ILX* or *IHX* order statistics in the first sample, a tie will be counted.

STAT— Vector of length 4 containing the statistics. (Output)
 In the description below, $(x_{(ILX)}, x_{(IHX)})$ is the interval from the *ILX* ordered first sample value to the *IHX* ordered first sample value (i.e., from the *ILX* to the *IHX* order statistics in the first sample).

I **STAT(*I*)**

1 Number of ties detected.

2 Number of untied elements in the second sample that are outside the interval $(x_{(ILX)}, x_{(IHX)})$.

3 Probability of `STAT(2)` or more second sample elements lying outside $(x_{(ILX)}, x_{(IHX)})$.

4 Probability of `STAT(1) + STAT(2)` or more elements in the second sample lying outside $(x_{(ILX)}, x_{(IHX)})$.

Optional Arguments

NOBSX — Number of observations in the first sample. (Input)

Default: $NOBSX = size(X,1)$.

NOBSY — Number of observations in the second sample. (Input)

Default: $NOBSY = size(Y,1)$.

NMISSX — Number of missing (NaN, not a number) values in X . (Output)

NMISSY — Number of missing (NaN, not a number) values in Y . (Output)

FORTRAN 90 Interface

Generic: `CALL INCLD (X, Y, ILX, IHX, FUZZ, STAT [,...])`

Specific: The specific interface names are `S_INCLD` and `D_INCLD`.

FORTRAN 77 Interface

Single: `CALL INCLD (NOBSX, X, NOBSY, Y, ILX, IHX, FUZZ, STAT, NMISSX, NMISSY)`

Double: The double precision name is `DINCLD`.

Example

The following example, which is an adaptation of a problem in Bradley (1968, page 234) illustrates the use of `INCLD` to test that equal proportions of two populations lie between the endpoints of the first sample.

```
USE INCLD_INT
USE WRRRL_INT

INTEGER    IHX, ILX, NOBSX, NOBSY
REAL      FUZZ
PARAMETER (FUZZ=0.0001, IHX=12, ILX=1, NOBSX=12, NOBSY=15)
!
REAL      STAT(4), X(NOBSX), Y(NOBSY)
CHARACTER CLABEL(5)*30, RLABEL(1)*4
!
DATA X/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12/
DATA Y/0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2/
DATA RLABEL/'NONE'/
DATA CLABEL/' ', '%/Number of ties', '%/Number outside', &
           'p-value%/untied', 'p-value%/both'/
!
           Perform inclusion test
CALL INCLD (X, Y, ILX, IHX, FUZZ, STAT)
!
           Print results
CALL WRRRL ('STAT', STAT, RLABEL, CLABEL, 1, 4, 1)
!
END
```

Output

	STAT		p-value	p-value
Number of ties	Number outside		untied	both
0.000	7.000		0.038	0.038

Comments

1. Workspace may be explicitly provided, if desired, by use of I2CLD/DI2CLD. The reference is:

```
CALL I2CLD (NOBSX, X, NOBSY, Y, ILX, IHX, FUZZ, STAT, NMISSX,
           NMISSY, WK)
```

The additional argument is:

WK — Work vector of length NOBSX. If X is not needed, X and WK can share the same storage locations.

2. If ILX = 1 and IHX = NOBSX, INCLD tests the hypothesis that the second population lies in equal proportion to the first population, between the endpoints of the first sample.
3. If ILX = (NOBSX + 1)/4 and IHX = 3 * (NOBSX + 1)/4, the first and the third quartile estimates of the first population are being considered. The null hypothesis may be that the first and second samples are drawn from the same population.

Description

Routine INCLD tests that an equal proportion of two populations lies between the ILX and IHX order statistics of the first sample, and that the densities are equal at the two points. Let X_{il} and X_{ih} denote the two order statistics in the first sample, where $l = \text{ILX}$, and $h = \text{IHX}$. Then, the probability of exactly i observations in the second sample being outside of the interval (X_{il}, X_{ih}) is hypergeometric and is given by

$$\text{Pr}_i = \frac{\binom{M-b+(ih-il+1)}{M-i} \binom{1}{0} \binom{N-(ih-il+1)}{b}}{\binom{N+M}{M}}$$

where M is the sample size in the first sample ($\text{NOBSX} - \text{NMISSX}$), N is the sample size in the second sample ($\text{NOBSY} - \text{NMISSY}$), and

$$\binom{n}{x}$$

denotes a binomial coefficient. The probability of b or fewer observations in the second sample being outside the interval is given by

$$\Pr = \sum_{i=0}^b \Pr_i$$

Use of this test requires that the population samples sizes, `ILX` and `IHX`, be set prior to sampling or viewing the data. Ties do not present any special problems except when they occur at the interval endpoints X_{il} and X_{ih} . When this occurs for the first sample, no action is taken, but an informative warning message is issued. When a second sample observation is within `FUZZ` of an endpoint, then a tie is counted in `STAT(1)`, and once more, a warning message is issued. In this case, `STAT(3)` and `STAT(4)` can be considered as bounds for the actual probability.

KRSKL

Performs a Kruskal-Wallis test for identical population medians.

Required Arguments

NI — Vector of length `NGROUP` containing the number of responses for each of the `NGROUP` groups. (Input)

Y — Vector of length `NI(1) + ... + NI(NGROUP)` that contains the responses for each of the `NGROUP` groups. (Input)

Y must be sorted by group, with the `NI(1)` observations in group 1 coming first, the `NI(2)` observations in group two coming second, and so on.

FUZZ — Constant used to determine ties in **Y**. (Input)

If (after sorting) $|Y(i) - Y(i + 1)|$ is less than or equal to `FUZZ`, then a tie is counted. `FUZZ` must be nonnegative.

STAT — Vector of length 4 containing the Kruskal-Wallis statistics. (Output)

I **STAT(I)**

1 Kruskal-Wallis H statistic.

2 Asymptotic probability of a larger H under the null hypothesis of identical population medians.

3 H corrected for ties.

4 Asymptotic probability of a larger H (corrected for ties) under the null hypothesis of identical populations.

Optional Arguments

NGROUP — Number of groups. (Input)

Default: `NGROUP = size(NI,1)`.

FORTRAN 90 Interface

Generic: CALL KRSKL (NI, Y, FUZZ, STAT [,...])

Specific: The specific interface names are S_KRSKL and D_KRSKL.

FORTRAN 77 Interface

Single: CALL KRSKL (NGROUP, NI, Y, FUZZ, STAT)

Double: The double precision name is DKRSKL.

Example

The following example is taken from Conover (1980, page 231). The data represents the yields per acre of four different methods for raising corn. Since $H = 25.5$, the four methods are clearly different. The warning error is always printed when the Beta approximation is used, unless printing for warning errors is turned off. See IMSL routine ERSET (Reference Material).

```
USE KRSKL_INT
USE UMACH_INT

INTEGER    NGROUP
REAL      FUZZ
PARAMETER (FUZZ=0.001, NGROUP=4)
!
INTEGER    NI(NGROUP), NOUT
REAL      STAT(4), Y(34)
!
DATA NI/9, 10, 7, 8/
DATA Y/83, 91, 94, 89, 89, 96, 91, 92, 90, 91, 90, 81, 83, 84, &
      83, 88, 91, 89, 84, 101, 100, 91, 93, 96, 95, 94, 78, 82, &
      81, 77, 79, 81, 80, 81/
!
!                               Perform Kruskal-Wallis test
CALL KRSKL (NI, Y, FUZZ, STAT)
!
!                               Print results
CALL UMACH (2, NOUT)
WRITE (NOUT,99999) STAT
!
99999 FORMAT (' H (no ties)      = ', F8.1, '/', ' Prob (no ties) = ', &
             F11.4, '/', ' H (ties)      = ', F8.1, '/', ' Prob (ties)   ' &
             ', ' = ', F11.4)
!
END
```

Output

```
*** WARNING ERROR 6 from KRSKL. The chi-squared degrees of freedom are
*** less than 5, so the Beta approximation is used.
H (no ties)      =      25.5
Prob (no ties) =      0.0000
H (ties)         =      25.6
Prob (ties)      =      0.0000
```

Comments

1. Workspace may be explicitly provided, if desired, by use of `K2SKL/DK2SKL`. The reference is:

```
CALL K2SKL (NGROUP, NI, Y, FUZZ, STAT, IWK, WK, YRNK)
```

The additional arguments are as follows:

IWK — Integer work vector of length m .

WK — Work vector of length m .

YRNK — Work vector of length m .

2. Informational errors

Type Code

3	4	At least one tie was detected in <code>Y</code> .
3	5	All elements of <code>Y</code> are tied. <code>STAT</code> is set to -1.0 .
3	6	The chi-squared degrees of freedom are less than 5, so the Beta approximation is used.

Description

The routine `KRSKL` generalizes the Wilcoxon two-sample test computed by routine `RNKSM` (page 609) to more than two populations. It computes a test statistic for testing that the population distribution functions in each of K populations are identical. Under appropriate assumptions, this is a nonparametric analogue of the one-way analysis of variance. Since more than two samples are involved, the alternative is taken as the analogue of the usual analysis of variance alternative, namely that the populations are not identical.

The calculations proceed as follows: All observations are ranked regardless of the population to which they belong. Average ranks are used for tied observations (observations within `FUZZ` of each other). Missing observations (observations equal to NaN, not a number) are not included in the ranking. Let R_i denote the sum of the ranks in the i -th population. The test statistic H is defined as:

$$H = \frac{1}{S^2} \sum_{i=1}^K \left(\frac{R_i^2}{n_i} - \frac{N(N+1)^2}{4} \right)$$

where N is the total of the sample sizes, n_i is the number of observations in the i -th sample, and S^2 is computed as the (bias corrected) sample variance of the R_i .

The null hypothesis is rejected when `STAT(4)` (or `STAT(2)`) is less than the significance level of the test. If the null hypothesis is rejected, then the procedures given in Conover (1980, page 231) may be used for multiple comparisons. The routine `KRSKL` computes asymptotic probabilities using the chi-squared distribution when the number of groups is 6 or greater, and a Beta approximation (see Wallace 1959) when the number of groups is 5 or less. Tables yielding exact probabilities in small samples may be obtained from Owen (1962).

BHAKV

Performs a Bhapkar V test.

Required Arguments

NI — Vector of length `NGROUP` containing the number of responses for each of the `NGROUP` groups. (Input)

Y — Vector of length `NI(1) + NI(2) + ... + NI(NGROUP)` containing the responses for each of the `NGROUP` groups. (Input)
Y must be sorted by group with the `NI(1)` observations for group 1 coming first.

V — Bhapkar V statistic. (Output)

PROB — Asymptotic probability of exceeding *v* under the null hypothesis that the populations are equal. (Output)
Asymptotically, V follows a chi-squared distribution with `NGROUP - 1` degrees of freedom.

Optional Arguments

NGROUP — Number of groups. (Input)
Default: `NGROUP = size (NI,1)`.

FORTRAN 90 Interface

Generic: `CALL BHAKV (NI, Y, V, PROB [,...])`

Specific: The specific interface names are `S_BHAKV` and `D_BHAKV`.

FORTRAN 77 Interface

Single: `CALL BHAKV (NGROUP, NI, Y, V, PROB)`

Double: The double precision name is `DBHAKV`.

Example

We want to test the null hypothesis that three samples of size 3, 2, and 4, respectively, are from the same population using the Bhapkar V statistic.

```
USE BHAKV_INT
USE UMACH_INT

INTEGER      NGROUP
PARAMETER   (NGROUP=3)
!
INTEGER      NI(NGROUP), NOUT
```



```

      REAL          PROB, V, Y(9)
!
      DATA NI/3, 2, 4/
      DATA Y/1, 3, 2, -1, 5, 4, 7, 2, 9/
!
      CALL BHAKV (NI, Y, V, PROB)          Perform Bhapkar V test
!
      CALL UMACH (2, NOUT)                 Print results
      WRITE (NOUT,99998) V
      WRITE (NOUT,99999) PROB
!
99998 FORMAT (' V      = ', F12.5)
99999 FORMAT (' Prob = ', F12.5)
!
      END

```

Output

```

V      =      1.89429
Prob =      0.38785

```

Comments

Workspace may be explicitly provided, if desired, by use of B2AKV/DB2AKV. The reference is

```
CALL B2AKV (NGROUP, NI, Y, V, PROB, IWK, WK, YWK)
```

The additional arguments are as follows:

IWK — Integer work vector of length $NI(1) + \dots + NI(NGROUP) + NGROUP$

WK — Work vector of length $NGROUP$

YWK — Work vector of length $NI(1) + \dots + NI(NGROUP)$. If Y is not needed, Y and YWK can share the same storage locations.

Description

Routine BHAKV tests the hypothesis that several samples are from the same population using the Bhapkar V statistic. Let the number of samples be denoted by $K = NGROUP$. To compute the Bhapkar V statistic, one first computes, for each group i , the statistic t_i = the number of K -tuples that can be formed with one observation from each sample such that the element from population i is the smallest. The sample variance of the ratio of t_i to the total number of such k -tuples is then computed. The Bhapkar V statistic is then a constant c multiplied by this variance, where $c = n(2m - 1)$, $m = NGROUP$, and n is the sum of the sample sizes (after missing values are eliminated).

FRDMN

Performs Friedman's test for a randomized complete block design.

Required Arguments

NB — Number of blocks. (Input)

NT — Number of treatments. (Input)

Y — Vector of length $NB * NT$ containing the observations. (Input)

The first NT positions of **Y** contain the observations on treatments 1, 2, ..., NT in the first block. The second NT positions contain the observations in the second block, etc., and so on.

FUZZ — Constant used to determine ties. (Input)

In the ordered observations, if $|Y(i) - Y(i + 1)|$ is less than or equal to **FUZZ**, then $Y(i)$ and $Y(i + 1)$ are said to be tied.

ALPHA — Critical level for multiple comparisons. (Input)

ALPHA should be between 0 and 1 exclusive.

STAT — Vector of length 6 containing the Friedman statistics. (Output)

Probabilities reported are computed under the appropriate null hypothesis.

I **STAT(I)**

1 Friedman two-sided test statistic.

2 Approximate F value for **STAT(1)**.

3 Page test statistic for testing the ordered alternative that the median of treatment i is less than or equal to the median of treatment $i + 1$, with strict inequality holding for some i .

4 Asymptotic p -value for **STAT(1)**. Chi-squared approximation.

5 Asymptotic p -value for **STAT(2)**. F approximation.

6 Asymptotic p -value for **STAT(3)**. Normal approximation.

SMRNK — Vector of length NT containing the sum of the ranks of each treatment. (Output)

D — Minimum absolute difference in two elements of **SMRNK** to infer at the alpha level of significance that the medians of the corresponding treatments are different. (Output)

FORTRAN 90 Interface

Generic: `CALL FRDMN (NB, NT, Y, FUZZ, ALPHA, STAT, SMRNK, D)`

Specific: The specific interface names are `S_FRDMN` and `D_FRDMN`.

FORTRAN 77 Interface

Single: CALL FRDMN (NB, NT, Y, FUZZ, ALPHA, STAT, SMRNK, D)

Double: The double precision name is DFRDMN.

Example

The following example is taken from Bradley (1968), page 127, and tests the hypothesis that 4 drugs have the same effects upon a person's visual acuity. Five subjects were used.

```
USE FRDMN_INT
USE UMACH_INT

INTEGER    NB, NT
REAL      ALPHA, FUZZ
PARAMETER (ALPHA=0.05, FUZZ=0.001, NB=5, NT=4)
!
INTEGER    NOUT
REAL      D, SMRNK(NT), STAT(6), Y(NB*NT)
!
DATA Y/.39, .55, .33, .41, .21, .28, .19, .16, .73, .69, .64, &
     .62, .41, .57, .28, .35, .65, .57, .53, .60/
!
                                Perform Friedman's test
CALL FRDMN (NB, NT, Y, FUZZ, ALPHA, STAT, SMRNK, D)
!
                                Print results
CALL UMACH (2, NOUT)
WRITE (NOUT,99999) STAT, SMRNK, D
!
99999 FORMAT (' Friedman T.....', F8.2, /, ' Friedman F.....', &
     F8.2, /, ' Page test.....', F8.2, /, ' Prob ', &
     'Friedman T....', F11.5, /, ' Prob Friedman F....', &
     F11.5, /, ' Prob Page test.....', F11.5, /, ' Sum of ', &
     'Ranks.....', 4F8.2, /, ' D.....', F11.5)
!
END
```

Output

```
Friedman T.....      8.28
Friedman F.....      4.93
Page test.....      111.00
Prob Friedman T....    0.04057
Prob Friedman F....    0.01859
Prob Page test.....    0.98495
Sum of Ranks.....    16.00   17.00   7.00   10.00
D.....                6.65638
```

The Friedman null hypothesis is rejected at the $\alpha = .05$ while the Page null hypothesis is not. (A Page test with a monotonic decreasing alternative would be rejected, however.) Using SMRNK and D, one can conclude that treatment 3 is different from treatments 1 and 2, and that treatment 4 is different from treatment 2, all at the $\alpha = .05$ level of significance.

Comments

1. Workspace may be explicitly provided, if desired, by use of F2DMN/DF2DMN. The reference is:

CALL F2DMN (NB, NT, Y, FUZZ, ALPHA, STAT, SMRNK, D, IWK, WK)

The additional arguments are as follows:

IWK — Integer work vector of length NT.

WK — Work vector of length 2 * NT.

2. Informational errors

Type Code

4	5	At least one missing value was detected in Y. No missing values are permitted in this routine since it assumes a complete block design.
3	6	At least one tie was detected within a block.
3	7	The ranks of the treatments were exactly the same in all the blocks.

Description

Routine FRDMN may be used to test the hypothesis of equality of treatment effects within each block in a randomized block design. No missing values are allowed. Ties are handled by using the average ranks. The test statistic is the nonparametric analogue of an analysis of variance F test statistic.

The test proceeds by first ranking the observations within each block. Let A denote the sum of the squared ranks, i.e., let

$$A = \sum_{i=1}^k \sum_{j=1}^b \text{Rank}(Y_{ij})^2$$

where $\text{Rank}(Y_{ij})$ is the rank of the i -th observation within the j -th block, $b = \text{NB}$ is the number of blocks, and $k = \text{NT}$ is the number of treatments. Let

$$B = \frac{1}{b} \sum_{i=1}^k R_i^2$$

where

$$R_i = \sum_{j=1}^b \text{Rank}(Y_{ij})$$

The Friedman test statistic (STAT(1)) is given by:

$$T = \frac{(k-1)(bB - b^2k(k+1)^2/4)}{A - bk(k+1)^2/4}$$

that, under the null hypothesis, has an approximate chi-squared distribution with $k - 1$ degrees of freedom. The asymptotic probability of obtaining a larger chi-squared random variable is returned in `STAT(4)`.

If the F distribution is used in place of the chi-squared distribution, then the usual oneway analysis of variance F -statistic computed on the ranks is used. This statistic, reported in `STAT(2)`, is given by

$$F = \frac{(b-1)T}{b(k-1)-T}$$

and asymptotically follows an F distribution with $(k - 1)$ and $(b - 1)(k - 1)$ degrees of freedom under the null hypothesis. `STAT(5)` is the asymptotic probability of obtaining a larger F random variable. (If $A = B$, `STAT(1)` and `STAT(2)` are set to machine infinity, and the significance levels are reported as $k!/(k!)^b$, unless this computation would cause underflow, in which case the significance levels are reported as zero.) Iman and Davenport (1980) discuss the relative advantages of the chi-squared and F approximations. In general, the F approximation is considered best.

The Friedman T statistic is related both to the Kendall coefficient of concordance and to the Spearman rank correlation coefficient. See Conover (1980) for a discussion of the relationships.

If, at the $\alpha = \text{ALPHA}$ level of significance, the Friedman test results in rejection of the null hypothesis, then an asymptotic test that treatments i and j are different is given by: reject H_0 if $|R_i - R_j| > D$, where

$$D = t_{1-\alpha/2} \sqrt{2b(A-B)/((b-1)(k-1))}$$

where t has $(b - 1)(k - 1)$ degrees of freedom. Page's statistic (`STAT(3)`) is used to test the same null hypothesis as the Friedman test but is sensitive to a monotonic increasing alternative. The Page test statistic is given by

$$Q = \sum_{i=1}^k jR_i$$

It is largest (and thus most likely to reject) when the R_i are monotonically increasing.

Assumptions

The assumptions in the Friedman test are as follows:

1. The k -vectors of responses within each of the b blocks are mutually independent (i.e., the results within one block have no effect on the results within another block).
2. Within each block, the observations may be ranked.

The hypothesis tested is that each ranking of the random variables within each block is equally likely. The alternative is that at least one of the treatments tends to have larger values than one or more of the other treatments. The Friedman test is a test for the equality of treatment means or medians.

QTEST

Performs a Cochran Q test for related observations.

Required Arguments

X — $NOBS$ by $NVAR$ matrix of dichotomized data, containing $NOBS$ readings of zero or one on each of $NVAR$ treatments. (Input)

Q — Cochran's Q statistic. (Output)

PQ — Asymptotic probability of exceeding Q under the null hypothesis of equality of the underlying populations. (Output)

Optional Arguments

$NOBS$ — Number of blocks for each treatment. (Input)
Default: $NOBS = \text{size}(X,1)$.

$NVAR$ — Number of treatments. (Input)
Default: $NVAR = \text{size}(X,2)$.

LDX — Leading dimension of X exactly as specified in the dimension statement in the calling program. (Input)
Default: $LDX = \text{size}(X,1)$.

FORTRAN 90 Interface

Generic: `CALL QTEST (X, Q, PQ [,...])`

Specific: The specific interface names are `S_QTEST` and `D_QTEST`.

FORTRAN 77 Interface

Single: `CALL QTEST (NOBS, NVAR, X, LDX, Q, PQ)`

Double: The double precision name is `DQTEST`.

Example

The following example is taken from Siegel (1956, page 164). It measures the responses of 18 housewives to 3 types of interviews.

```
USE QTEST_INT
USE UMACH_INT

INTEGER    LDX, NVAR
PARAMETER (NVAR=3, LDX=18)
```

!

```

      INTEGER      NOUT
      REAL         PQ, Q, X(LDX,NVAR)
!
      DATA X/0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, &
            1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, &
            0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0/
!
      CALL QTEST (X, Q, PQ)
!
      CALL UMACH (2, NOUT)
      WRITE (NOUT,99999) Q, PQ
!
99999 FORMAT (' Q = ', F6.3, '/', ' PQ = ', F9.5)
!
      END

```

Output

```

Q = 16.667
PQ = 0.00024

```

Comments

- Informational errors

Type	Code	
4	5	X must consist of zeros and ones only.
3	6	X consists of either all ones or all zeros. Q is set to NaN (not a number). PQ is set to 1.0.
- The input data must consist of zeros and ones only. For example, the data may be pass/fail information on `NVAR` questions asked of `NOBS` people or the test responses of `NOBS` individuals to `NVAR` different conditions.
- The resulting statistic is distributed approximately as chi-squared with `NVAR - 1` degrees of freedom if `NOBS` is not too small. `NOBS` greater than or equal to $5 * NVAR$ is a conservative recommendation.

Description

Routine `QTEST` computes the Cochran Q test statistic that may be used to determine whether or not M matched sets of responses differ significantly among themselves. The data may be thought of as arising out of a randomized block design in which the outcome variable must be success (= 1.0) or failure (= 0.0). Within each block a multivariate vector of 1's or 0's is observed. The hypothesis is that the probability of success within a block does not depend upon the treatment.

Assumptions

- The blocks are a random sample from the population of all possible blocks.
- The outcome of each treatment is dichotomous.

Hypothesis

The hypothesis being tested may be stated in at least two ways.

1. H_0 : All treatments have the same effect.
 H_1 : The treatments do not all have the same effect.
2. Let p_{ij} denote the probability of outcome 1.0 in block i , treatment j .

$$H_0 : p_{i1} = p_{i2} = \dots = p_{ic} \text{ for each } i.$$

$$H_1 : p_{ij} \neq p_{ik} \text{ for some } i, \text{ and some } j \neq k.$$

where $c(= \text{NVAR})$ is the number of treatments.

The null hypothesis is rejected if Cochran's Q statistic is too large.

KTRND

Performs k -sample trends test against ordered alternatives.

Required Arguments

NI — Vector of length `NGROUP` that contains the number of responses for each of the `NGROUP` groups. (Input)

X — Vector of length `NI(1) + NI(2) + ... + NI(NGROUP)` containing the responses for each of the `NGROUP` groups. (Input)
All of the responses for group 1 come first, followed by group 2, and so on.

STAT — Vector of length 17 containing the test results. (Output)

I **STAT(I)**

- 1 Test statistic (ties are randomized).
- 2 Conservative test statistic with ties counted in favor of the null hypothesis.
- 3 p -value associated with `STAT(1)`.
- 4 p -value associated with `STAT(2)`.
- 5 Continuity corrected `STAT(3)`.
- 6 Continuity corrected `STAT(4)`.
- 7 Expected mean of the statistic.
- 8 Expected kurtosis of the statistic. (The expected skewness is zero.)

- 9 Total sample size.
- 10 Coefficient of rank correlation based upon STAT(1).
- 11 Coefficient of rank correlation based upon STAT(2).
- 12 Total number of ties between samples.
- 13 The *t*-statistic associated with STAT(3).
- 14 The *t*-statistic associated with STAT(4).
- 15 The *t*-statistic associated with STAT(5).
- 16 The *t*-statistic associated with STAT(6).
- 17 Degrees of freedom for each *t*-statistic.

Optional Arguments

NGROUP — Number of groups. (Input)
 NGROUP must be greater than or equal to 3.
 Default: NGROUP = size (NI,1).

FORTRAN 90 Interface

Generic: CALL KTRND (NI, X, STAT [,...])
 Specific: The specific interface names are S_KTRND and D_KTRND.

FORTRAN 77 Interface

Single: CALL KTRND (NGROUP, NI, X, STAT)
 Double: The double precision name is DKTRND.

Example

The following example is taken from Jonckheere (1954, page 135). It involves four observations in four independent samples.

```

USE RNSET_INT
USE KTRND_INT
USE UMACH_INT

INTEGER    NGROUP
PARAMETER (NGROUP=4)

!
INTEGER    NI(NGROUP), NOUT
REAL      STAT(17), X(16)

```

```

!
DATA NI/4, 4, 4, 4/
DATA X/19, 20, 60, 130, 21, 61, 80, 129, 40, 99, 100, 149, 49, &
    110, 151, 160/
!
CALL RNSET (123457)
!
CALL KTRND (NI, X, STAT)          Get the statistics
!
CALL UMACH (2, NOUT)             Print the results
WRITE (NOUT,99999) STAT
!
99999 FORMAT (' STAT(1) - Test statistic (random) .....', F8.1, &
/, ' STAT(2) - Test statistic (null hypothesis) ..', &
F8.1, /, ' STAT(3) - p-value for STAT(1) .....', &
/, F12.5, /, ' STAT(4) - p-value for STAT(2) ', &
'.....', F12.5, /, ' STAT(5) - Continuity ', &
'corrected STAT(3) .....', F12.5, /, ' STAT(6) - ', &
'Continuity corrected STAT(4) .....', F12.5, /, &
' STAT(7) - Expected mean .....', F8.1, &
/, ' STAT(8) - Expected kurtosis .....', &
F12.5, /, ' STAT(9) - Total sample size .....', &
/, F8.1, /, ' STAT(10)- Rank corr. coef. based on STAT(1) ' &
', ', F12.5, /, ' STAT(11)- Rank corr. coef. based on ', &
'STAT(2) .', F12.5, /, ' STAT(12)- Total number of ties ' &
', '.....', F8.1, /, ' STAT(13)- t-statistic ', &
't-associated w/STAT(3) ..', F10.3, /, ' STAT(14)- ', &
't-statistic associated w/STAT(4) ..', F10.3, /, &
' STAT(15)- t-statistic associated w/STAT(5) ..', F10.3, &
/, ' STAT(16)- t-statistic associated w/STAT(6) ..', &
F10.3, /, ' STAT(17)- Degrees of freedom .....', &
/, F10.3)
!
END

```

Output

```

STAT(1) - Test statistic (random) ..... 46.0
STAT(2) - Test statistic (null hypothesis) .. 46.0
STAT(3) - p-value for STAT(1) ..... 0.01483
STAT(4) - p-value for STAT(2) ..... 0.01483
STAT(5) - Continuity corrected STAT(3) ..... 0.01683
STAT(6) - Continuity corrected STAT(4) ..... 0.01683
STAT(7) - Expected mean ..... 458.7
STAT(8) - Expected kurtosis ..... -0.15365
STAT(9) - Total sample size ..... 16.0
STAT(10)- Rank corr. coef. based on STAT(1) . 0.47917
STAT(11)- Rank corr. coef. based on STAT(2) . 0.47917
STAT(12)- Total number of ties ..... 0.0
STAT(13)- t-statistic associated w/STAT(3) .. 2.264
STAT(14)- t-statistic associated w/STAT(4) .. 2.264
STAT(15)- t-statistic associated w/STAT(5) .. 2.208
STAT(16)- t-statistic associated w/STAT(6) .. 2.208
STAT(17)- Degrees of freedom ..... 36.050

```

Comments

1. Informational errors
Type Code
3 4 At least one tie is detected in X . Randomization is used to break all ties.
3 5 There are no degrees of freedom for the t -statistics. $STAT(3)$ to $STAT(6)$ are set to 0.
2. The closer $STAT(10)$ and $STAT(11)$ are to unity, the more one would be inclined to reject the hypothesis of randomness.
3. Routine $RNUN$ (see Chapter 18, Random Number Generation) is used to randomly break ties. Routine $RNSET$ (see Chapter 18, Random Number Generation) can be used to initialize the seed of the random number generator. The routine $RNOPT$ (see Chapter 18, Random Number Generation) can be used to select the form of the generator.

Description

Routine $KTRND$ performs a k -sample trends test against ordered alternatives. The alternative to the null hypothesis of equality is that $F_1(X) < F_2(X) < \dots < F_k(X)$, where F_1, F_2, \dots , are cumulative distribution functions, and the operator $<$ implies that the less than relationship holds for all values of X . While the trends test used in $KTRND$ requires that the background populations be continuous, ties occurring within a sample have no effect on the test statistic or associated probabilities. Ties between samples are important, however. Two methods for handling ties between samples are used. These are:

1. Ties are randomly split ($STAT(1)$).
2. Ties are counted in a manner that is unfavorable to the alternative hypothesis ($STAT(2)$).

Computational Procedure

Consider the matrices

$$M^{km} = (m_{ij}^{km}) = \begin{pmatrix} 2 & \text{if } X_{ki} < X_{mj} \\ 0 & \text{otherwise} \end{pmatrix}$$

where X_{ki} is the i -th observation in the k -th population, X_{mj} is the j -th observation in the m -th population, and each matrix M^{km} is n_k by n_m where $n_i = NI(i)$. Let S_{km} denote the sum of all elements in M^{km} . Then, $STAT(2)$ is computed as the sum over all elements in S_{km} , minus the expected value of this sum (computed as

$$\sum_{k < m} n_k n_m$$

when there are no ties and the distributions in all populations are equal). In $STAT(1)$, ties are broken randomly, and the element in the summation is taken as 2.0 or 0.0 depending upon the result of breaking the tie.

$STAT(3)$ and $STAT(4)$ are computed using the t distribution. The probabilities reported are asymptotic approximations based upon the t statistics in $STAT(13)$ and $STAT(14)$, which are computed as in Jonckheere (1954, page 141). Similarly, $STAT(5)$ and $STAT(6)$ give the probabilities for $STAT(15)$ and $STAT(16)$, the continuity corrected versions of $STAT(3)$ and $STAT(4)$. The degrees of freedom for each t statistic ($STAT(17)$) are computed so as to make the t distribution selected as close as possible to the actual distribution of the statistic (see Jonckheere 1954, page 141).

$STAT(7)$, the variance of the test statistic $STAT(1)$, and $STAT(8)$, the kurtosis of the test statistic, are computed as in Jonckheere (1954, page 138). The coefficients of rank correlation in $STAT(9)$ and $STAT(10)$ reduce to the Kendall τ statistic when there are just two groups.

Exact probabilities in small samples can be obtained from tables in Jonckheere (1954). Note, however, that the t approximation appears to be a good one.

Assumptions

1. The X_{mi} for each sample are independently and identically distributed according to a single continuous distribution.
2. The samples are independent.

Hypothesis tests

$$H_0 : F_1(x) \geq F_2(x) \geq \dots \geq F_k(x)$$

$$H_1 : F_1(x) < F_2(x) < \dots < F_k(x)$$

Reject if $STAT(3)$ (or $STAT(4)$, or $STAT(5)$ or $STAT(6)$, depending upon the method used) is too large.

Chapter 7: Tests of Goodness of Fit and Randomness

Routines

7.1. General Goodness-of-Fit Tests for a Specified Distribution	
One-sample continuous data Kolmogorov-Smirnov	KSONE 634
Chi-squared test goodness-of-fit test.....	CHIGF 638
Shapiro-Wilk <i>W</i> -test for normality	SPWLK 644
Lilliefors test for an exponential or a normal distribution	LILLF 646
Mardia's test for multivariate normality	MVMMT 649
7.2. Two Sample Tests	
Kolmogorov-Smirnov	KSTWO 654
7.3. Tests for Randomness	
Runs test.....	RUNS 657
Pairs-serial test	PAIRS 661
d^2 test	DSQAR 664
Triplets test	DCUBE 667

Usage Notes

The routines in this chapter are used to test for goodness of fit and randomness. The goodness-of-fit tests are described in Conover (1980). There are two goodness-of-fit tests for general distributions, a Kolmogorov-Smirnov test and a chi-squared test. The user supplies the hypothesized cumulative distribution function for these two tests. There are three routines that can be used to test specifically for the normal or exponential distributions.

The tests for randomness are often used to evaluate the adequacy of pseudorandom number generators. These tests are discussed in Knuth (1981).

The Kolmogorov-Smirnov routines in this chapter compute exact probabilities in small to moderate sample sizes. The chi-squared goodness-of-fit test may be used with discrete as well as continuous distributions.

The Kolmogorov-Smirnov and chi-squared goodness-of-fit test routines allow for missing values (NaN, not a number) in the input data. The routines that test for randomness do not allow for missing values.

KSONE

Performs a Kolmogorov-Smirnov one-sample test for continuous distributions.

Required Arguments

CDF — User-supplied FUNCTION to compute the cumulative distribution function (CDF) at a given value. The form is CDF(Y), where

Y — Value at which CDF is to be evaluated. (Input)

CDF — Value of CDF at Y. (Output)

CDF must be declared EXTERNAL in the calling program.

X — Vector of length NOBS containing the observations. (Input)

PDIF — Vector of length 6 containing the output statistics. (Output)

I **PDIF(I)**

1 $D_n =$ Maximum of

$$(D_n^+, D_n^-)$$

2 $D_n^+ =$ Maximum difference between the theoretical and empirical CDF's

3 $D_n^- =$ Maximum difference between the empirical and theoretical CDF's

4 $Z = \sqrt{\text{NOBS}} * (\text{PDIF}(1)).$

5 Probability of the statistic exceeding D_n under the null hypothesis of equality and against the one-sided alternative. An exact probability is computed for $\text{NOBS} \leq 80$, and an approximate probability is computed for $\text{NOBS} > 80$. See function AKSLDF (see Chapter 17, Probability Distribution Functions and Inverses).

6 Probability of the statistic exceeding D_n under the null hypothesis of equality and against the two-sided alternative. This probability is twice the probability reported in PDIF(5), (or 1.0 if $2 * \text{PDIF}(5)$ is greater than 1.0). This approximation is nearly exact when PDIF(5) is less than 0.10.

Optional Arguments

NOBS — Number of observations. (Input)
Default: NOBS = size (X,1).

NMISS — Number of missing (NaN, not a number) values. (Output)

FORTRAN 90 Interface

Generic: CALL KSONE (CDF, X, PDIF [,...])

Specific: The specific interface names are S_KSONE and D_KSONE.

FORTRAN 77 Interface

Single: CALL KSONE (CDF, NOBS, X, PDIF, NMISS)

Double: The double precision name is DKSONE.

Example

In this example, a random sample of size 100 is generated via routine RNUN (see Chapter 18, Random Number Generation) for the uniform (0, 1) distribution. We want to test the null hypothesis that the CDF is the standard normal distribution with a mean of 0.5 and a variance equal to the uniform (0, 1) variance (1/12).

```
USE RNSET_INT
USE RNUN_INT
USE KSONE_INT
USE UMACH_INT

INTEGER ISEED, NOBS
PARAMETER (ISEED=123457, NOBS=100)
!
INTEGER NMISS, NOUT
REAL CDF, PDIF(6), X(100)
EXTERNAL CDF
!
!                                     Generate the sample
CALL RNSET (ISEED)
CALL RNUN (X)
!
CALL KSONE (CDF, X, PDIF, NMISS=NMISS)
!
CALL UMACH (2, NOUT)
WRITE (NOUT,99999) NMISS, PDIF
99999 FORMAT ('NMISS = ', I4/' D+      = ', F8.4/' D+      = ', F8.4/ &
' D-      = ', F8.4/' Z      = ', F8.4/' Prob greater D', &
' one-sided = ', F8.4/' Prob greater D two-sided = ', &
F8.4)
END
!
!                                     The CDF
!
REAL FUNCTION CDF (X)
REAL X
!
REAL AMEAN, STD
PARAMETER (AMEAN=0.50, STD=0.2886751)
```



```

!
REAL          ANORDF, Z
EXTERNAL     ANORDF
!
!                               Standardize
Z = (X-AMEAN)/STD
!
!                               Get the probability
CDF = ANORDF(Z)
!
RETURN
END

```

Output

```

NMISS =      0
D      =    0.1471
D+     =    0.0810
D-     =    0.1471
Z      =    1.4708
Prob greater D one-sided =    0.0132
Prob greater D two-sided =    0.0264

```

Comments

1. Workspace may be explicitly provided, if desired, by use of `K2ONE/DK2ONE`. The reference is:

```
CALL K2ONE (CDF, NOBS, X, PDIF, NMISS, XWK)
```

The additional argument is:

XWK — Work vector of length $3 * (\text{NOBS} + 1)$ if $\text{NOBS} \leq 80$, or of length NOBS if $\text{NOBS} > 80$.

2. Informational errors

Type Code

4	2	<code>PDIF</code> , the output cumulative distribution value from <code>CDF</code> , must be greater than or equal to 0.0 and less than or equal to 1.0 (by definition of a probability distribution function).
4	3	At least one tie is detected in <code>X</code> . Ties are not allowed in <code>KSONE</code> .
4	4	<code>PDIF</code> , the output cumulative distribution value from <code>CDF</code> , cannot decrease with increasing <code>X</code> (by the definition of a cumulative distribution function).
4	6	All the elements of <code>X</code> are missing (NaN, not a number) values.

3. No check is made for the validity of the input data. Thus, although one or more of the `X(I)` may be inconsistent with the distribution in that an observation may be outside of the range of the distribution, `KSONE` will not detect the anomaly (unless the user causes it to be detected via the function `CDF`).

Description

The routine `KSONE` performs a Kolmogorov-Smirnov goodness-of-fit test in one sample. The hypotheses tested follow:

- $H_0 : F(x) = F^*(x) \quad H_1 : F(x) \neq F^*(x)$
- $H_0 : F(x) \geq F^*(x) \quad H_1 : F(x) < F^*(x)$
- $H_0 : F(x) \leq F^*(x) \quad H_1 : F(x) > F^*(x)$

where F is the cumulative distribution function (CDF) of the random variable, and the theoretical CDF, F^* , is specified via the user-supplied `FUNCTION CDF`. Let $n = \text{NOBS} - \text{NMISS}$. The test statistics for both one-sided alternatives

$$D_n^+ = \text{PDIF}(2)$$

and

$$D_n^- = \text{PDIF}(3)$$

and the two-sided ($D_n = \text{PDIF}(1)$) alternative are computed as well as an asymptotic z -score ($\text{PDIF}(4)$) and p -values associated with the one-sided ($\text{PDIF}(5)$) and two-sided ($\text{PDIF}(6)$) hypotheses. For $n > 80$, asymptotic p -values are used (see Gibbons 1971). For $n \leq 80$, exact one-sided p -values are computed according to a method given by Conover (1980, page 350). An approximate two-sided test p -value is obtained as twice the one-sided p -value. The approximation is very close for one-sided p -values less than 0.10 and becomes very bad as the one-sided p -values get larger.

Programming Notes

1. The theoretical CDF is assumed to be continuous. If the CDF is not continuous, the statistics

$$D_n^*$$

will not be computed correctly.

2. Estimation of parameters in the theoretical CDF from the sample data will tend to make the p -values associated with the test statistics too liberal. The empirical CDF will tend to be closer to the theoretical CDF than it should be.
3. No attempt is made to check that all points in the sample are in the support of the theoretical CDF. If all sample points are not in the support of the CDF, the null hypothesis must be rejected.
4. The user must supply an external `FUNCTION` that calculates the theoretical CDF for a given abscissa. The calling program must contain an `EXTERNAL` statement with the name of this routine. Often, IMSL functions in Chapter 17, "Probability Distribution Functions and Inverses," may be used. Examples of possible user-supplied routines follow. Each FORTRAN function would be preceded by the statement

REAL FUNCTION CDF(X)

and ended by a RETURN and an END statement.

- a. Normal (μ, σ^2) $Z = (X - \mu)/\sigma$
CDF = ANORDF(Z)
- b. Uniform[a, b] If (X .LT. a) THEN
CDF = 0.0
ELSE IF (X .GT. b) THEN
CDF = 1.0
ELSE
CDF = (X - a)/(b - a)
END IF
- c. Minimum of n CDF = 1.0 - (1.0 - X)**n
Uniform(0, 1) random numbers

CHIGF

Performs a chi-squared goodness-of-fit test.

Required Arguments

CDF — User-supplied FUNCTION to compute the cumulative distribution function (CDF) at a given value. The form is CDF(Y), where

Y — Value at which the CDF is to be evaluated. (Input)

CDF — Value of the CDF at Y. (Output)

CDF must be declared EXTERNAL in the calling program.

NELM — The absolute value of NELM is the number of data elements currently input in X. (Input)

NELM may be positive, zero, or negative. Negative NELM means delete the -NELM data elements from the analysis.

X — Vector of length |NELM| containing the data elements for this call. (Input)

If the data element is missing (NaN, not a number), then the observation is ignored.

NCAT — The absolute value of NCAT is the number of cells into which the observations are to be tallied. (Input)

If NCAT is negative, then CHIGF chooses the cutpoints in CUTP so that the cells are equiprobable in continuous distributions. NCAT should not be negative in discrete distributions. The user must be careful to define cutpoints in discrete distributions since no error message can be generated in this situation if NCAT is negative.

RNGE — Vector of length 2 containing the lower and upper endpoints of the range of the distribution, respectively. (Input)

If the lower and upper endpoints are equal, a range on the whole real line is used. If the

lower and upper endpoints are different, points outside of the range are ignored so that distributions conditional on the range can be used. In this case, the point `RNGE(1)` is excluded from the first interval, but the point `RNGE(2)` is included in the last interval.

NDFEST — Number of parameters estimated in computing the CDF. (Input)

CUTP — Vector of length $|\text{NCAT}| - 1$ containing the cutpoints defining the cells. (Input, if `NCAT` is positive, output, otherwise)

$|\text{NCAT}| - 1$ cutpoints define the cells to be used. If `NCAT` is positive, then the cutpoints are input by the user. The intervals defined by the cutpoints are such that the lower endpoint is not included while the upper endpoint is included in the interval.

P — p -value for the chi-squared statistic in `CHISQ(|NCAT| + 1)`. (Output)

This chi-squared statistic has `DF` degrees of freedom.

Optional Arguments

IDO — Processing option. (Input)

Default: `IDO = 0`.

IDO Action

- 0 This is the only call to `CHIGF`, and all of the data are input on this call.
- 1 This is the first call to `CHIGF`, and additional calls to `CHIGF` will be made. Initialization and updating for the data in `X` are performed.
- 2 This is an intermediate call to `CHIGF`. Updating for the data in `X` is performed.
- 3 This is the final call to `CHIGF`. Updating for the data in `X` and wrap-up computations are performed.

Calls to `CHIGF` with `IDO = 2` or `3` may be intermixed. It is permissible for a call with `IDO = 2` to follow a call with `IDO = 3`.

FRQ — Vector containing the frequencies. (Input)

If the first element of `FRQ` is `-1.0`, then all frequencies are taken to be 1 and `FRQ` is of length 1. Otherwise, `FRQ` is of length $|\text{NELM}|$, and the elements in `FRQ` contain the frequency of the corresponding observation in `X`. If the frequency is missing (NaN, not a number) (and `FRQ(1)` is not `-1.0`), the observation is ignored.

Default: `FRQ(1) = -1.0`.

COUNTS — Vector of length $|\text{NCAT}|$ containing the counts in each of the cells. (Output, if `IDO = 0` or 1; input/output, if `IDO > 1`)

EXPECT — Vector of length $|\text{NCAT}|$ containing the expected count in each cell. (Output, if `IDO = 0` or 3; not referenced otherwise)

CHISQ — Vector of length $|NCAT| + 1$ containing the contributions to chi-squared. (Output, if $IDO = 0$ or 3, not referenced otherwise)
 Elements 1 through $|NCAT|$ contain the contributions to chi-squared for the corresponding cell. Element $|NCAT| + 1$ contains the total chi-squared statistic.

DF — Degrees of freedom in chi-squared. (Output)

FORTRAN 90 Interface

Generic: `CALL CHIGF (CDF, NELM, X, NCAT, RNGE, NDFEST, CUTP, P [,...])`

Specific: The specific interface names are `S_CHIGF` and `D_CHIGF`.

FORTRAN 77 Interface

Single: `CALL CHIGF (IDO, CDF, NELM, X, FRQ, NCAT, RNGE, NDFEST, CUTP, COUNTS, EXPECT, CHISQ, P, DF)`

Double: The double precision name is `DCHIGF`.

Example 1

In this example, a discrete binomial random sample of size 1000 with binomial parameter $p = 0.3$ and binomial sample size 5 is generated via routine `RNBIN` (see Chapter 18, Random Number Generation). routine `RNSET` is first used to set the seed. One call to `CHIGF` is made. Routine `BINDF` (see Chapter 17, Probability Distribution Functions and Inverses) is used to compute the CDF.

```

USE IMSL_LIBRARIES
INTEGER ISEED, NCAT, NDFEST, NELM
PARAMETER (ISEED=123457, NCAT=6, NDFEST=0, NELM=1000)
!
INTEGER I, IX(NELM), NOUT
REAL CDF, CHISQ(NCAT+1), COUNTS(NCAT), CUTP(NCAT-1), DF, &
      EXPECT(NCAT), P, RNGE(2), X(NELM)
EXTERNAL CDF
!
DATA RNGE/0.0, 0.0/
DATA CUTP/.5, 1.5, 2.5, 3.5, 4.5/
!
CALL RNSET (ISEED)
!
                                Generate the data
CALL RNBIN (5, 0.3, IX)
DO 10 I=1, NELM
    X(I) = IX(I)
10 CONTINUE
!
CALL CHIGF (CDF, NELM, X, NCAT, RNGE, NDFEST, CUTP, P, &
            COUNTS=COUNTS, EXPECT=EXPECT, CHISQ=CHISQ, DF=DF)
!
                                Print results
CALL WRRRN ('Counts', COUNTS, 1, NCAT, 1)

```

```

CALL WRRRN ('Expect', EXPECT, 1, NCAT, 1)
CALL WRRRN ('Contributions to Chi-squared', CHISQ, 1, NCAT, 1)
CALL UMACH (2, NOUT)
WRITE (NOUT,99999) CHISQ(NCAT+1), P, DF
99999 FORMAT (///'0Chi-squared      ', F8.4, /, ' P-value          ' &
, F8.4, /, ' Degrees of freedom', F8.4)
END
!
REAL FUNCTION CDF (Y)
REAL      Y
!
INTEGER   I
REAL      BINDF
EXTERNAL  BINDF
!
I      = Y
CDF = BINDF(I,5,0.3)
RETURN
END

```

Output

*** WARNING ERROR 7 from CHIGF. An expected value is less than 5.

Counts					
1	2	3	4	5	6
170.0	331.0	320.0	148.0	28.0	3.0

Expect					
1	2	3	4	5	6
168.1	360.2	308.7	132.3	28.3	2.4

Contributions to Chi-squared					
1	2	3	4	5	6
0.022	2.359	0.414	1.863	0.004	0.134

Chi-squared	4.7963
P-value	0.4412
Degrees of freedom	5.0000

Comments

Informational errors

Type Code

- | | | |
|---|----|--|
| 4 | 4 | There are more observations deleted from a cell than added. |
| 4 | 5 | All observations are missing. |
| 3 | 6 | An expected value is less than 1. |
| 3 | 7 | An expected value is less than 5. |
| 4 | 8 | The function CDF is not a cumulative distribution function. |
| 4 | 9 | The probability of the range of the distribution is not positive. |
| 4 | 10 | An error has occurred when inverting the cumulative distribution function. This function must be continuous and defined over |

the whole real line. If all else fails, you must specify the cutpoints (i.e., `NCAT` must be positive).

Description

Routine `CHIGF` performs a chi-squared goodness-of-fit test that a random sample of observations is distributed according to a specified theoretical cumulative distribution. The theoretical distribution, which may be continuous, discrete, or a mixture of discrete and continuous distributions, is specified via a user-defined `FUNCTION`. Because the user is allowed to specify a range for the observations, a test that is conditional upon the specified range is performed.

`|NCAT|` gives the number of intervals into which the observations are to be divided. These intervals can be specified via the vector `CUTP`, which contains the cutpoints (or endpoints) for the intervals. Or if `NCAT` is negative, equiprobable intervals computed by `CHIGF` can be used. Regardless of the method used to obtain them, the intervals are such that the lower endpoint is not included in the interval while the upper endpoint is always included. The user should determine the cutpoints when the cumulative distribution function has discrete elements since `CHIGF` cannot determine them in this case. Regardless of how the cutpoints are determined, the lower endpoint of the first interval is specified by `RNGE(1)` when `RNGE(1) ≠ RNGE(2)` and is given as minus machine infinity otherwise. The upper endpoint of the last interval is defined similarly.

Routine `CHIGF` tallies the observations in X as follows. If the cutpoints are determined by `CHIGF`, then the cumulative probability at x_i , $F(x_i)$, is computed via function `CDF`. The tally for x_i is made in interval number $\lfloor mF(x) + 1 \rfloor$, where $m = |\text{NCAT}|$ and $\lfloor \cdot \rfloor$ is the function that takes the greatest integer that is no larger than the argument of the function. If the cutpoints are specified by the user, the tally is made in the interval to which x_i belongs using the endpoints specified by the user. Thus, if the computer time required to calculate the cumulative distribution function is large, user-specified cutpoints may be preferred in order to reduce the total computing time.

If the expected count in any cell is less than 1, then a rule of thumb is that the chi-squared approximation may be suspect. A warning message to this effect is issued in this case, as well as when an expected value is less than 5.

Programming Notes

The user must supply a function `CDF` with calling sequence `CDF(Y)`, which returns the value of the cumulative distribution function at any point Y in the range of the distribution. The supplied function must be declared in an `EXTERNAL` statement in the calling program. Many of the IMSL cumulative distribution functions in Chapter 17, “Probability Distribution Functions and Inverses,” can be used for `CDF`, either directly, if the calling sequence is correct, or indirectly, if, for example, the sample means and standard deviations are to be used in computing the theoretical `CDF`.

Additional Example

Example 2

This example illustrates the use of CHIGF on a randomly generated sample from the normal distribution. One thousand randomly generated observations are tallied into 10 equiprobable intervals. Twelve calls to CHIGF are made. The first call is solely for initialization since IDO = 1 and NROW = 0. The next 10 calls tally the data, 100 observations at a time, with IDO = 2 and NROW = 100. The last call is for wrap up only since IDO = 3 and NROW = 0. All twelve calls could have been replaced with one call to CHIGF with IDO = 0 and NROW = 1000. X would need to be of length 1000 if one call were used. In this example, the null hypothesis is not rejected.

```
USE IMSL_LIBRARIES
INTEGER ISEED, NCAT, NDFEST
PARAMETER (ISEED=123457, NCAT=-10, NDFEST=0)
!
INTEGER I, IDO, NOUT, NELM
REAL CHISQ(-NCAT+1), COUNTS(-NCAT), CUTP(-NCAT-1), &
      DF, EXPECT(-NCAT), P, RNGE(2), X(100)
!
DATA RNGE/0.0, 0.0/
!
CALL RNSET (ISEED)
!
!                               Initialization
IDO = 1
NELM = 0
CALL CHIGF (S_ANORDF, NELM, X, NCAT, RNGE, NDFEST, CUTP, P, &
           IDO=IDO, COUNTS=COUNTS, EXPECT=EXPECT, &
           CHISQ=CHISQ, DF=DF)
!
!                               Add the data
IDO = 2
NELM = 100
DO 10 I=1, 10
    CALL RNNOR (X)
    CALL CHIGF (S_ANORDF, NELM, X, NCAT, RNGE, NDFEST, CUTP, P, &
              IDO=IDO, COUNTS=COUNTS, EXPECT=EXPECT, &
              CHISQ=CHISQ, DF=DF)
10 CONTINUE
!
!                               Wrap up
IDO = 3
NELM = 0
CALL CHIGF (S_ANORDF, NELM, X, NCAT, RNGE, NDFEST, CUTP, &
           P, IDO=IDO, COUNTS=COUNTS, EXPECT=EXPECT, &
           CHISQ=CHISQ, DF=DF)
!
!                               Print results
CALL WRRRN ('Cutpoints', CUTP, 1, -NCAT, 1)
CALL WRRRN ('Counts', COUNTS, 1, -NCAT, 1)
CALL WRRRN ('Expect', EXPECT, 1, -NCAT, 1)
CALL WRRRN ('Contributions to Chi-squared', CHISQ, 1, -NCAT, 1)
CALL UMACH (2, NOUT)
WRITE (NOUT,99999) CHISQ(-NCAT+1), P, DF
99999 FORMAT (///'0Chi-squared          ', F8.4, /, ' P-value          ' &
            , F8.4, /, ' Degrees of freedom', F8.4)
END
```


Output

										Cutpoints									
1		2		3		4		5		6		7		8		9			
-1.282		-0.842		-0.524		-0.253		0.000		0.253		0.524		0.842		1.282			
										Counts									
1		2		3		4		5		6		7		8		9		10	
106.0		109.0		89.0		92.0		83.0		87.0		110.0		104.0		121.0		99.0	
										Expect									
1		2		3		4		5		6		7		8		9		10	
100.0		100.0		100.0		100.0		100.0		100.0		100.0		100.0		100.0		100.0	
										Contributions to Chi-squared									
1		2		3		4		5		6		7		8		9		10	
0.360		0.810		1.210		0.640		2.890		1.690		1.000		0.160		4.410		0.010	

Chi-squared 13.1806
P-value 0.1546
Degrees of freedom 9.0000

SPWLK

Performs a Shapiro-Wilk W -test for normality.

Required Arguments

X — Vector of length `NOBS` containing the observations. (Input)

W — Shapiro Wilk W statistic. (Output)

P — P -value for a test of normality. (Output)

Optional Arguments

$NOBS$ — Number of observations. (Input)
 `NOBS` must be in the range from 3 to 2000 inclusive.
 Default: `NOBS = size(x,1)`.

$NMISS$ — Number of missing observations. (Output)

FORTRAN 90 Interface

Generic: `CALL SPWLK (X, W, P [, ...])`

Specific: The specific interface names are `S_SPWLK` and `D_SPWLK`.

FORTRAN 77 Interface

Single: `CALL SPWLK (NOBS, X, W, P, NMISS)`

Double: The double precision name is DSPWLK.

Example

The following example is taken from Conover (1980, pages 364 and 195). The data consists of 50 two digit numbers taken from a telephone book. The W test fails to reject the null hypothesis of normality at the .05 level of significance.

```
USE SPWLK_INT
USE UMACH_INT

INTEGER      NMISS, NOBS
PARAMETER   (NOBS=50)
REAL        P, W, X(NOBS)

!
DATA X/23, 36, 54, 61, 73, 23, 37, 54, 61, 73, 24, 40, 56, 62, &
      74, 27, 42, 57, 63, 75, 29, 43, 57, 64, 77, 31, 43, 58, 65, &
      81, 32, 44, 58, 66, 87, 33, 45, 58, 68, 89, 33, 48, 58, 68, &
      93, 35, 48, 59, 70, 97/

!
CALL SPWLK (X, W, P, NMISS=NMISS)
!
                                Write out results
CALL UMACH(2, NOUT)
WRITE(NOUT,5) W, P, NMISS
5 FORMAT(/ ' W      = ', F6.4 / ' P      = ', F6.4 / &
        ' NMISS = ', I3)
END
```

Output

```
W      = 0.9642
P      = 0.2309
NMISS  = 0
```

Comments

1. Workspace may be explicitly provided, if desired, by use of S2WLK/DS2WLK. The reference is:

```
CALL S2WLK (NOBS, X, W, P, NMISS, WK)
```

The additional argument is:

WK — Work vector of length NOBS. If X is not needed, then WK and X can share the same storage locations. On output, WK will contain the sorted nonmissing elements of X. If X is sorted, WK is not used.

2. Informational errors
Type Code

4	2	There are too many missing (NaN, “not a number”) values in X for the test to be performed.
3	3	All observations in X are tied.

Description

Routine `SPWLK` computes the Shapiro-Wilk W -statistic for testing for normality. This test is thought to be one of the best omnibus tests of normality (see D'Agostino and Stevens 1986, page 406). Routine `SPWLK` is based upon the approximations and code given by Royston (1982a, b, c). It may be used in samples as large as 2000, or as small as 3. In the Shapiro and Wilk test, W is given by.

$$W = \left\{ \sum_{i=1}^n a_i x_{(i)} \right\}^2 / \sum_{i=1}^n (x_i - \bar{x})^2$$

where $x_{(i)}$ is the i -th largest order statistic,

$$\bar{x}$$

is the sample mean, and n is the number of observations. Royston (1982) gives approximations and tabled values which may be used to compute the coefficients a_i , $i = 1, K, n$, and obtain the significance level of the W statistic.

LILLF

Performs Lilliefors test for an exponential or normal distribution.

Required Arguments

X — Vector of length `NOBS` containing the observations. (Input)

XMEAN — Sample mean. (Output)

STD — Sample standard deviation. (Output)

DIF — Maximum absolute difference between the empirical and the theoretical distributions. (Output)

PROB — Approximate probability of a greater `DIF`. (Output)

Probabilities less than 0.01 are reported as 0.01. Probabilities greater than 0.15 for the exponential distribution or greater than 0.10 for the normal distribution are reported as 0.5. Otherwise an approximate probability is computed.

NMISS — Number of missing (NaN, not a number) values. (Output)

Optional Arguments

NOBS — Number of observations. (Input)

`NOBS` must be greater than 4.

Default: `NOBS = size(x,1)`.

IPDF — Distribution option. (Input)

`IPDF = 0` means a test for normality is to be performed. `IPDF = 1` means a test for the

exponential distribution is to be performed.
 Default: IPDF = 0.

FORTRAN 90 Interface

Generic: CALL LILLF (X, XMEAN, STD, DIF, PROB, NMISS [,...])

Specific: The specific interface names are S_LILLF and D_LILLF.

FORTRAN 77 Interface

Single: CALL LILLF (NOBS, X, IPDF, XMEAN, STD, DIF, PROB, NMISS)

Double: The double precision name is DLILLF.

Example

The following example is taken from Conover (1980, page 358). It consists of 50 observations drawn at random from a telephone book. In this example, the null hypothesis is accepted. Note that the computed probability is outside the range (0.01, 0.10), and has thus been set to .50. Because many observations in X are tied, a warning message is issued. The printing of this message can be turned off through the use of routine ERSET (Reference Material).

```

USE LILLF_INT
USE UMACH_INT

INTEGER      NOBS
PARAMETER   (NOBS=50)

!
INTEGER      NMISS, NOUT
REAL         DIF, PROB, STD, X(NOBS), XMEAN
!
DATA X/23, 23, 24, 27, 29, 31, 32, 33, 33, 35, 36, 37, 40, 42, &
      43, 43, 44, 45, 48, 48, 54, 54, 56, 57, 58, 57, 58, 58, 58, &
      59, 61, 61, 62, 63, 64, 65, 66, 68, 68, 70, 73, 73, 74, 75, &
      77, 81, 87, 89, 93, 97/
!
CALL LILLF (X, XMEAN, STD, DIF, PROB, NMISS)
!
CALL UMACH (2, NOUT)
WRITE (NOUT, '( " XMEAN = ', F9.2, ', /, " STD = ', F12.3, ', /, ' // &
      ' " DIF = ', F13.4, ', /, " PROB = ', F12.4, ', /, ' // &
      ' " NMISS = ', I6)') XMEAN, STD, DIF, PROB, NMISS
END

```

Output

```

*** WARNING  ERROR 3 from L4LLF.  Two or more elements in X are tied.
Here is a traceback of subprogram calls in reverse order:
Routine name          Error type  Error code
-----
L4LLF                 6          3      (Called internally)
L3LLF                 0          0      (Called internally)

```

```

L2LLF          0          0      (Called internally)
L1LLF          0          0
USER           0          0
XMEAN =       55.04
STD =         19.005
DIF =          0.0811
PROB =         0.5000
NMISS =        0

```

Comments

1. Workspace may be explicitly provided, if desired, by use of L2LLF/DL2LLF. The reference is:

```
CALL L2LLF (NOBS, X, IPDF, XMEAN, STD, DIF, PROB, NMISS, XWK)
```

The additional argument is:

XWK — Work vector of length NOBS.

2. Informational errors

Type Code

1	1	The computed probability of DIF is greater than 0.15 for an exponential distribution. PROB is set to 0.50.
1	2	The computed probability of DIF is less than the tabled probability of 0.01. PROB is set to 0.01.
1	3	The computed probability of DIF is greater than 0.10 for a normal distribution. PROB is set to 0.50.
1	4	The computed probability of DIF is less than 0.01. PROB is set to 0.01.
4	5	A negative value is encountered in X when IPDF = 1. Negative values are impossible for exponential distributions.
4	6	All elements in X are tied.

Description

Routine L1LLF computes Lilliefors test and its p -values for either a normal distribution in which both the mean and variance are estimated, or an exponential distribution in which the mean is estimated. Routine L1LLF uses a modified version of IMSL routine KSONE (page 634) to compute the one-sample two-sided Kolmogorov-Smirnov statistic D (DIF). p -values are then computed for the exponential distribution via linear interpolation on the tabled values given by Stephens (1974). For the normal distribution, p -values are computed using an analytic approximation given by Dallal and Wilkinson (1986). Because Stephens' (1974) tables are in the inclusive range (0.01, 0.15) and Dallal and Wilkinson (1986) give approximations in the range (0.01, 0.10), if the computed probability of a greater D is less than 0.01, a level 1 message is issued (such messages are not generally printed, see the Reference Material) and the probability is set to 0.01. Similarly, if the probability is greater than 0.15 (0.10 for the normal), a level 1 message is issued and the p -value is set to 0.50. Note that because parameters are estimated, p -values in Lilliefors test are not the same as in the Kolmogorov-Smirnov test.

Observations from exponential or normal distributions should not be tied. If tied observations are found, an informational message is printed. Printing of this message can be turned off via a call to routine `ERSET` as is discussed in the Reference Material.

A general reference for Lilliefors test is Conover (1980). The original reference for the test for normality is Lilliefors (1967), while Lilliefors (1969) introduces the test for the exponential distribution.

MVMMT

Computes Mardia's multivariate measures of skewness and kurtosis and test for multivariate normality.

Required Arguments

- X** — `NOBS` by `NVAR`+ `m` matrix containing the data. (Input)
`m` is 0, 1, or 2 depending upon whether any columns in `X` contain frequencies or weights.
- IND** — Vector of length `NVAR` containing the column numbers in `X` for which statistics are desired. (Input)
- STAT** — Vector of length 13 containing the output statistics. (Output)
If a statistic is not computed, the corresponding element of `STAT` is set to not a number (NaN).
- `STAT(1)` = estimated skewness.
- `STAT(2)` = expected skewness assuming a multivariate normal distribution.
- `STAT(3)` = asymptotic chi-squared statistic assuming a multivariate normal distribution.
- `STAT(4)` = probability of a greater chi-squared.
- `STAT(5)` = Mardia and Foster's standard normal score for skewness.
- `STAT(6)` = estimated kurtosis.
- `STAT(7)` = expected kurtosis assuming a multivariate normal distribution.
- `STAT(8)` = asymptotic standard error of the estimated kurtosis.
- `STAT(9)` = standard normal score obtained from `STAT(6)` through `STAT(8)`.
- `STAT(10)` = *p*-value corresponding to `STAT(9)`.
- `STAT(11)` = Mardia and Foster's standard normal score for kurtosis.

STAT(12) = Mardia's S_W statistic based upon STAT(5) and STAT(11).

STAT(13) = p -value for STAT(12).

STAT(12) and STAT(13) are only computed when ICMPUT = 0.

Optional Arguments

NOBS — Number of rows of data in X . (Input)

Default: NOBS = size (X ,1).

NVAR — Dimensionality of the multivariate space for which the skewness and kurtosis are to be computed. (Input)

Default: NVAR = size (IND,1).

NCOL — Number of columns in matrix X . (Input)

Default: NCOL = size (X ,2).

LDX — Leading dimension of X exactly as specified in the dimension statement in the calling program. (Input)

Default: LDX = size (X ,1).

IFRQ — Frequency option. (Input)

IFRQ = 0 means that all frequencies are 1.0. Positive IFRQ indicates that column number IFRQ of X contains the frequencies. All frequencies should be integer values. The NINT (nearest integer) function is used to obtain integer frequencies if this is not the case.

Default: IFRQ = 0.

IWT — Weighting option. (Input)

IWT = 0 means that all weights are 1.0. Positive IWT means that column IWT of X contains the weights. Negative weights are not allowed.

Default: IWT = 0.

ICMPUT — Option parameter giving the statistics to compute. (Input)

Default: ICMPUT = 0.

ICMPUT Output Statistics

0 Both skewness and kurtosis.

1 Kurtosis only.

2 Skewness only.

NI — The sum of the frequencies of all observations used in the computations. (Output)

SWT — The sum of the weights times the frequencies for all observations used in the computations. (Output)

XMEAN — Vector of length *NVAR* containing the sample means. (Output)

R — *NVAR* by *NVAR* upper triangular matrix containing the Cholesky $R^T R$ factorization of the covariance matrix. (Output)

LDR — Leading dimension of *R* exactly as specified in the dimension statement in the calling program. (Input)

NRMISS — Number of rows of data in *X* containing any missing values (NaN, not a number). (Output)
Rows with missing values in the columns *IND*, *IFRQ*, and *IWT* are excluded from the analysis.

FORTRAN 90 Interface

Generic: CALL MVMMT (X, IND, STAT [,...])

Specific: The specific interface names are S_MVMMT and D_MVMMT.

FORTRAN 77 Interface

Single: CALL MVMMT (NOBS, NVAR, NCOL, X, LDX, IND, IFRQ, IWT,
 ICMPUT, NI, SWT, XMEAN, R, LDR, STAT, NRMISS)

Double: The double precision name is DMVMMT.

Example

In the following example, 150 observations from a 5 dimensional standard normal distribution are generated via routine *RNNOR* (see Chapter 18, Random Number Generation). The skewness and kurtosis statistics are then computed for these observations.

```
USE IMSL_LIBRARIES
INTEGER    LDR, LDX, NCOL, NVAR, I
PARAMETER (NCOL=5, LDX=150, NVAR=NCOL, LDR=NVAR)
!
INTEGER    IND(5), NI, NOUT, NRMISS
REAL      R(LDR,NVAR), STAT(13), SWT, X(LDX,NCOL), XMEAN(NVAR)
!
DATA IND/1, 2, 3, 4, 5/
!
CALL RNSET (123457)
DO 10 I=1, NCOL
CALL RNNOR(X(:, I))
10 CONTINUE
!
```



```

CALL MVMMT (X, IND, STAT, NI=NI, SWT=SWT, XMEAN=XMEAN, R=R, &
            NRMIS=NRMISS)
!
CALL UMACH (2, NOUT)
WRITE (NOUT,*) ' NI = ', NI, ' SWT = ', SWT, ' NRMIS = ', NRMIS
CALL WRRRN ('XMEAN', XMEAN, 1, NVAR, 1)
CALL WRRRN ('R', R)
CALL WRRRN ('STAT', STAT, 1, 13, 1)
!
END

```

Output

NI = 150 SWT = 150.0 NRMIS = 0

XMEAN				
1	2	3	4	5
0.0355	0.0467	0.0599	0.0957	0.1007

R					
	1	2	3	4	5
1	1.033	-0.022	-0.037	0.055	-0.003
2	0.000	0.993	-0.119	-0.076	-0.056
3	0.000	0.000	0.997	-0.089	0.017
4	0.000	0.000	0.000	1.008	-0.040
5	0.000	0.000	0.000	0.000	1.027

STAT									
1	2	3	4	5	6	7	8	9	10
1.52	1.36	38.71	0.31	0.42	34.21	34.54	1.27	-0.26	0.80
11	12	13							
0.18	0.21	0.90							

Comments

1. Workspace may be explicitly provided, if desired, by use of M2MMT/DM2MMT. The reference is:

```

CALL M2MMT (NOBS, NVAR, NCOL, X, LDX, IND, IFRQ, IWT, ICMPUT,
            NI, SWT, XMEAN, R, LDR, STAT, NRMIS, D, OB, CC)

```

The additional arguments are as follows:

D — Work vector of length NVAR.

OB — Work vector of length NVAR.

CC — Work vector of length m , where $m = \text{NVAR} * \text{NVAR}$ if ICMPUT = 1 or $m = \text{NVAR} * \text{NVAR} * \text{NVAR}$ otherwise.

2. Informational errors

Type Code

- | | | |
|---|---|--|
| 4 | 1 | At least one of the variables in x is linearly related to the other variables in x . |
| 4 | 2 | The sum of the frequencies must be greater than the maximum of 3 and the number of variables plus one. |

Description

Routine `MVMMT` computes Mardia's (1970) measures $b_{1,p}$ and $b_{2,p}$ of multivariate skewness and kurtosis, respectfully, for $p = \text{NVAR}$. These measures are then used in computing tests for multivariate normality. Three test statistics, one based upon $b_{1,p}$ alone, one based upon $b_{2,p}$ alone, and an omnibus test statistic formed by combining normal scores obtained from $b_{1,p}$ and $b_{2,p}$ are computed. On the order of np^3 , operations are required in computing $b_{1,p}$ when the method of Isogai (1983) is used, where $n = \text{NOBS}$. On the order of np^2 , operations are required in computing $b_{2,p}$.

Let

$$d_{ij} = \sqrt{w_i w_j} (x_i - \bar{x})^T S^{-1} (x_j - \bar{x})$$

where

$$S = \frac{\sum_{i=1}^n w_i f_i (x_i - \bar{x})(x_i - \bar{x})^T}{\sum_{i=1}^n f_i}$$

$$\bar{x} = \frac{1}{\sum_{i=1}^n w_i f_i} \sum_{i=1}^n w_i f_i x_i$$

f_i is the frequency of the i -th observation, and w_i is the weight for this observation. (Weights w_i are defined such that x_i is distributed according to a multivariate normal, $N(\mu, \Sigma/w_i)$ distribution, where Σ is the covariance matrix.) Mardia's multivariate skewness statistic is defined as:

$$b_{1,p} = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n f_i f_j d_{ij}^3$$

while Mardia's kurtosis is given as:

$$b_{2,p} = \frac{1}{n} \sum_{i=1}^n f_i d_{ii}^2$$

Both measures are invariant under the affine (matrix) transformation $AX + D$, and reduce to the univariate measures when $p = \text{NVAR} = 1$. Using formulas given in Mardia and Foster (1983), the approximate expected value, asymptotic standard error, and asymptotic p -value for $b_{2,p}$, and the approximate expected value, an asymptotic chi-squared statistic, and p -value for the $b_{1,p}$ statistic are computed. These statistics are all computed under the null hypothesis of a multivariate normal distribution. In addition, standard normal scores $W_1(b_{1,p})$ and $W_2(b_{2,p})$ (different from but similar to the asymptotic normal and chi-squared statistics above) are computed. These scores are combined into an asymptotic chi-squared statistic with two degrees of freedom:

$$S_W = W_1^2(b_{1,p}) + W_2^2(b_{2,p})$$

This chi-squared statistic may be used to test for multivariate normality. A p -value for the chi-squared statistic is also computed.

KSTWO

Performs a Kolmogorov-Smirnov two-sample test.

Required Arguments

X — Vector of length `NOBSX` containing the observations in sample one. (Input)

Y — Vector of length `NOBSY` containing the observations in sample two. (Input)

PDIF— Vector of length 6 containing the output statistics. (Output)

I **PDIF(I)**

1 D_{mn} = Maximum of the absolute values of

$$D_{mn}^+ \text{ and } D_{mn}^-$$

2 D_{mn}^+ = Maximum difference between the empirical cumulative distribution function (CDF) of X minus the empirical CDF of Y .

3 D_{mn}^- = Maximum difference between the empirical CDF of X minus the empirical CDF of Y . (The maximum of the negative differences.)

4 Z = Standardized value of D_{mn} . A two-sample approximation with no correction for continuity is used.

5 One-sided probability of a larger D_{mn} under the null hypothesis of equal distributions.

6 Two-sided probability of exceeding D_{mn} under the null hypothesis of equal distributions.

Optional Arguments

NOBSX— Number of observations in sample one. (Input)

Default: `NOBSX = size(x,1)`.

NOBSY — Number of observations in sample two. (Input)
Default: NOBSY = size (Y,1).

NMISSX — Number of missing observations in the X sample. (Output)

NMISSY — Number of missing observations in the Y sample. (Output)

FORTRAN 90 Interface

Generic: CALL KSTWO (X, Y, PDIF [,...])

Specific: The specific interface names are S_KSTWO and D_KSTWO.

FORTRAN 77 Interface

Single: CALL KSTWO (NOBSX, X, NOBSY, Y, PDIF, NMISSX, NMISSY)

Double: The double precision name is DKSTWO.

Example

The following example illustrates the KSTWO routine with two randomly generated samples from a uniform(0,1) distribution. Since the two theoretical distributions are identical, we would not expect to reject the null hypothesis.

```
USE RNSET_INT
USE RNUN_INT
USE KSTWO_INT
USE UMACH_INT

INTEGER ISEED, NOBSX, NOBSY, NMISSX, NMISSY
PARAMETER (ISEED=123457, NOBSX=100, NOBSY=60)
REAL X(NOBSX), Y(NOBSY), PDIF(6)
!                                     Generate the sample
CALL RNSET(ISEED)
CALL RNUN (X)
CALL RNUN (Y)
!
CALL KSTWO (X, Y, PDIF, NMISSX=NMISSX, NMISSY=NMISSY)
!
CALL UMACH(2, NOUT)
WRITE(NOUT, 5) PDIF, NMISSX, NMISSY
5 FORMAT(' D      = ', F8.4 / ' D+    = ', F8.4 / ' D-    = ', F8.4 / &
' Z      = ', F8.4 / ' Prob greater D one sided = ', F8.4 / &
' Prob greater D two sided = ', F8.4 / &
' Missing X = ', I3 / ' Missing Y = ', I3)
END
```

Output

```
D      = 0.1800
D+    = 0.1800
D-    = 0.0100
```

```

Z      = 1.1023
Prob greater D one sided = 0.0720
Prob greater D two sided = 0.1440
Missing X = 0
Missing Y = 0

```

Comments

Workspace may be explicitly provided, if desired, by use of `K2TWO/DK2TWO`. The reference is:

```
CALL K2TWO (NOBSX, X, NOBSY, Y, PDIF, NMISSX, NMISSY, XWK, YWK)
```

The additional arguments are as follows:

XWK — Work vector of length `NOBSX + 1`.

YWK — Work vector of length `NOBSY + 1`.

Description

Routine `KSTWO` computes Kolmogorov-Smirnov two-sample test statistics for testing that two continuous cumulative distribution functions (CDF's) are identical based upon two random samples. One- or two-sided alternatives are allowed. Exact p -values are computed for the two-sided test when `NOBSX * NOBSY` is less than 104.

Let $F_n(x)$ denote the empirical CDF in the X sample, let $G_m(y)$ denote the empirical CDF in the Y sample, where $n = \text{NOBSX} - \text{NMISSX}$ and $m = \text{NOBSY} - \text{NMISSY}$, and let the corresponding population distribution functions be denoted by $F(x)$ and $G(y)$, respectively. Then, the hypotheses tested by `KSTWO` are as follows:

- $H_0 : F(x) = G(x)$ $H_1 : F(x) \neq G(x)$
- $H_0 : F(x) \leq G(x)$ $H_1 : F(x) > G(x)$
- $H_0 : F(x) \geq G(x)$ $H_1 : F(x) < G(x)$

The test statistics are given as follows:

$$D_{mn} = \max(D_{mn}^+, D_{mn}^-) \quad (\text{PDIF}(1))$$

$$D_{mn}^+ = \max_x (F_n(x) - G_m(x)) \quad (\text{PDIF}(2))$$

$$D_{mn}^- = \max_x (G_m(x) - F_n(x)) \quad (\text{PDIF}(3))$$

Asymptotically, the distribution of the statistic

$$Z = D_{mn} \sqrt{(m+n)/(mn)}$$

(returned in `PDIF(4)`) converges to a distribution given by Smirnov (1939).

Exact probabilities for the two-sided test are computed when nm is less than or equal to 10^4 , according to an algorithm given by Kim and Jennrich (1973), and computed here via function `AKS2DF` (see Chapter 17, Probability Distribution Functions and Inverses). When nm is greater than 10^4 , the very good approximations given by Kim and Jennrich are used to obtain the two-

sided p -values. The one-sided probability is taken as one half the two-sided probability. This is a very good approximation when the p -value is small (say, less than 0.10) and not very good for large p -values.

RUNS

Performs a runs up test.

Required Arguments

X — Vector of length `NRAN` containing the data elements to be added to the test on this invocation. (Input)

COUNT — Vector of length `NRUN` containing the counts of the number of runs up of each length. (Output, if `IDO = 0` or `1`; Input/Output, if `IDO = 2` or `3`)

EXPECT — Vector of length `NRUN` containing the expected number of runs of each length. (Output, if `IDO = 0` or `3`; not referenced otherwise)

COVAR — `NRUN` by `NRUN` matrix containing the variances and covariances of the counts (Output, if `IDO = 0` or `3`; not referenced otherwise)

CHISQ — Chi-squared statistic for testing the null hypothesis of a uniform distribution. (Output, if `IDO = 0` or `3`; not referenced otherwise)

DF — Degrees of freedom for chi-squared. (Output, if `IDO = 0` or `3`; not referenced otherwise)

PROB — Probability of a larger chi-squared. (Output, if `IDO = 0` or `3`; not referenced otherwise)

Optional Arguments

IDO — Processing option. (Input)
Default: `IDO = 0`.

<code>IDO</code>	Action
------------------	--------

0	This is the only invocation of <code>RUNS</code> , and all the data are input at once.
---	--

1	This is the first invocation of <code>RUNS</code> , and additional calls will be made. Initialization and updating for the <code>NRAN</code> data elements are performed.
---	---

2	This is an intermediate invocation of <code>RUNS</code> , and updating for the <code>NRAN</code> data elements is performed.
---	--

IDO Action

- 3 This is the final invocation of `RUNS` for this data. Updating for the `NRAN` data elements is performed, followed by the wrap-up computations.

NRAN— Number of data points currently input in `X`. (Input)

`NRAN` may be positive or zero on any invocation of `RUNS`.

Default: `NRAN = size (X,1)`.

NRUN— Length of the longest run for which tabulation is desired. (Input)

Runs of length 1, 2, `K`, `NRUN - 1` are counted in `COUNT(1) - COUNT(NRUN - 1)`.

`COUNT(NRUN)` contains the number of runs of length `NRUN` or greater. `NRUN` must be greater than or equal to one.

Default: `NRUN = size (COUNT,1)`.

LDCOVA— Leading dimension of `COVAR` exactly as specified in the dimension statement in the calling program. (Input)

Default: `LDCOVA = size (COVAR,1)`.

FORTRAN 90 Interface

Generic: `CALL RUNS (X, COUNT, EXPECT, COVAR, CHISQ, DF, PROB [,...])`

Specific: The specific interface names are `S_RUNS` and `D_RUNS`.

FORTRAN 77 Interface

Single: `CALL RUNS (IDO, NRAN, X, NRUN, COUNT, EXPECT, COVAR, LDCOVA, CHISQ, DF, PROB)`

Double: The double precision name is `DRUNS`.

Example

The following example illustrates the use of the runs test on 10^4 pseudo-random uniform deviates. In the example, 2000 deviates are generated for each call to `RUNS`. The `IDO` parameter is set to 1 on the first call to `RUNS`, 2 on the second, third, and fourth calls, and 3 on the last call. Since the probability of a larger chi-squared statistic is 0.1872, there is no strong evidence to support rejection of this null hypothesis of randomness.

```
USE IMSL_LIBRARIES
INTEGER LDCOVA, NRAN, NRUN
PARAMETER (LDCOVA=6, NRAN=2000, NRUN=6)
!
INTEGER I, IDO, NOUT
REAL CHISQ, COUNT(NRUN), COVAR(LDCOVA,NRUN), DF, &
    EXPECT(NRUN), PROB, X(NRAN)
!
CALL RNSET (123457)
```

```

!
DO 10 I=1, 5
!
      Set IDO
      IF (I .EQ. 1) THEN
        IDO = 1
      ELSE IF (I .EQ. 5) THEN
        IDO = 3
      ELSE
        IDO = 2
      END IF
!
      Generate the random numbers
      CALL RNUN (X)
!
      CALL RUNS (X, COUNT, EXPECT, COVAR, CHISQ, DF, PROB, IDO=IDO)
10 CONTINUE
!
      CALL WRRRN ('COUNT', COUNT, 1, NRUN, 1)
      CALL WRRRN ('EXPECT', EXPECT, 1, NRUN, 1)
      CALL WRRRN ('COVAR', COVAR)
      CALL UMACH (2, NOUT)
      WRITE (NOUT,*) ' CHISQ = ', CHISQ
      WRITE (NOUT,*) ' DF = ', DF
      WRITE (NOUT,*) ' PROB = ', PROB
      END

```

Output

		COUNT					
		1	2	3	4	5	6
1709.0	2046.0	953.0	260.0	55.0	4.0		

		EXPECT					
		1	2	3	4	5	6
1667.3	2083.4	916.5	263.8	57.5	11.9		

		COVAR					
		1	2	3	4	5	6
1	1278.2	-194.6	-148.9	-71.6	-22.9	-6.7	
2	-194.6	1410.1	-490.6	-197.2	-55.2	-14.4	
3	-148.9	-490.6	601.4	-117.4	-31.2	-7.8	
4	-71.6	-197.2	-117.4	222.1	-10.8	-2.6	
5	-22.9	-55.2	-31.2	-10.8	54.8	-0.6	
6	-6.7	-14.4	-7.8	-2.6	-0.6	11.7	

CHISQ =	8.76514
DF =	6.00000
PROB =	0.187225

Comments

1. Workspace may be explicitly provided, if desired, by use of R2NS/DR2NS. The reference is:

```

CALL R2NS (IDO, NRAN, X, NRUN, COUNT, EXPECT, COVAR, LDCOVA,
CHISQ, DF, PROB, RWK, CWK, LRUN, NOBS, XLAST)

```


The additional arguments are as follows:

RWK — Work vector of length `NRUN`.

CWK — Work vector of length `NRUN2`.

LRUN — Scalar used to keep track of number of last runs. (Output, if `IDO = 0` or `1`; input/output, otherwise)
`LRUN` should not be changed between calls with the same data set.

NOBS — Scalar used to keep track of total number of observations. (Output, if `IDO = 0` or `1`; input/output, otherwise)
`NOBS` should not be changed between calls with the same data set.

XLAST — Scalar used to keep track of last run. (Output, if `IDO = 0` or `1`; input/output, otherwise)
`XLAST` should not be changed between calls with the same data set.

2. Informational errors

Type Code

- | | | |
|---|---|--|
| 3 | 1 | At least one tie is detected in <code>x</code> . |
| 4 | 2 | The covariance matrix of the runs score is not positive definite. Use a smaller value of <code>NRUN</code> . |

Description

Routine `RUNS` computes statistics for the runs up test. Runs tests are used to test for cyclical trend in sequences of random numbers. Routine `RUNS` may be called once (`IDO = 0`) or several times (`IDO = 1, 2, and 3`). If all of the data will not fit into memory, the second mode of operation must be used. If the data fit into memory, then the first mode of operation is slightly more efficient. If the runs down test is desired, each observation should first be multiplied by `-1` to change its sign, and `RUNS` called with the modified vector of observations.

Routine `RUNS` first tallies the number of runs up (increasing sequences) of each desired length. For $i = 1, K, r - 1$, where $r = \text{NRUN}$, `COUNT(i)` contains the number of runs of length i . `COUNT(NRUN)` contains the number of runs of length `NRUN` or greater. As an example of how runs are counted, the sequence (1, 2, 3, 1) contains 1 run up of length 3, and one run up of length 1.

After tallying the number of runs up of each length, `RUNS` computes the expected values and the covariances of the counts according to methods given by Knuth (1981, pages 65–67). Let R denote a vector of length `NRUN` containing the number of runs of each length so that the i -th element of R , r_i , contains the count of the runs of length i . Let Σ_R denote the covariance matrix of R under the null hypothesis of randomness, and let μ_R denote the vector of expected values for R under this null hypothesis. Then, an approximate chi-squared statistic with `NRUN` degrees of freedom is given as

$$\chi^2 = (R - \mu_R)^T \Sigma_R^{-1} (R - \mu_R)$$

In general, the larger the value of each element of μ_R , the better the chi-squared approximation.

PAIRS

Performs a pairs test.

Required Arguments

X — Vector of length `NRAN` containing the data elements to be added to the test on this invocation. (Input)

LAG — The lag to be used in computing the pairs statistic. (Input)
Pairs $(X(i), X(i + LAG))$ for $i = 1, K, N - LAG$ are tabulated, where N is the total sample size.

COUNT — `NCELL` by `NCELL` matrix containing the count of the number of pairs in each cell. (Output, if `IDO = 0` or `1`; input/output, if `IDO = 2` or `3`)

EXPECT — Expected number of counts in each cell. (Output, if `IDO = 0` or `3`; not referenced otherwise)

CHISQ — Chi-squared statistic for testing the null hypothesis of a uniform distribution. (Output, if `IDO = 0` or `3`; not referenced otherwise)

DF — Degrees of freedom for chi-squared. (Output, if `IDO = 0` or `3`; not referenced otherwise)

PROB — Probability of a larger chi-squared. (Output, if `IDO = 0` or `3`; not referenced otherwise)

Optional Arguments

IDO — Processing option. (Input)
Default: `IDO = 0`.

IDO	Action
------------	---------------

- | | |
|---|---|
| 0 | This is the only invocation of <code>PAIRS</code> , and all the data are input at once. |
| 1 | This is the first invocation of <code>PAIRS</code> , and additional calls will be made. Initialization and updating for the <code>N_{RAN}</code> data elements are performed. |
| 2 | This is an intermediate invocation of <code>PAIRS</code> , and updating for the <code>N_{RAN}</code> data elements is performed. |
| 3 | This is the final invocation of <code>PAIRS</code> . Updating for the <code>N_{RAN}</code> data elements is performed, followed by the wrap-up computations. |

NRAN— Number of random deviates currently input in *x*. (Input)

NRAN may be positive or zero on any invocation of *PAIRS*.

Default: *NRAN* = size (*x*,1).

NCELL— Number of equiprobable cells on each axis into which the pairs statistics are to be tabulated. (Input)

Default: *NCELL* = size (*COUNT*,1).

LDCOUN— Leading dimension of *COUNT* exactly as specified in the dimension statement of the calling program. (Input)

Default: *LDCOUN* = size (*COUNT*,1).

FORTRAN 90 Interface

Generic: CALL *PAIRS* (*x*, *LAG*, *COUNT*, *EXPECT*, *CHISQ*, *DF*, *PROB* [,...])

Specific: The specific interface names are *S_PAIRS* and *D_PAIRS*.

FORTRAN 77 Interface

Single: CALL *PAIRS* (*IDO*, *NRAN*, *x*, *NCELL*, *LAG*, *COUNT*, *LDCOUN*,
 EXPECT, *CHISQ*, *DF*, *PROB*)

Double: The double precision name is *DPAIRS*.

Example

The following example illustrates the calculations of the *PAIRS* statistics when a random sample of size 10^4 is used and the *LAG* is 1. The results are not significant. On each call to *PAIRS*, 2000 random deviates are processed. On the first call, initialization is also performed, while on the fifth call the wrap-up computations are performed. Routine *RNUN* (see Chapter 18, Random Number Generation) is used in obtaining the pseudorandom deviates.

```
USE IMSL_LIBRARIES
INTEGER    LAG, LDCOUN, NCELL, NOBS
PARAMETER  (LAG=5, LDCOUN=10, NCELL=10, NOBS=2000)
!
INTEGER    I, IDO, NOUT
REAL       CHISQ, COUNT(LDCOUN,NCELL), DF, EXPECT, PROB, X(NOBS)
!
CALL RNSET (123467)
!
DO 10 I=1, 5
  CALL RNUN (X)
  IF (I .EQ. 1) THEN
    IDO = 1
  ELSE IF (I .EQ. 5) THEN
    IDO = 3
  ELSE
    IDO = 2
  END IF
```

```

        CALL PAIRS (X, LAG, COUNT, EXPECT, CHISQ, DF, PROB, IDO=IDO)
10 CONTINUE
    CALL UMACH (2, NOUT)
    CALL WRRRN ('COUNT', COUNT)
    WRITE(NOUT, '(' Expect = ', F12.2, /, ' Chi-squared = ', F12.2, &
&         ' DF = ', F12.0, /, ' PROBABILITY = ', F12.4)') &
        EXPECT, CHISQ, DF, PROB
    END

```

Output

	COUNT								
	1	2	3	4	5	6	7	8	9
1	111.0	82.0	95.0	117.0	102.0	102.0	112.0	84.0	90.0
2	104.0	106.0	109.0	108.0	101.0	97.0	102.0	92.0	109.0
3	88.0	111.0	86.0	105.0	112.0	79.0	103.0	105.0	106.0
4	91.0	110.0	108.0	92.0	88.0	108.0	113.0	93.0	105.0
5	104.0	105.0	103.0	104.0	101.0	94.0	96.0	86.0	93.0
6	98.0	104.0	103.0	104.0	79.0	89.0	92.0	104.0	92.0
7	103.0	91.0	97.0	101.0	116.0	83.0	117.0	118.0	106.0
8	105.0	105.0	110.0	91.0	92.0	82.0	100.0	104.0	110.0
9	92.0	102.0	82.0	101.0	93.0	128.0	101.0	109.0	125.0
10	79.0	99.0	103.0	97.0	104.0	101.0	93.0	93.0	98.0

```

10
1 73.0
2 88.0
3 99.0
4 114.0
5 103.0
6 99.0
7 99.0
8 89.0
9 98.0
10 105.0

```

```

Expect =          99.75
Chi-squared =      104.31 DF =          99.
Probability =      0.3379

```

Comments

Informational errors

Type Code

- | | | |
|---|---|--|
| 3 | 1 | For better efficiency, it is recommended that N _{RAN} be at least twice as large as LAG |
| 4 | 2 | The sum of the counts is zero. All output statistics are set to NaN (not a number). |

Description

Routine PAIRS computes the pairs test (or the Good's serial test) on a hypothesized sequence of uniform (0,1) pseudorandom numbers. The test proceeds as follows. Subsequent pairs

$(X(i), X(i + \text{LAG}))$ are tallied into a $k \times k$ matrix, where $k = \text{NCELL}$. In this tally, element (j, m) of the matrix is incremented, where

$$j = \lfloor kX(i) \rfloor + 1$$

$$m = \lfloor kX(i + l) \rfloor + 1$$

where $l = \text{LAG}$, and the notation $\lfloor \cdot \rfloor$ represents the greatest integer function, $\lfloor Y \rfloor$ is the greatest integer less than or equal to Y , where Y is a real number. If $l = 1$, then $i = 1, 3, 5, \dots, n - 1$. If $l > 1$, then $i = 1, 2, 3, \dots, n - l$, where n is the total number of pseudorandom numbers input on the current invocation of `PAIRS` (i.e., $n = \text{NRAN}$).

Given the tally matrix in `COUNT`, chi-squared is computed as

$$\chi^2 = \sum_{i,j=1}^k \frac{(o_{ij} - e)^2}{e}$$

where $e = \sum o_{ij}/k^2$, and o_{ij} is the observed count in cell (i, j) ($o_{ij} = \text{COUNT}(i, j)$).

Because pair statistics for the trailing observations are not tallied on any call, the user should call `PAIRS` with `NRAN` as large as possible. For `LAG` < 20 and `NRAN` = 2000, little power is lost.

DSQAR

Performs a d^2 test.

Required Arguments

X — Vector of length `NRAN` containing the data elements to be added to the test on this invocation. (Input)

COUNT — Vector of length `NCELL` containing the count of the number of d^2 values in each cell. (Output, if `IDO` = 0 or 1. Input/Output, if `IDO` = 2 or 3.)

EXPECT — The expected number of counts in each cell. (Output, if `IDO` = 0 or 3; not referenced otherwise)

CHISQ — Chi-squared statistic for testing the null hypothesis of a uniform distribution. (Output, if `IDO` = 0 or 3; not referenced otherwise)

DF — Degrees of freedom for chi-squared. (Output, if `IDO` = 0 or 3; not referenced otherwise)

PROB — Probability of a larger chi-squared. (Output, if `IDO` = 0 or 3; not referenced otherwise)

Optional Arguments

IDO — Processing Option. (Input)

Default: IDO = 0.

IDO Action

- 0 This is the only invocation of DSQAR, and all the data are input at once.
- 1 This is the first invocation of DSQAR, and additional calls will be made. Initialization and updating for the NRAN data elements are performed.
- 2 This is an intermediate invocation of DSQAR, and updating for the NRAN data elements is performed.
- 3 This is the final invocation of DSQAR for this data set. Updating for the NRAN data elements is performed, followed by the wrap-up computations.

NRAN — Number of data elements currently input in X. (Input)

NRAN may be positive or zero on any invocation of DSQAR.

Default: NRAN = size (X,1).

NCELL — The number of equiprobable cells into which the d^2 statistics are to be tabulated.

(Input)

Default: NCELL = size (COUNT,1).

FORTRAN 90 Interface

Generic: CALL DSQAR (X, COUNT, EXPECT, CHISQ, DF, PROB [,...])

Specific: The specific interface names are S_DSQAR and D_DSQAR.

FORTRAN 77 Interface

Single: CALL DSQAR (IDO, NRAN, X, NCELL, COUNT, EXPECT, CHISQ, DF, PROB)

Double: The double precision name is DDSQAR.

Example

In the following example, 2000 observations generated via routine RNUN (see Chapter 18, Random Number Generation) are input to DSQAR in one call. In the example, the null hypothesis of a uniform distribution is not rejected.

```
USE IMSL_LIBRARIES
INTEGER IDO, NCELL, NROW
PARAMETER (NCELL=6, NROW=2000)
```

!

```

INTEGER      NOUT
REAL         CHISQ, COUNT(NCELL), DF, EXPECT, PROB, X(NROW)
!
CALL RNSET (123457)
!
!                               Generate the random numbers
CALL RNUN (X)
!
CALL DSQAR (X, COUNT, EXPECT, CHISQ, DF, PROB)
!
CALL WRRRN ('COUNT', COUNT, 1, NCELL, 1)
CALL UMACH (2, NOUT)
WRITE (NOUT,*) ' EXPECT = ', EXPECT
WRITE (NOUT,*) ' CHISQ = ', CHISQ
WRITE (NOUT,*) ' DF = ', DF
WRITE (NOUT,*) ' PROB = ', PROB
END

```

Output

```

                COUNT
      1         2         3         4         5         6
87.00    84.00    78.00    76.00    92.00    83.00
EXPECT =          83.3333
CHISQ   =          2.056
DF      =           5.0
PROB    =          0.841343

```

Comments

Informational errors

Type Code

- | | | |
|---|---|---|
| 3 | 1 | The expected value of a each cell is less than 5. The chi-squared approximation may not be good. |
| 4 | 2 | The sum of the counts is equal to zero. There are no data elements so the chi-squared statistic cannot be computed. |

Description

Routine DSQAR computes the d^2 test for succeeding quadruples of hypothesized pseudorandom uniform (0, 1) deviates. The d^2 test is performed as follows. Let X_1, X_2, X_3 , and X_4 denote four pseudorandom uniform deviates, and consider

$$D^2 = (X_3 - X_1)^2 + (X_4 - X_2)^2$$

The probability distribution of D^2 is given as

$$\Pr(D^2 \leq d^2) = d^2\pi - \frac{8d^3}{3} + \frac{d^4}{2}$$

when $D^2 \leq 1$, where π denotes the value of pi. If $D^2 > 1$, this probability is given as

$$\Pr(D^2 \leq d^2) = \frac{1}{3} + (\pi - 2)d^2 + 4\sqrt{d^2 - 1} + 8\frac{(d^2 - 1)^{\frac{3}{2}}}{3} - \frac{d^4}{2} - 4d^2 \arctan\left(\frac{\sqrt{1 - \frac{1}{d^2}}}{\frac{1}{d}}\right)$$

See Gruenberger and Mark (1951) for a derivation of this distribution.

For each succeeding set of 4 pseudorandom uniform numbers input in x , d^2 and the cumulative probability of d^2 ($\Pr(D^2 \leq d^2)$) are computed. The resulting probability is tallied into one of $k = \text{NCELL}$ equally spaced intervals.

Let n denote the number of sets of four random numbers input ($n =$ the total number of observations/4). Then, under the null hypothesis that the numbers input are random uniform (0, 1) numbers, the expected value for each element in `COUNT` is $e = n/k$. An approximate chi-squared statistic is computed as

$$\chi^2 = \sum_{i=1}^k \frac{(o_i - e)^2}{e}$$

where $o_i = \text{COUNT}(i)$ is the observed count. Thus, χ^2 has $k - 1$ degrees of freedom, and the null hypothesis of pseudorandom uniform (0, 1) deviates is rejected if χ^2 is too large. As n increases, the chi-squared approximation becomes better. A useful generalization is that $e > 5$ yields a good chi-squared approximation.

DCUBE

Performs a triplets test.

Required Arguments

X — Vector of length `NRAN` containing the data elements to be added to the test on this invocation. (Input)

COUNT — `NCELL` by `NCELL` by `NCELL` array containing the tabulations for the triplets test. (Output, if `IDO = 0` or 1. Input/Output, if `IDO = 2` or 3.)

EXPECT — Expected number of counts in each cell. (Output, if `IDO = 0` or 3; not referenced otherwise)

CHISQ — Chi-squared statistic for testing the null hypothesis of a uniform distribution. (Output, if `IDO = 0` or 3; not referenced otherwise)

DF — Degrees of freedom for chi-squared. (Output, if `IDO = 0` or 3; not referenced otherwise)

PROB — Probability of a larger chi-squared. (Output, if `IDO = 0` or `3`; not referenced otherwise)

Optional Arguments

IDO — Processing Option. (Input)
Default: `IDO = 0`.

IDO Action

- 0 This is the only invocation of `DCUBE`, and all the data are input at once.
- 1 This is the first invocation of `DCUBE`, and additional calls will be made. Initialization and updating for the `NRAN` data elements are performed.
- 2 This is an intermediate invocation of `DCUBE`, and updating for the `NRAN` data elements is performed.
- 3 This is the final invocation of `DCUBE` for this data set. Updating for the `NRAN` data elements is performed, followed by the wrap-up computations.

NRAN — Number of random deviates currently input in `X`. (Input)
`NRAN` may be positive or zero on any invocation of `DCUBE`. `NRAN` must be evenly divisible by 3.
Default: `NRAN = size (X,1)`.

NCELL — The number of equiprobable cells on each of the three axes into which the triplets are to be tabulated. (Input)
Each set of three data elements is tabulated into a three dimensional cube, each axis of which has `NCELL` cells.
Default: `NCELL = size (COUNT,1)`.

LDCOUN — Leading and second dimension of matrix `COUNT` exactly as specified in the dimension statement in the calling program. (Input)
Default: `LDCOUN = size (COUNT,1)`.

FORTRAN 90 Interface

Generic: `CALL DCUBE (X, COUNT, EXPECT, CHISQ, DF, PROB [...])`

Specific: The specific interface names are `S_DCUBE` and `D_DCUBE`.

FORTRAN 77 Interface

Single: `CALL DCUBE (IDO, NRAN, X, NCELL, COUNT, LDCOUN, EXPECT, CHISQ, DF, PROB)`

Double: The double precision name is `DDCUBE`.

Example

In the following example, 2001 deviates generated by IMSL routine `RNUN` (see Chapter 18, Random Number Generation) are input to `DCUBE`, and tabulated in 27 equally sized cubes. In the example, the null hypothesis is not rejected.

```
USE IMSL_LIBRARIES
INTEGER   LDCOUN, NCELL, NRAN
PARAMETER (LDCOUN=3, NCELL=3, NRAN=2001)
!
INTEGER   I, NOUT
REAL      CHISQ, COUNT(LDCOUN,LDCOUN,NCELL), DF, EXPECT, PROB, &
          X(NRAN)
!
CALL RNSET (123457)
!                                     Generate the random numbers
CALL RNUN (X)
!
CALL DCUBE (X, COUNT, EXPECT, CHISQ, DF, PROB)
!
DO 10 I=1, NCELL
    CALL WRRRN ('COUNT', COUNT(1:,1:,I))
10 CONTINUE
CALL UMACH (2, NOUT)
WRITE (NOUT,*) ' EXPECT = ', EXPECT
WRITE (NOUT,*) ' CHISQ  = ', CHISQ
WRITE (NOUT,*) ' DF     = ', DF
WRITE (NOUT,*) ' PROB   = ', PROB
END
```

Output

```
          COUNT
          1      2      3
1  26.00  27.00  24.00
2  20.00  17.00  32.00
3  30.00  18.00  21.00
```

```
          COUNT
          1      2      3
1  20.00  16.00  26.00
2  22.00  22.00  27.00
3  30.00  24.00  26.00
```

```
          COUNT
          1      2      3
1  28.00  30.00  22.00
2  23.00  24.00  22.00
3  33.00  30.00  27.00
```

```
EXPECT =    24.7037
CHISQ  =    21.7631
DF     =    26.0000
PROB   =    0.701586
```

Comments

Informational error

Type Code

4 1

The sum of the counts is equal to zero. There are no data elements so the chi-squared statistic cannot be computed. CHISQ and PROB are set to NaN (not a number).

Description

Routine DCUBE computes the triplets test on a sequence of hypothesized pseudorandom uniform(0, 1) deviates. The triplets test is computed as follows: Each set of three successive deviates, X_1 , X_2 , and X_3 , is tallied into one of m^3 equal sized cubes, where $m = \text{NCELL}$. Let $i = [mX_1] + 1$, $j = [mX_2] + 1$, and $k = [mX_3] + 1$. For the triplet (X_1, X_2, X_3) , $\text{COUNT}(i, j, k)$ is incremented.

Under the null hypothesis of pseudorandom uniform(0, 1) deviates, the m^3 cells are equally probable and each has expected value $e = n/m^3$, where n is the number of triplets tallied. An approximate chi-squared statistic is computed as

$$\chi^2 = \sum_{i,j,k=1}^m \frac{(o_{ijk} - e)^2}{e}$$

where $o_{ijk} = \text{COUNT}(i, j, k)$.

The computed chi-squared has $m^3 - 1$ degrees of freedom, and the null hypothesis of pseudorandom uniform (0, 1) deviates is rejected if χ^2 is too large.

Appendix A: GAMS Index

Description

This index lists routines in STAT/LIBRARY by a tree-structured classification scheme known as GAMS. Boisvert, Howe, Kahaner, and Springmann (1990) give the GAMS classification scheme. The classification scheme given here is Version 2.0.

The first level of the full classification scheme is denoted by a letter A thru Z as follows:

- A. Arithmetic, Error Analysis
- B. Number Theory
- C. Elementary and Special Functions
- D. Linear Algebra
- E. Interpolation
- F. Solution of Nonlinear Equations
- G. Optimization
- H. Differentiation and Integration
- I. Differential and Integral Equations
- J. Integral Transforms
- K. Approximation
- L. Statistics, Probability
- M. Simulation, Stochastic Modeling
- N. Data Handling
- O. Symbolic Computation
- P. Computational Geometry
- Q. Graphics
- R. Service Routines
- S. Software Development Tools
- Z. Other

There are seven levels in the classification scheme. Classes in the first level are identified by a capital letter as is given above. Classes in the remaining levels are identified by alternating letter-and-number combinations. A single letter (a–z) is used with the odd-numbered levels. A number (1–26) is used within the even-numbered levels.

IMSL STAT/LIBRARY

C.....ELEMENTARY AND SPECIAL FUNCTIONS (*search also class L5*)

- C3Polynomials
 - C3aOrthogonal
 - OPOLY Generate orthogonal polynomials with respect to x values and specified weights.
- C7Gamma
 - C7eIncomplete gamma
 - CHIDF Evaluate the chi-squared distribution function.
 - CHIIN Evaluate the inverse of the chi-squared distribution function.
 - GAMDF Evaluate the gamma distribution function.
 - GAMIN Evaluate the inverse of the gamma distribution function.
 - C7fIncomplete gamma
 - BETDF Evaluate the beta probability distribution function.
 - BETIN Evaluate the inverse of the beta distribution function.
- C8Error functions
 - C8aError functions, their inverses, integrals, including the normal distribution function
 - ANORDF Evaluate the standard normal (Gaussian) distribution function.
 - ANORIN Evaluate the inverse of the standard normal (Gaussian) distribution function.
- KAPPROXIMATION (*search also class L8*)
 - K1Least squares (L_2) approximation
 - K1aLinear least squares (*search also classes D5, D6, D9*)
 - K1a1Unconstrained
 - RCOV Fit a multiple linear regression model given the variance-covariance matrix.
 - RGIVN Fit a multivariate linear regression model via fast Givens transformations.
 - RGLM Fit a multivariate general linear model.
 - RLSE Fit a multiple linear regression model using least squares.
 - K1a1a ...Univariate data (curve fitting)
 - K1a1a2 ..Polynomials
 - RCURV Fit a polynomial curve using least squares.
 - RFORP Fit an orthogonal polynomial regression model.
 - RPOLY Analyze a polynomial regression model.
 - K1a2Constrained
 - K1a2a ...Linear constraints
 - RLEQU Fit a multivariate linear regression model with linear equality restrictions $HB = G$ imposed on the regression

parameters given results from IMSL routine `RGIVN` after
`IDO = 1` and `IDO = 2` and prior to `IDO = 3`.

K1b..... Nonlinear least squares

K1b1..... Unconstrained

K1b1a... Smooth functions

K1b1a1. User provides no derivatives

`RNLIN` Fit a nonlinear regression model.

K1b1a2. User provides first derivatives

`RNLIN` Fit a nonlinear regression model.

K2..... Minimax (L_∞) approximation

`RLMV` Fit a multiple linear regression model using the minimax
criterion.

K3..... Least absolute value (L_1) approximation

`RLLP` Fit a multiple linear regression model using the L_p norm
criterion.

K4..... Other analytic approximations (e.g., Taylor polynomial, Pade)

`RLLP` Fit a multiple linear regression model using the L_p norm
criterion.

L..... STATISTICS, PROBABILITY

L1..... Data summarization

L1a..... One-dimensional data

L1a1..... Raw data

`EQTIL` Compute empirical quantiles.

`LETTR` Produce a letter value summary.

`ORDST` Determine order statistics.

L1a1a... Location

`UVSTA` Compute basic univariate statistics.

L1a1b... Dispersion

`UVSTA` Compute basic univariate statistics.

L1a1c... Shape

`UVSTA` Compute basic univariate statistics.

L1a1e... Ties

`NTIES` Compute tie statistics for a sample of observations.

L1a3..... Grouped data

`GRPES` Compute basic statistics from grouped data.

L1c..... Multi-dimensional data

L1c1..... Raw data

- CSTAT Compute cell frequencies, cell means, and cell sums of squares for multivariate data.
- L1c1b ... Covariance, correlation
- CORVC Compute the variance-covariance or correlation matrix.
- PCORR Compute partial correlations or covariances from the covariance or correlation matrix.
- RBCOV Compute a robust estimate of a covariance matrix and mean vector.
- L2Data manipulation
- L2a Transform (*search also classes L10a, N6, and N8*)
- BCTR Perform a forward or an inverse Box-Cox (power) transformation.
- GCSCP Generate centered variables, squares, and crossproducts.
- OPOLY Generate orthogonal polynomials with respect to x values and specified weights.
- RANKS Compute the ranks, normal scores, or exponential scores for a vector of observations.
- L2b Tally
- CSTAT Compute cell frequencies, cell means, and cell sums of squares for multivariate data.
- FREQ Tally multivariate observations into a multi-way frequency table.
- OWFRQ Tally observations into a one-way frequency table.
- TWFRQ Tally observations into a two-way frequency table.
- L2e Construct new variables (e.g., indicator variables)
- GRGLM Generate regressors for a general linear model.
- L3Elementary statistical graphics (*search also class Q*)
- L3aOne-dimensional data
- L3a1Histograms
- HHSTP Print a horizontal histogram.
- VHSTP Print a vertical histogram.
- L3a2Frequency, cumulative frequency, percentile plots
- CDFP Print a sample cumulative distribution function (CDF), a theoretical CDF, and confidence band information.
- L3a3EDA graphics (e.g., box plots)
- BOXP Print boxplots for one or more samples.
- STMLP Print a stem-and-leaf plot.
- L3a4Bar charts
- HHSTP Print a horizontal histogram.
- VHSTP Print a vertical histogram.
- L3bTwo-dimensional data (*search also class L3e*)
- L3b1Histograms (superimposed and bivariate)

VHS2P Print a vertical histogram with every bar subdivided into two parts.

L3b2 Frequency, cumulative frequency
 CDF2P Print a plot of two sample cumulative distribution functions.

L3e Multi-dimensional data

L3e3 Scatter diagrams

L3e3a ... Superimposed Y vs. X
 PLOTP Print a plot of up to ten sets of points.
 SCTP Print a scatterplot of several groups of data.

L3e4 EDA
 BOXP Print boxplots for one or more samples.

L4 Elementary data analysis

L4a One-dimensional data

L4a1 Raw data

L4a1a ... Parametric analysis
 CDFP Print a sample cumulative distribution function (CDF), a theoretical CDF, and confidence band information.

L4a1a2 .Probability plots

L4a1a2e Exponential, extreme value
 PROBP Print a probability plot.

L4a1a2h Halfnormal
 PROBP Print a probability plot.

L4a1a2l Lambd α , logistic, lognormal
 PROBP Print a probability plot.

L4a1a2n Negative binomial, normal
 PROBP Print a probability plot.

L4a1a2w Weibull
 PROBP Print a probability plot.

L4a1a4 .Parameter estimates and tests

L4a1a4b Binomial
 BINES Estimate the parameter p of the binomial distribution.

L4a1a4p Poisson
 POIES Estimate the parameter of the Poisson distribution.

L4a1b ... Nonparametric analysis

L4a1b1 .Estimates and test regarding location (e.g., median), dispersion and shape

- SIGNT Perform a sign test of the hypothesis that a given value is a specified quantile of a distribution.
- SNRNK Perform a Wilcoxon signed rank test.

L4a1b2 .Density function estimation

- DESKN Perform nonparametric probability density function estimation by the kernel method.
- DESPL Perform nonparametric probability density function estimation by the penalized likelihood method.
- DESPT Estimate a probability density function at specified points using linear or cubic interpolation.
- DNFFT Compute Gaussian kernel estimates of a univariate density via the fast Fourier transform over a fixed interval.

L4a1c....Goodness-of-fit tests

- CHIGF Perform a chi-squared goodness-of-fit test.
- KSONE Perform a Kolmogorov-Smirnov one-sample test for continuous distributions.
- LILLF Perform Lilliefors test for an exponential or normal distribution.
- SPWLK Perform a Shapiro-Wilk W -test for normality.

L4aldAnalysis of a sequence of numbers (*search also class L10a*)

- DCUBE Perform a triplets test.
- DSQAR Perform a D-square test.
- NCTRD Perform the Noether test for cyclical trend.
- PAIRS Perform a pairs test.
- RUNS Perform a runs up test.
- SDPLC Perform the Cox and Stuart sign test for trends in dispersion and location.

L4a3Grouped (and/or censored) data

- GRPES Compute basic statistics from grouped data.
- NRCES Compute maximum likelihood estimates of the mean and variance from grouped and/or censored normal data.

L4a4Data sampled from a finite population

- SMPPR Compute statistics for inferences regarding the population proportion and total, given proportion data from a simple random sample.
- SMPPS Compute statistics for inferences regarding the population proportion and total, given proportion data from a stratified random sample.
- SMPSC Compute statistics for inferences regarding the population mean and total using single-stage cluster sampling with continuous data.
- SMPSR Compute statistics for inferences regarding the population mean and total, given data from a simple random sample.
- SM PSS Compute statistics for inferences regarding the population mean and total, given data from a stratified random sample.

- SMPST Compute statistics for inferences regarding the population mean and total, given continuous data from a two-stage sample with equisized primary units.
- L4b Two dimensional data (*search also class L4c*)
 - L4b1 Pairwise independent data
 - L4b1a ... Parametric analysis
 - L4b1a4 . Parameter estimates and hypothesis tests
 - TWOMV Compute statistics for mean and variance inferences using samples from two normal populations.
 - L4b1b ... Nonparametric analysis (e.g., tests based on ranks)
 - CNCRD Calculate and test the significance of the Kendall coefficient of concordance.
 - INCLD Perform an inclusion test.
 - KENDL Compute and test Kendall's rank correlation coefficient.
 - RNKSM Perform the Wilcoxon rank sum test.
 - L4b1c ... Goodness-of-fit tests
 - KSTWO Perform a Kolmogorov-Smirnov two-sample test.
 - L4b4 Pairwise dependent grouped data
 - CTRHO Estimate the bivariate normal correlation coefficient using a contingency table.
 - TETCC Categorize bivariate data and compute the tetrachoric correlation coefficient.
 - L4b5 Data sampled from a finite population
 - SMPRR Compute statistics for inferences regarding the population mean and total using ratio or regression estimation, or inferences regarding the population ratio, given a simple random sample.
 - SMPRS Compute statistics for inferences regarding the population mean and total using ratio or regression estimation, given continuous data from a stratified random sample.
- L4c Multi-dimensional data (*search also classes L4b and L7a1*)
 - L4c1 Independent data
 - L4c1b ... Nonparametric analysis
 - BHAKV Perform a Bhapkar V test.
 - KRSKL Perform a Kruskal-Wallis test for identical population medians.
 - KTRND Perform a k -sample trends test against ordered alternatives.
 - MVMMT Compute Mardia's multivariate measures of skewness and kurtosis and test for multivariate normality.
 - QTEST Perform a Cochran Q test for related observations.
 - L4e Multiple multi-dimensional data sets

- MVIND Compute a test for the independence of k sets of multivariate normal variables.
- L5Function evaluation (*search also class C*)
- L5aUnivariate
- L5a1Cumulative distribution functions, probability density functions
- L5a1b ...Beta, binomial
- BETDF Evaluate the beta probability distribution function.
 - BINDF Evaluate the binomial distribution function.
 - BINPR Evaluate the binomial probability function.
- L5a1c....Cauchy, chi-squared
- CHIDF Evaluate the chi-squared distribution function.
 - CSNDF Evaluate the noncentral chi-squared distribution function.
- L5a1f.... F distribution
- FDF Evaluate the F distribution function.
- L5a1g ...Gamma, general, geometric
- GAMDF Evaluate the gamma distribution function.
 - GCDF Evaluate a general continuous cumulative distribution function given ordinates of the density.
- L5a1h ...Halfnormal, hypergeometric
- HYPDF Evaluate the hypergeometric distribution function.
 - HYPPR Evaluate the hypergeometric probability function.
- L5a1k ...Kendall F statistic, Kolmogorov-Smirnov
- AKS1DF Evaluate the distribution function of the one-sided Kolmogorov-Smirnov goodness-of-fit D^+ or D^- test statistic based on continuous data for one sample.
 - AKS2DF Evaluate the distribution function of the Kolmogorov-Smirnov goodness-of-fit D test statistic based on continuous data for two samples.
 - KENDP Compute the frequency distribution of the total score in Kendall's rank correlation coefficient.
- L5a1n ...Negative binomial, normal
- ANORDF Evaluate the standard normal (Gaussian) distribution function.
- L5a1p ...Pareto, Poisson
- POIDF Evaluate the Poisson distribution function.
 - POIPR Evaluate the Poisson probability function.
- L5a1t t distribution
- TDF Evaluate the Student's t distribution function.
 - TNDF Evaluate the noncentral Student's t distribution function.
- L5a2Inverse cumulative distribution functions, sparsity functions

- L5a2b ... Beta, binomial
 BETIN Evaluate the inverse of the beta distribution function.
- L5a2c ... Cauchy, chi-squared
 CHIIN Evaluate the inverse of the chi-squared distribution function.
 CSNIN Evaluate the inverse of the noncentral chi-squared function.
- L5a2f.... *F* distribution
 FIN Evaluate the inverse of the *F* distribution function.
- L5a2g ... Gamma, general, geometric
 GAMIN Evaluate the inverse of the gamma distribution function.
 GCIN Evaluate the inverse of a general continuous cumulative distribution function given ordinates of the density.
 GFNIN Evaluate the inverse of a general continuous cumulative distribution function given in a subprogram.
- L5a2t.... *t* distribution
 TIN Evaluate the inverse of the Student's *t* distribution function.
 TNIN Evaluate the inverse of the noncentral Student's *t* distribution function.
- L5b Multivariate
- L5b1 Cumulative distribution functions, probability density functions
- L5b1n... Normal
 BNRDF Evaluate the bivariate normal distribution function.
- L6 Random number generation
- L6a Univariate
- L6a2 Beta, binomial, Boolean
 RNBET Generate pseudorandom numbers from a beta distribution.
 RNBIN Generate pseudorandom numbers from a binomial distribution.
- L6a3 Cauchy, chi-squared
 RNCHI Generate pseudorandom numbers from a chi-squared distribution.
 RNCHY Generate pseudorandom numbers from a Cauchy distribution.
- L6a5 Exponential, extreme value
 RNEXP Generate pseudorandom numbers from a standard exponential distribution.
 RNEXT Generate pseudorandom numbers from a mixture of two exponential distributions.
- L6a7 Gamma, general (continuous, discrete), geometric

- RNGAM Generate pseudorandom numbers from a standard gamma distribution.
 - RNGCS Set up table to generate pseudorandom numbers from a general continuous distribution.
 - RNGCT Generate pseudorandom numbers from a general continuous distribution.
 - RNGDA Generate pseudorandom numbers from a general discrete distribution using an alias method.
 - RNGDS Set up table to generate pseudorandom numbers from a general discrete distribution.
 - RNGDT Generate pseudorandom numbers from a general discrete distribution using a table lookup method.
 - RNGEO Generate pseudorandom numbers from a geometric distribution.
- L6a8Halfnormal, hypergeometric
- RNHYP Generate pseudorandom numbers from a hypergeometric distribution.
- L6a12 ...Lambda, logistic, lognormal
- RNLGR Generate pseudorandom numbers from a logarithmic distribution.
 - RNLNL Generate pseudorandom numbers from a lognormal distribution.
- L6a14 ...Negative binomial, normal, normal order statistics
- RNNBN Generate pseudorandom numbers from a negative binomial distribution.
 - RNNOA Generate pseudorandom numbers from a standard normal distribution using an acceptance/rejection method.
 - RNNOF Generate a pseudorandom number from a standard normal distribution.
 - RNNOR Generate pseudorandom numbers from a standard normal distribution using an inverse CDF method.
 - RNNOS Generate pseudorandom order statistics from a standard normal distribution.
- L6a16 ...Pareto, Pascal, permutations, Poisson
- RNNPP Generate pseudorandom numbers from a nonhomogeneous Poisson process.
 - RNPER Generate a pseudorandom permutation.
 - RNPOI Generate pseudorandom numbers from a Poisson distribution.
- L6a19 ...Samples, stable distribution
- RNSRI Generate a simple pseudorandom sample of indices.
 - RNSRS Generate a simple pseudorandom sample from a finite population.
 - RNSTA Generate pseudorandom numbers from a stable distribution.

- L6a20 ... *t* distribution, time series, triangular
- RNARM Generate a time series from a specified ARMA model.
 - RNNPF Generate pseudorandom numbers from a nonhomogeneous Poisson process.
 - RNSTT Generate pseudorandom numbers from a Student's *t* distribution.
 - RNTRI Generate pseudorandom numbers from a triangular distribution on the interval (0,1).
- L6a21 ... Uniform (continuous, discrete), uniform order statistics
- RNUN Generate pseudorandom numbers from a uniform (0,1) distribution.
 - RNUND Generate pseudorandom numbers from a discrete uniform distribution.
 - RNUNF Generate a pseudorandom number from a uniform (0, 1) distribution.
 - RNUNO Generate pseudorandom order statistics from a uniform (0, 1) distribution.
- L6a22 ... Von Mises
- RNVMS Generate pseudorandom numbers from a von Mises distribution.
- L6a23 ... Weibull
- RNWIB Generate pseudorandom numbers from a Weibull distribution.
- L6b Multivariate
- RNDAT Generate pseudorandom numbers from a multivariate distribution determined from a given sample.
- L6b3 Contingency table, correlation matrix
- RNCOR Generate a pseudorandom orthogonal matrix or a correlation matrix.
 - RNTAB Generate a pseudorandom two-way table.
- L6b13 ... Multinomial
- RNMTN Generate pseudorandom numbers from a multinomial distribution.
- L6b14 ... Normal
- RNMVN Generate pseudorandom numbers from a multivariate normal distribution.
- L6b15 ... Orthogonal matrix
- RNCOR Generate a pseudorandom orthogonal matrix or a correlation matrix.
- L6b21 ... Linear L-1 (least absolute value) approximation random numbers
- FAURE_INIT Shuffles Faure sequence initialization.
 - FAURE_FREE Frees the structure containing information about the Faure sequence.

- FAURE_NEXT Computes a shuffled Faure sequence.
- L6b21 ... Uniform
- RNSPH Generate pseudorandom points on a unit circle or κ -dimensional sphere.
- L6c Service routines (e.g., seed)
- RNGEF Retrieve the current value of the array used in the IMSL GFSR random number generator.
- RNGES Retrieve the current value of the table in the IMSL random number generators that use shuffling.
- RNGET Retrieve the current value of the seed used in the IMSL random number generators.
- RNISD Determine a seed that yields a stream beginning 100,000 numbers beyond the beginning of the stream yielded by a given seed used in IMSL multiplicative congruential generators (with no shufflings).
- RNOPG Retrieve the indicator of the type of uniform random number generator.
- RNOPT Select the uniform (0, 1) multiplicative congruential pseudorandom number generator.
- RNSEF Initialize the array used in the IMSL GFSR random number generator.
- RNSES Initialize the table in the IMSL random number generators that use shuffling.
- RNSET Initialize a random seed for use in the IMSL random number generators.
- L7 Analysis of variance (including analysis of covariance)
- L7a One-way
- L7a1 Parametric
- AONEC Analyze a one-way classification model with covariates.
- AONEW Analyze a one-way classification model.
- CTRST Compute contrast estimates and sums of squares.
- SCIPM Compute simultaneous confidence intervals on all pairwise differences of means.
- SNKMC Perform Student-Newman-Keuls multiple comparison test.
- L7b Two-way (*search also class L7d*)
- ATWOB Analyze a randomized block design or a two-way balanced design.
- FRDMN Perform Friedman's test for a randomized complete block design.
- MEDPL Compute a median polish of a two-way table.
- L7c Three-way (e.g., Latin squares) (*search also class L7d*)
- ALATN Analyze a Latin square design.
- L7d Multi-way

- L7d1 Balanced complete data (e.g., factorial designs)
- ABALD Analyze a balanced complete experimental design for a fixed, random, or mixed model.
 - ANEST Analyze a completely nested random model with possibly unequal numbers in the subgroups.
 - ANWAY Analyze a balanced n -way classification model with fixed effects.
 - CIDMS Compute a confidence interval on a variance component estimated as proportional to the difference in two mean squares in a balanced complete experimental design.
 - ROREX Reorder the responses from a balanced complete experimental design.
- L7d2 Balanced incomplete data (e.g., fractional factorial designs)
- ABIBD Analyze a balanced incomplete block design or a balanced lattice design.
- L7d3 General linear models (unbalanced data)
- ANEST Analyze a completely nested random model with possibly unequal numbers in the subgroups.
 - RGLM Fit a multivariate general linear model.
- L7e Multivariate
- RGLM Fit a multivariate general linear model.
- L7f Generate experimental designs
- RCOMP Generate an orthogonal central composite design.
- L8 Regression (*search also classes D5, D6, D9, G, K*)
- L8a Simple linear (e.g., $y = \beta_0 + \beta_1 x + \varepsilon$)
- L8a1 Ordinary least squares
- RONE Analyze a simple linear regression model.
- L8a1a ... Parameter estimation
- L8a1a1 . Unweighted data
- RLINE Fit a line to a set of data points using least squares.
- L8a1d ... Inference (e.g., calibration) (*search also class L8a1a*)
- RINCF Perform response control given a fitted simple linear regression model.
 - RINPF Perform inverse prediction given a fitted simple linear regression model.
- L8a2 L_p for p different from 2 (e.g., least absolute value, minimax)
- RLAV Fit a multiple linear regression model using the least absolute values criterion.
 - RLLP Fit a multiple linear regression model using the L_p norm criterion.
 - RLMV Fit a multiple linear regression model using the minimax criterion.

- L8bPolynomial (e.g., $y = \beta_0 + \beta_1x + \beta_2x^2 + \varepsilon$) (*search also class L8c*)
 - L8b1Ordinary least squares
 - L8b1a ...Degree determination
 - RFORP Fit an orthogonal polynomial regression model.
 - RPOLY Analyze a polynomial regression model.
 - L8b1b ...Parameter estimation
 - L8b1b2 .Using orthogonal polynomials
 - RCURV Fit a polynomial curve using least squares.
 - RFORP Fit an orthogonal polynomial regression model.
 - RPOLY Analyze a polynomial regression model.
 - L8b1c ...Analysis (*search also class L8b1b*)
 - RCASP Compute case statistics for a polynomial regression model given the fit based on orthogonal polynomials.
 - RPOLY Analyze a polynomial regression model.
 - RSTAP Compute summary statistics for a polynomial regression model given the fit based on orthogonal polynomials.
 - L8b1d ...Inference (*search also class L8b1b*)
 - RCASP Compute case statistics for a polynomial regression model given the fit based on orthogonal polynomials.
 - RPOLY Analyze a polynomial regression model.
 - RSTAP Compute summary statistics for a polynomial regression model given the fit based on orthogonal polynomials.
 - L8cMultiple linear (e.g., $y = \beta_0 + \beta_1x_1 + \dots + \beta_kx_k + \varepsilon$)
 - L8c1Ordinary least squares
 - L8c1a....Variable selection
 - L8c1a2..Using correlation or covariance data
 - GSWEP Perform a generalized sweep of a row of a nonnegative definite matrix.
 - RBEST Select the best multiple linear regression models.
 - RSTEP Build multiple linear regression models using forward selection, backward selection, or stepwise selection.
 - L8c1b ...Parameter estimation (*search also class L8c1a*)
 - L8c1b1 .Using raw data
 - RGIVN Fit a multivariate linear regression model via fast Givens transformations.
 - RGLM Fit a multivariate general linear model.
 - RLSE Fit a multiple linear regression model using least squares.

L8c1b2 . Using correlation data

RCOV Fit a multiple linear regression model given the variance-covariance matrix.

L8c1c ... Analysis (*search also classes L8c1a and L8c1b*)

RCASE Compute case statistics and diagnostics given data points, coefficient estimates $\hat{\beta}$, and the R matrix for a fitted general linear model.

RCOVB Compute the estimated variance-covariance matrix of the estimated regression coefficients given the R matrix.

RLOFE Compute a lack-of-fit test based on exact replicates for a fitted regression model.

RLOFN Compute a lack-of-fit test based on near replicates for a fitted regression model.

ROTIN Compute diagnostics for detection of outliers and influential data points given residuals and the R matrix for a fitted general linear model.

RSTAT Compute statistics related to a regression fit given the coefficient estimates $\hat{\beta}$ and the R matrix.

L8c1d ... Inference (*search also classes L8c1a and L8c1b*)

CESTI Construct an equivalent completely testable multivariate general linear hypothesis $HB U = G$ from a partially testable hypothesis $H_p B U = G_p$.

RCASE Compute case statistics and diagnostics given data points, coefficient estimates $\hat{\beta}$, and the R matrix for a fitted general linear model.

RHPSS Compute the matrix of sums of squares and crossproducts for the multivariate general linear hypothesis $HB U = G$ given the coefficient estimates \hat{B} and the R matrix.

RHPTE Perform tests for a multivariate general linear hypothesis $HB U = G$ given the hypothesis sums of squares and crossproducts matrix S_H and the error sums of squares and crossproducts matrix S_E .

RSTAT Compute statistics related to a regression fit given the coefficient estimates $\hat{\beta}$ and the R matrix.

L8c3 L_p for p different from 2

RLAV Fit a multiple linear regression model using the least absolute values criterion.

RLLP Fit a multiple linear regression model using the L_p norm criterion.

RLMV Fit a multiple linear regression model using the minimax criterion.

L8d Polynomial in several variables

RCOMP Generate an orthogonal central composite design.

- TCSCP Transform coefficients from a quadratic regression model generated from squares and crossproducts of centered variables to a model using uncentered variables.
- L8e Nonlinear (i.e., $y = f(X; \theta) + \varepsilon$)
- L8e1 Ordinary least squares
- L8e1b ... Parameter estimation
- RNLIN Fit a nonlinear regression model.
- L8f Simultaneous (i.e., $Y = XB + \varepsilon$)
- RCOV Fit a multiple linear regression model given the variance-covariance matrix.
- RGIVN Fit a multivariate linear regression model via fast Givens transformations.
- RGLM Fit a multivariate general linear model.
- RHPSS Compute the matrix of sums of squares and crossproducts for the multivariate general linear hypothesis $HB = G$ given the coefficient estimates \hat{B} and the R matrix.
- RHPTE Perform tests for a multivariate general linear hypothesis $HB = G$ given the hypothesis sums of squares and crossproducts matrix S_H and the error sums of squares and crossproducts matrix S_E .
- RLEQU Fit a multivariate linear regression model with linear equality restrictions $HB = G$ imposed on the regression parameters given results from IMSL routine RGIVN after $IDO = 1$ and $IDO = 2$ and prior to $IDO = 3$.
- L8i Service routines (e.g., matrix manipulation for variable selection)
- GCLAS Get the unique values of each classification variable.
- GCSCP Generate centered variables, squares, and crossproducts.
- GRGLM Generate regressors for a general linear model.
- RORDM Reorder rows and columns of a symmetric matrix.
- RSUBM Retrieve a symmetric submatrix from a symmetric matrix.
- L9 Categorical data analysis
- CTGLM Analyze categorical data using logistic, Probit, Poisson, and other generalized linear models.
- CTRAN Perform generalized Mantel-Haenszel tests in a stratified contingency table.
- L9a 2-by-2 tables
- CTTWO Perform a chi-squared analysis of a 2 by 2 contingency table.
- L9b Two-way tables (search also class L9d)
- CTCHI Perform a chi-squared analysis of a two-way contingency table.

- CTEPR Compute Fisher's exact test probability and a hybrid approximation to the Fisher exact test probability for a contingency table using the network algorithm.
 - CTPRB Compute exact probabilities in a two-way contingency table.
 - CTRHO Estimate the bivariate normal correlation coefficient using a contingency table.
 - CTWLS Perform a generalized linear least squares analysis of transformed probabilities in a two-dimensional contingency table.
 - MEDPL Compute a median polish of a two-way table.
 - TWFRQ Tally observations into a two-way frequency table.
- L9c Log-linear model
- CTASC Compute partial association statistics for log-linear models in a multidimensional contingency table.
 - CTLLN Compute model estimates and associated statistics for a hierarchical log-linear model.
 - CTPAR Compute model estimates and covariances in a fitted log-linear model.
 - CTSTP Build hierarchical log-linear models using forward selection, backward selection, or stepwise selection.
 - PRPFT Perform iterative proportional fitting of a contingency table using a loglinear model.
- L9d EDA (e.g., median polish)
- MEDPL Compute a median polish of a two-way table.
- L10 Time series analysis (*search also class J*)
- L10a Univariate
- L10a1 ... Transformations
- L10a1b .Stationarity (*search also class L8a1*)
- BCTR Perform a forward or an inverse Box-Cox (power) transformation.
- L10a1c .Filters
- L10a1c1 Difference (nonseasonal and seasonal)
- DIFF Difference a time series.
- L10a2 ... Time domain analysis
- AUTO_FPE_
MUL_AR Automatic selection and fitting of a multivariate autoregressive time series model using Akaike's Multivariate Final Prediction Error (MFPE) criteria.
 - AUTO_FPE_
UNI_AR Automatic selection and fitting of a univariate autoregressive time series model using Akaike's Final Prediction Error (FPE) criteria.

AUTO_MUL_AR	Automatic selection and fitting of a multivariate autoregressive time series model.
AUTO_UNI_AR	Automatic selection and fitting of a univariate autoregressive time series model.
BAY_SEA	Model allows for a decomposition of a time series into trend, seasonal, and an error component.
GARCH	Computes estimates of the parameters of a GARCH(p,q) model.
MAX_ARMA	Exact maximum likelihood estimation of the parameters in a univariate ARMA (auto-regressive, moving average) time series model.
L10a2a..Summary statistics	
L10a2a1 Autocovariances and autocorrelations	
ACF	Compute the sample autocorrelation function of a stationary time series.
LOFCF	Perform lack-of-fit test for a univariate time series or transfer function given the appropriate correlation function.
L10a2a2 Partial autocorrelations	
PACF	Compute the sample partial autocorrelation function of a stationary time series.
L10a2c..Autoregressive models	
SPWF	Compute the Wiener forecast operator for a stationary stochastic process.
L10a2d .ARMA and ARIMA models (including Box-Jenkins methods)	
L10a2d2 Parameter estimation	
ARMME	Compute method of moments estimates of the autoregressive parameters of an ARMA model.
MAMME	Compute method of moments estimates of the moving average parameters of an ARMA model.
NSLSE	Compute least squares estimates of parameters for a nonseasonal ARMA model.
NSPE	Compute preliminary estimates of the autoregressive and moving average parameters of an ARMA model.
L10a2d3 Forecasting	
NSBJF	Compute Box-Jenkins forecasts and their associated probability limits for a nonseasonal ARMA model.
L10a2e..State-space analysis (e.g., Kalman filtering)	
KALMN	Perform Kalman filtering and evaluate the likelihood function for the state-space model.
L10a3 ...Frequency domain analysis (<i>search also class J1</i>)	

L10a3a . Spectral Analysis

ARMA_SPEC Computes the rational power spectrum for an ARMA model.

L10a3a2 Periodogram analysis

PFFT Compute the periodogram of a stationary time series using a fast Fourier transform.

L10a3a3 Spectrum estimation using the periodogram

SSWD Estimate the nonnormalized spectral density of a stationary time series using a spectral window given the time series data.

SSWP Estimate the nonnormalized spectral density of a stationary time series using a spectral window given the periodogram.

SWED Estimation of the nonnormalized spectral density of a stationary time series based on specified periodogram weights given the time series data.

SWEP Estimation of the nonnormalized spectral density of a stationary time series based on specified periodogram weights given the periodogram.

L10a3a6 Spectral windows

DIRIC Compute the Dirichlet kernel.

FEJER Compute the Fejér kernel.

L10b Two time series (*search also classes L10c and L10d*)

L10b2 ... Time domain analysis

L10b2a . Summary statistics (e.g., cross-correlations)

CCF Compute the sample cross-correlation function of two stationary time series.

L10b2b . Transfer function models

IRNSE Compute estimates of the impulse response weights and noise series of a univariate transfer function model.

TFPE Compute preliminary estimates of parameters for a univariate transfer function model.

L10b3 ... Frequency domain analysis (*search also class J1*)

L10b3a . Cross-spectral analysis

L10b3a3 Cross-spectrum estimation using the cross-periodogram

CSSWD Estimate the nonnormalized cross-spectral density of two stationary time series using a spectral window given the time series data.

CSSWP Estimate the nonnormalized cross-spectral density of two stationary time series using a spectral window given the spectral densities and cross periodogram.

- CSWED Estimate the nonnormalized cross-spectral density of two stationary time series using a weighted cross periodogram given the time series data.
- CSWEP Estimate the nonnormalized cross-spectral density of two stationary time series using a weighted cross periodogram given the spectral densities and cross periodogram.
- L10cMultivariate time series (*search also classes J1, L3e3 and L10b*)
 - KALMN Perform Kalman filtering and evaluate the likelihood function for the state-space model.
 - OPT_DES Allows for multiple channels for both the controlled and manipulated variables
- L10dTwo multi-channel time series
 - MCCF Compute the multichannel cross-correlation function of two mutually stationary multichannel time series.
 - MLSE Compute least squares estimates of a linear regression model for a multichannel time series with a specified base channel.
 - MWFE Compute least squares estimates of the multichannel Wiener filter coefficients for two mutually stationary multichannel time series.
- L11Correlation analysis (*search also classes L4 and L13c*)
 - BSCAT Compute the biserial correlation coefficient for a dichotomous variable and a classification variable.
 - BSPBS Compute the biserial and point-biserial correlation coefficients for a dichotomous variable and a numerically measurable classification variable.
 - CORVC Compute the variance-covariance or correlation matrix.
 - COVPL Compute a pooled variance-covariance matrix from the observations.
 - CTRHO Estimate the bivariate normal correlation coefficient using a contingency table.
 - KENDP Compute the frequency distribution of the total score in Kendall's rank correlation coefficient.
 - PCORR Compute partial correlations or covariances from the covariance or correlation matrix.
 - RBCOV Compute a robust estimate of a covariance matrix and mean vector.
 - TETCC Categorize bivariate data and compute the tetrachoric correlation coefficient.
- L12Discriminant analysis
 - DMSCR Use Fisher's linear discriminant analysis method to reduce the number of variables.

- DSCRM Perform a linear or a quadratic discriminant function analysis among several known groups.
 - NNBRD Perform a k nearest neighbor discrimination.
- L13 Covariance structures models
- L13a Factor analysis
- FACTR Extract initial factor-loading estimates in factor analysis.
 - FCOEF Compute a matrix of factor score coefficients for input to the following IMSL routine (FSCOR).
 - FDOBL Compute a direct oblimin rotation of a factor-loading matrix.
 - FGCRF Compute direct oblique rotation according to a generalized fourth-degree polynomial criterion.
 - FHARR Compute an oblique rotation of an unrotated factor-loading matrix using the Harris-Kaiser method.
 - FIMAG Compute the image transformation matrix.
 - FOPCS Compute an orthogonal Procrustes rotation of a factor-loading matrix using a target matrix.
 - FPRMX Compute an oblique Promax or Procrustes rotation of a factor-loading matrix using a target matrix, including pivot and power vector options.
 - FRESI Compute commonalities and the standardized factor residual correlation matrix.
 - FROTA Compute an orthogonal rotation of a factor-loading matrix using a generalized orthomax criterion, including quartimax, varimax, and equamax rotations.
 - FRVAR Compute the factor structures and the variance explained by each factor.
 - FSCOR Compute a set of factor scores given the factor score coefficient matrix.
- L13b Principal components analysis
- KPRIN Maximum likelihood or least-squares estimates for principle components from one or more matrices.
 - PRINC Compute principal components from a variance-covariance matrix or a correlation matrix.
- L13c Canonical correlation
- CANCR Perform canonical correlation analysis from a data matrix.
 - CANVC Perform canonical correlation analysis from a variance-covariance matrix or a correlation matrix.
- L14 Cluster analysis
- L14a One-way
- L14a1 ... Unconstrained
- L14a1a .Nested
- L14a1a1 Joining (e.g., single link)

- CLINK Perform a hierarchical cluster analysis given a distance matrix.
- L14a1b .Non-nested (e.g., *K* means)
 KMEAN Perform a *K*-means (centroid) cluster analysis.
- L14cDisplay
 TREEP Print a binary tree.
- L14dService routines (e.g., compute distance matrix)
 CDIST Compute a matrix of dissimilarities (or similarities) between the columns (or rows) of a matrix.
 CNUMB Compute cluster membership for a hierarchical cluster tree.
- L15Life testing, survival analysis
 ACTBL Produce population and cohort life tables.
 HAZEZ Perform nonparametric hazard rate estimation using kernel functions. Easy-to-use version of the previous IMSL subroutine (HAZRD).
 HAZRD Perform nonparametric hazard rate estimation using kernel functions and quasi-likelihoods.
 HAZST Perform hazard rate estimation over a grid of points using a kernel function.
 KAPMR Compute Kaplan-Meier estimates of survival probabilities in stratified samples.
 KTBLE Print Kaplan-Meier estimates of survival probabilities in stratified samples.
 NRCES Compute maximum likelihood estimates of the mean and variance from grouped and/or censored normal data.
 PHGLM Analyze time event data via the proportional hazards model.
 STBLE Estimate survival probabilities and hazard rates for various parametric models.
 SVGLM Analyze censored survival data using a generalized linear model.
 TRNBL Compute Turnbull's generalized Kaplan-Meier estimates of survival probabilities in samples with interval censoring.
- L16Multidimensional scaling
 MSDBL Obtain normalized product-moment (double centered) matrices from dissimilarity matrices.
 MSDST Compute distances in a multidimensional scaling model.
 MSIDV Perform individual-differences multidimensional scaling for metric data using alternating least squares.
 MSINI Compute initial estimates in multidimensional scaling models.
 MSSTN Transform dissimilarity/similarity matrices and replace missing values by estimates to obtain standardized dissimilarity matrices.

MSTRS Compute various stress criteria in multidimensional scaling.

L17..... Statistical data sets

 GDATA Retrieve a commonly analyzed data set.

N..... DATA HANDLING (*search also class L2*)

N1..... Input, output

 PGOPT Set or retrieve page width and length for printing.

 WRIRL Print an integer rectangular matrix with a given format and labels.

 WRIRN Print an integer rectangular matrix with integer row and column labels.

 WROPT Set or retrieve an option for printing a matrix.

 WRRRL Print a real rectangular matrix with a given format and labels.

 WRRRN Print a real rectangular matrix with integer row and column labels.

N3..... Character manipulation

 ACHAR Return a character given its ASCII value.

 CVTSI Convert a character string containing an integer number into the corresponding integer form.

 IACHAR Return the integer ASCII value of a character argument.

 ICASE Return the ASCII value of a character converted to uppercase.

 IICSR Compare two character strings using the ASCII collating sequence without regard to case.

 IINDEX Determine the position in a string at which a given character sequence begins without regard to case.

N5..... Searching

N5a..... Extreme value

 EQTIL Compute empirical quantiles.

 ORDST Determine order statistics.

N5b..... Insertion position

 ISRCH Search a sorted integer vector for a given integer and return its index.

 SRCH Search a sorted vector for a given scalar and return its index.

 SSRCH Search a character vector, sorted in ascending ASCII order, for a given string and return its index.

N5c..... On a key

 IINDEX Determine the position in a string at which a given character sequence begins without regard to case.

- ISRCH Search a sorted integer vector for a given integer and return its index.
- SRCH Search a sorted vector for a given scalar and return its index.
- SSRCH Search a character vector, sorted in ascending ASCII order, for a given string and return its index.

N6.....Sorting

N6a.....Internal

N6a1Passive (i.e., construct pointer array, rank)

N6a1a ...Integer

- SVIGP Sort an integer array by algebraic value and return the permutations.

N6a1b...Real

- RANKS Compute the ranks, normal scores, or exponential scores for a vector of observations.
- SCOLR Sort columns of a real rectangular matrix using keys in rows.
- SROWR Sort rows of a real rectangular matrix using keys in columns.
- SVRGP Sort a real array by algebraic value and return the permutations.

N6a2.....Active

N6a2a ...Integer

- SVIGN Sort an integer array by algebraic value.
- SVIGP Sort an integer array by algebraically increasing value and return the permutation that rearranges the array.

N6a2b...Real

- SCOLR Sort columns of a real rectangular matrix using keys in rows.
- SROWR Sort rows of a real rectangular matrix using keys in columns.
- SVRGN Sort a real array by algebraic value.
- SVRGP Sort a real array by algebraic value and return the permutations.

N8.....Permuting

- MVNAN Move any rows of a matrix with the IMSL missing value code NaN (not a number) in the specified columns to the last rows of the matrix.
- PERMA Permute the rows or columns of a matrix.
- PERMU Rearrange the elements of an array as specified by a permutation.
- RORDM Reorder rows and columns of a symmetric matrix.

Q.....GRAPHICS (*search also classes L3*)

BOXP Print boxplots for one or more samples.
 CDF2P Print a plot of two sample cumulative distribution functions.
 CDFP Print a sample cumulative distribution function (CDF), a theoretical CDF, and confidence band information.
 HHSTP Print a horizontal histogram.
 PLOTP Print a plot of up to ten sets of points.
 PROBP Print a probability plot.
 SCTP Print a scatterplot of several groups of data.
 STMLP Print a stem-and-leaf plot.
 TREEP Print a binary tree.
 VHS2P Print a vertical histogram with every bar subdivided into two parts.
 VHSTP Print a vertical histogram.

R..... SERVICE ROUTINES

IDYWK Compute the day of the week for a given date.
 NDAYS Compute the number of days from January 1, 1900, to the given date.
 NDYIN Give the date corresponding to the number of days since January 1, 1900.
 TDATE Get today's date.
 TIMDY Get time of day.
 VERSL Obtain STAT/LIBRARY-related version, system and license numbers.

R1..... Machine-dependent constants

AMACH Retrieve machine constants.
 IFNAN Check if a floating-point number is NaN (not a number).
 IMACH Retrieve integer machine constants.
 UMACH Set or retrieve input or output device unit numbers.

R3..... Error handling

R3b..... Set unit number for error messages

UMACH Set or retrieve input or output device unit numbers.

R3c..... Other utilities

ERSET Set error handler default print and stop actions.
 IERCD Retrieve the code for an informational error.
 N1RTY Retrieve an error type for the most recently called IMSL routine.

S..... SOFTWARE DEVELOPMENT TOOLS

CPSEC Return CPU time used in seconds.

Appendix B: Alphabetical Summary of Routines

IMSL STAT/LIBRARY

ABALD	434	Analyzes a balanced complete experimental design for a fixed, random, or mixed model.
ABIBD	415	Analyzes a balanced incomplete block design or a balanced lattice design.
ACF	697	Computes the sample autocorrelation function of a stationary time series.
ACHAR	1458	Returns a character given its ASCII value.
ACTBL	1121	Produces population and cohort life tables.
AKS1DF	1257	Evaluates the distribution function of the one-sided Kolmogorov-Smirnov goodness-of-fit D^+ or D^- test statistic based on continuous data for one sample.
AKS2DF	1260	Evaluates the distribution function of the Kolmogorov-Smirnov goodness-of-fit D test statistic based on continuous data for two samples.
ALATN	422	Analyzes a Latin square design.
AMACH	1507	Retrieves machine constants.
AMILLR	1489	Evaluates Mill's ratio (the ratio of the ordinate to the upper tail area of the standardized normal distribution).
ANEST	449	Analyzes a completely nested random model with possibly unequal numbers in the subgroups.
ANORDF	1262	Evaluates the standard normal (Gaussian) distribution function.
ANORIN	1264	Evaluates the inverse of the standard normal (Gaussian) distribution function.
ANWAY	427	Analyzes a balanced n -way classification model with fixed effects.

AONEC	395	Analyzes a one-way classification model with covariates.
AONEW	392	Analyzes a one-way classification model.
ARMA_SPEC	822	Calculates the rational power spectrum for an ARMA model.
ARMME	720	Computes method of moments estimates of the autoregressive parameters of an ARMA model.
ATWOB	409	Analyzes a randomized block design or a two-way balanced design.
AUTO_FPE MUL_AR	801	Automatic selection and fitting of a multivariate autoregressive time series model using Akaike's Multivariate Final Prediction Error (MFPE) criteria.
AUTO_FPE UNI_AR	794	Automatic selection and fitting of a univariate autoregressive time series model using Akaike's Final Prediction Error (FPE) criteria.
AUTO_MUL_AR	797	Automatic selection and fitting of a multivariate autoregressive time series model.
AUTO_UNI_AR	791	Automatic selection and fitting of a multivariate autoregressive time series model.
BAY_SEA	803	Allows for a decomposition of a time series into trend, seasonal, and an error component.
BCTR	689	Performs a forward or an inverse Box-Cox (power) transformation.
BETDF	1265	Evaluates the beta probability distribution function.
BETIN	1268	Evaluates the inverse of the beta distribution function.
BHAKV	619	Performs a Bhapkar V test.
BINDF	1246	Evaluates the binomial distribution function.
BINES	48	Estimates the parameter p of the binomial distribution.
BINPR	1248	Evaluates the binomial probability function.
BNRDF	1270	Evaluates the bivariate normal distribution function.
BOXP	1219	Prints boxplots for one or more samples.
BSCAT	377	Computes the biserial correlation coefficient for a dichotomous variable and a classification variable.
BSPBS	374	Computes the biserial and point-biserial correlation coefficients for a dichotomous variable and a numerically measurable classification variable.

CANCR	957	Performs canonical correlation analysis from a data matrix.
CANVC	971	Performs canonical correlation analysis from a variance-covariance matrix or a correlation matrix.
CCF	706	Computes the sample cross-correlation function of two stationary time series.
CDF2P	1226	Prints a plot of two sample cumulative distribution functions.
CDFP	1223	Prints a sample cumulative distribution function (CDF), a theoretical CDF, and confidence band information.
CDIST	1006	Computes a matrix of dissimilarities (or similarities) between the columns (or rows) of a matrix.
CESTI	170	Constructs an equivalent completely testable multivariate general linear hypothesis $HBU = G$ from a partially testable hypothesis $H_pBU = G_p$.
CHFAC	1480	Computes an upper triangular factorization of a real symmetric nonnegative definite matrix.
CHIDF	1271	Evaluates the chi-squared distribution function.
CHIGF	638	Performs a chi-squared goodness-of-fit test.
CHIIN	1274	Evaluates the inverse of the chi-squared distribution function.
CIDMS	469	Computes a confidence interval on a variance component estimated as proportional to the difference in two mean squares in a balanced complete experimental design.
CLINK	1011	Performs a hierarchical cluster analysis given a distance matrix.
CNCRD	379	Calculates and tests the significance of the Kendall coefficient of concordance.
CNUMB	1015	Computes cluster membership for a hierarchical cluster tree.
CORVC	340	Computes the variance-covariance or correlation matrix.
COVPL	348	Computes a pooled variance-covariance matrix from the observations.
CPFFT	855	Computes the cross periodogram of two stationary time series using a fast Fourier transform.
CPSEC	1465	Returns CPU time used in seconds.
CSNDF	1275	Evaluates the noncentral chi-squared distribution function.

CSNIN	1278	Evaluates the inverse of the noncentral chi-squared function.
CSSWD	862	Estimates the nonnormalized cross-spectral density of two stationary time series using a spectral window given the time series data.
CSSWP	873	Estimates the nonnormalized cross-spectral density of two stationary time series using a spectral window given the spectral densities and cross periodogram.
CSTAT	60	Computes cell frequencies, cell means, and cell sums of squares for multivariate data.
CSWED	880	Estimates the nonnormalized cross-spectral density of two stationary time series using a weighted cross periodogram given the time series data.
CSWEP	889	Estimates the nonnormalized cross-spectral density of two stationary time series using a weighted cross periodogram given the spectral densities and cross periodogram.
CTASC	528	Computes partial association statistics for log-linear models in a multidimensional contingency table.
CTCHI	491	Performs a chi-squared analysis of a two-way contingency table.
CTEPR	504	Computes Fisher's exact test probability and a hybrid approximation to the Fisher exact test probability for a contingency table using the network algorithm.
CTGLM	557	Analyzes categorical data using logistic, Probit, Poisson, and other generalized linear models.
CTLLN	513	Computes model estimates and associated statistics for a hierarchical log-linear model.
CTPAR	522	Computes model estimates and covariances in a fitted log-linear model.
CTPRB	501	Computes exact probabilities in a two-way contingency table.
CTRAN	548	Performs generalized Mantel-Haenszel tests in a stratified contingency table.
CTRHO	367	Estimates the bivariate normal correlation coefficient using a contingency table.
CTRST	458	Computes contrast estimates and sums of squares.
CTSTP	536	Builds hierarchical log-linear models using forward selection, backward selection, or stepwise selection.

CTTWO	480	Performs a chi-squared analysis of a 2 by 2 contingency table.
CTWLS	574	Performs a generalized linear least squares analysis of transformed probabilities in a two-dimensional contingency table.
CVTSI	1464	Converts a character string containing an integer number into the corresponding integer form.
DCUBE	667	Performs a triplets test.
DESKN	1176	Performs nonparametric probability density function estimation by the kernel method.
DESPL	1172	Performs nonparametric probability density function estimation by the penalized likelihood method.
DESPT	1185	Estimates a probability density function at specified points using linear or cubic interpolation.
DIFF	693	Difference a time series.
DIRIC	818	Computes the Dirichlet kernel.
DMACH	1509	See AMACH.
DMSCR	993	Uses Fisher's linear discriminant analysis method to reduce the number of variables.
DNFFT	1180	Computes Gaussian kernel estimates of a univariate density via the fast Fourier transform over a fixed interval.
DSCRM	979	Performs a linear or a quadratic discriminant function analysis among several known groups.
DSQAR	664	Performs a D-square test.
ENOS	1487	Evaluates the expected value of a normal order statistic.
EQTIL	38	Computes empirical quantiles.
ERSET	1501	Sets error handler default print and stop actions.
FACTR	909	Extracts initial factor-loading estimates in factor analysis.
FAURE_FREE	1405	Frees the structure containing information about the Faure sequence.
FAURE_INIT	1405	Computes a shuffled Faure sequence.
FAURE_NEXT	1406	Shuffled Faure sequence initialization.
FCOEF	944	Computes a matrix of factor score coefficients for input to the following IMSL routine (FSCOR).
FDF	1280	Evaluates the F distribution function.

FDOBL	924	Computes a direct oblimin rotation of a factor-loading matrix.
FEJER	820	Computes the Fejér kernel.
FGCRF	936	Computes direct oblique rotation according to a generalized fourth-degree polynomial criterion.
FHARR	932	Computes an oblique rotation of an unrotated factor-loading matrix using the Harris-Kaiser method.
FIMAG	940	Computes the image transformation matrix.
FIN	1282	Evaluates the inverse of the F distribution function.
FOPCS	921	Computes an orthogonal Procrustes rotation of a factor-loading matrix using a target matrix.
FPRMX	927	Computes an oblique Promax or Procrustes rotation of a factor-loading matrix using a target matrix, including pivot and power vector options.
FRDMN	620	Performs Friedman's test for a randomized complete block design.
FREQ	14	Tallies multivariate observations into a multi-way frequency table.
FRESI	952	Computes commonalities and the standardized factor residual correlation matrix.
FROTA	918	Computes an orthogonal rotation of a factor-loading matrix using a generalized orthomax criterion, including quartimax, varimax, and equamax rotations.
FRVAR	942	Computes the factor structures and the variance explained by each factor.
FSCOR	949	Computes a set of factor scores given the factor score coefficient matrix.
GAMDF	1284	Evaluates the gamma distribution function.
GAMIN	1286	Evaluates the inverse of the gamma distribution function.
GARCH	745	Computes estimates of the parameters of a GARCH(p,q) model.
GCDF	1295	Evaluates a general continuous cumulative distribution function given ordinates of the density.
GCIN	1298	Evaluates the inverse of a general continuous cumulative distribution function given ordinates of the density.
GCLAS	222	Gets the unique values of each classification variable.
GCSCP	296	Generates centered variables, squares, and crossproducts.
GDATA	1474	Retrieves a commonly analyzed data set.

GFNIN	1301	Evaluates the inverse of a general continuous cumulative distribution function given in a subprogram.
GIRTS	1477	Solves a triangular (possibly singular) set of linear systems and/or compute a generalized inverse of an upper triangular matrix.
GRGLM	225	Generates regressors for a general linear model.
GRPES	56	Computes basic statistics from grouped data.
GSWEP	247	Performs a generalized sweep of a row of a nonnegative definite matrix.
HAZEZ	1195	Performs nonparametric hazard rate estimation using kernel functions. Easy-to-use version of the previous IMSL subroutine (HAZRD).
HAZRD	1188	Performs nonparametric hazard rate estimation using kernel functions and quasi-likelihoods.
HAZST	1203	Performs hazard rate estimation over a grid of points using a kernel function.
HHSTP	1213	Prints a horizontal histogram.
HYPDF	1250	Evaluates the hypergeometric distribution function.
HYPPR	1252	Evaluates the hypergeometric probability function.
IACHAR	1459	Returns the integer ASCII value of a character argument.
ICASE	1460	Returns the ASCII value of a character converted to uppercase.
IDYWK	1471	Computes the day of the week for a given date.
IERCD	1502	Retrieves the code for an informational error.
IFNAN	1509	Checks if a floating-point number is NaN (not a number).
IICSR	1462	Compares two character strings using the ASCII collating sequence without regard to case.
IIDEX	1463	Determines the position in a string at which a given character sequence begins without regard to case.
IMACH	1506	Retrieves integer machine constants.
INCLD	613	Performs an inclusion test.
IRNSE	758	Computes estimates of the impulse response weights and noise series of a univariate transfer function model.
ISRCH	1454	Searches a sorted integer vector for a given integer and returns its index.
KALMN	780	Performs Kalman filtering and evaluate the likelihood function for the state-space model.

KAPMR	1063	Computes Kaplan-Meier estimates of survival probabilities in stratified samples.
KENDL	383	Computes and tests Kendall's rank correlation coefficient.
KENDP	387	Computes the frequency distribution of the total score in Kendall's rank correlation coefficient.
KMEAN	1019	Performs a <i>K</i> -means (centroid) cluster analysis.
KPRIN	905	Maximum likelihood or least-squares estimates for principle components from one or more matrices.
KRSKL	616	Performs a Kruskal-Wallis test for identical population medians.
KSONE	634	Performs a Kolmogorov-Smirnov one-sample test for continuous distributions.
KSTWO	654	Performs a Kolmogorov-Smirnov two-sample test.
KTBLE	1068	Prints Kaplan-Meier estimates of survival probabilities in stratified samples.
KTRND	627	Performs a <i>k</i> -sample trends test against ordered alternatives.
LETTR	32	Produces a letter value summary.
LILLF	646	Performs Lilliefors test for an exponential or normal distribution.
LOFCF	815	Performs lack-of-fit test for a univariate time series or transfers function given the appropriate correlation function.
MAMME	723	Computes method of moments estimates of the moving average parameters of an ARMA model.
MAX_ARMA	741	Exact maximum likelihood estimation of the parameters in a univariate ARMA (auto-regressive, moving average) time series model.
MCCF	711	Computes the multichannel cross-correlation function of two mutually stationary multichannel time series.
MCHOL	1483	Computes an upper triangular factorization of a real symmetric matrix <i>A</i> plus a diagonal matrix <i>D</i> , where <i>D</i> is determined sequentially during the Cholesky factorization in order to make <i>A</i> + <i>D</i> nonnegative definite.
MEDPL	60	Computes a median polish of a two-way table.
MLSE	768	Computes least squares estimates of a linear regression model for a multichannel time series with a specified base channel.

MSDBL	1154	Obtains normalized product-moment (double centered) matrices from dissimilarity matrices.
MSDST	1146	Computes distances in a multidimensional scaling model.
MSIDV	1131	Performs individual-differences multidimensional scaling for metric data using alternating least squares.
MSINI	1159	Computes initial estimates in multidimensional scaling models.
MSSTN	1150	Transforms dissimilarity/similarity matrices and replace missing values by estimates to obtain standardized dissimilarity matrices.
MSTRS	1166	Computes various stress criteria in multidimensional scaling.
MVIND	954	Computes a test for the independence of k sets of multivariate normal variables.
MVMMT	649	Computes Mardia's multivariate measures of skewness and kurtosis and tests for multivariate normality.
MVNAN	1435	Moves any rows of a matrix with the IMSL missing value code NaN (not a number) in the specified columns to the last rows of the matrix.
MWFE	774	Computes least squares estimates of the multichannel Wiener filter coefficients for two mutually stationary multichannel time series.
N1RTY	1502	Retrieves an error type for the most recently called IMSL routine.
NCTRD	599	Performs the Noether test for cyclical trend.
NDAYS	1468	Computes the number of days from January 1, 1900, to the given date.
NDYIN	1470	Gives the date corresponding to the number of days since January 1, 1900.
NGHBR	1494	Searches a k - d tree for the k nearest neighbors of a key.
NNBRD	997	Performs a k nearest neighbor discrimination.
NRCES	53	Computes maximum likelihood estimates of the mean and variance from grouped and/or censored normal data.
NSBJF	752	Computes Box-Jenkins forecasts and their associated probability limits for a nonseasonal ARMA model.
NLSLE	733	Computes least squares estimates of parameters for a nonseasonal ARMA model.
NSPE	727	Computes preliminary estimates of the autoregressive and moving average parameters of an ARMA model.

NTIES	607	Computes tie statistics for a sample of observations.
OPOLY	292	Generates orthogonal polynomials with respect to x values and specified weights.
OPT_DES	810	Allows for multiple channels for both the controlled and manipulated variables
ORDST	34	Determines order statistics.
OWFRQ	3	Tallies observations into a one-way frequency table.
PACF	702	Computes the sample partial autocorrelation function of a stationary time series.
PAIRS	661	Performs a pairs test.
PCORR	353	Computes partial correlations or covariances from the covariance or correlation matrix.
PERMA	1431	Permutes the rows or columns of a matrix.
PERMU	1429	Rearranges the elements of an array as specified by a permutation.
PFFT	825	Computes the periodogram of a stationary time series using a fast Fourier transform.
PGOPT	1428	Sets or retrieves page width and length for printing.
PHGLM	1078	Analyzes time event data via the proportional hazards model.
PLOTP	1232	Prints a plot of up to ten sets of points.
POIDF	1254	Evaluates the Poisson distribution function.
POIES	50	Estimates the parameter of the Poisson distribution.
POIPR	1255	Evaluates the Poisson probability function.
PRINC	901	Computes principal components from a variance-covariance matrix or a correlation matrix.
PROBP	1229	Prints a probability plot.
PRPFT	509	Performs iterative proportional fitting of a contingency table using a loglinear model.
QTEST	625	Performs a Cochran Q test for related observations.
QUADT	1490	Forms a $k-d$ tree.
RANKS	26	Computes the ranks, normal scores, or exponential scores for a vector of observations.
RBCOV	358	Computes a robust estimate of a covariance matrix and mean vector.
RBEST	231	Selects the best multiple linear regression models.

RCASE	204	Computes case statistics and diagnostics given data points, coefficient estimates $\hat{\beta}$, and the R matrix for a fitted general linear model.
RCASP	285	Computes case statistics for a polynomial regression model given the fit based on orthogonal polynomials.
RCOMP	268	Generates an orthogonal central composite design.
RCOV	113	Fits a multiple linear regression model given the variance-covariance matrix.
RCOVB	164	Computes the estimated variance-covariance matrix of the estimated regression coefficients given the R matrix.
RCURV	255	Fits a polynomial curve using least squares.
RFORP	272	Fits an orthogonal polynomial regression model.
RGIVN	117	Fits a multivariate linear regression model via fast Givens transformations.
RGLM	128	Fits a multivariate general linear model.
RHPSS	176	Computes the matrix of sums of squares and crossproducts for the multivariate general linear hypothesis $HB = G$ given the coefficient estimates \hat{B} and the R matrix.
RHPTE	183	Performs tests for a multivariate general linear hypothesis $HB = G$ given the hypothesis sums of squares and crossproducts matrix S_H and the error sums of squares and crossproducts matrix S_E .
RINCF	99	Performs response control given a fitted simple linear regression model.
RINPF	103	Performs inverse prediction given a fitted simple linear regression model.
RLAV	317	Fits a multiple linear regression model using the least absolute values criterion.
RLEQU	142	Fits a multivariate linear regression model with linear equality restrictions $HB = G$ imposed on the regression parameters given results from IMSL routine RGIVN after $IDO = 1$ and $IDO = 2$ and prior to $IDO = 3$.
RLEQU	131	Fits a multivariate linear regression model with linear equality restrictions $HB = G$ imposed on the regression parameters given results from IMSL routine RGIVN after $IDO = 1$ and $IDO = 2$ and prior to $IDO = 3$.
RLINE	86	Fits a line to a set of data points using least squares.

RLLP	322	Fits a multiple linear regression model using the L_p norm criterion.
RLMV	333	Fits a multiple linear regression model using the minimax criterion.
RLOFE	189	Computes a lack-of-fit test based on exact replicates for a fitted regression model.
RLOFN	195	Computes a lack-of-fit test based on near replicates for a fitted regression model.
RLSE	107	Fits a multiple linear regression model using least squares.
RNARM	1390	Generates a time series from a specified ARMA model.
RNBET	1341	Generates pseudorandom numbers from a beta distribution.
RNBIN	1319	Generates pseudorandom numbers from a binomial distribution.
RNCHI	1343	Generates pseudorandom numbers from a chi-squared distribution.
RNCHY	1344	Generates pseudorandom numbers from a Cauchy distribution.
RNCOR	1370	Generates a pseudorandom orthogonal matrix or a correlation matrix.
RNDAT	1373	Generates pseudorandom numbers from a multivariate distribution determined from a given sample.
RNEXP	1346	Generates pseudorandom numbers from a standard exponential distribution.
RNEXT	1347	Generates pseudorandom numbers from a mixture of two exponential distributions.
RNGAM	1349	Generates pseudorandom numbers from a standard gamma distribution.
RNGCS	1351	Sets up table to generate pseudorandom numbers from a general continuous distribution.
RNGCT	1354	Generates pseudorandom numbers from a general continuous distribution.
RNGDA	1321	Generates pseudorandom numbers from a general discrete distribution using an alias method.
RNGDS	1324	Sets up table to generate pseudorandom numbers from a general discrete distribution.
RNGDT	1328	Generates pseudorandom numbers from a general discrete distribution using a table lookup method.

RNGEF	1312	Retrieves the current value of the array used in the IMSL GFSR random number generator.
RNGEO	1331	Generates pseudorandom numbers from a geometric distribution.
RNGES	1312	Retrieves the current value of the table in the IMSL random number generators that use shuffling.
RNGET	1312	Retrieves the current value of the seed used in the IMSL random number generators.
RNHYP	1333	Generates pseudorandom numbers from a hypergeometric distribution.
RNISD	1315	Determines a seed that yields a stream beginning 100,000 numbers beyond the beginning of the stream yielded by a given seed used in IMSL multiplicative congruential generators (with no shufflings).
RNKSM	609	Performs the Wilcoxon rank sum test.
RNLGR	1335	Generates pseudorandom numbers from a logarithmic distribution.
RNLIN	304	Fits a nonlinear regression model.
RNLNL	1356	Generates pseudorandom numbers from a lognormal distribution.
RNMTN	1377	Generates pseudorandom numbers from a multinomial distribution.
RNMVN	1379	Generates pseudorandom numbers from a multivariate normal distribution.
RNNBN	1336	Generates pseudorandom numbers from a negative binomial distribution.
RNNOA	1358	Generates pseudorandom numbers from a standard normal distribution using an acceptance/rejection method.
RNNOF	1359	Generates a pseudorandom number from a standard normal distribution.
RNNOR	1361	Generates pseudorandom numbers from a standard normal distribution using an inverse CDF method.
RNNOS	1386	Generates pseudorandom order statistics from a standard normal distribution.
RNNPP	1394	Generates pseudorandom numbers from a nonhomogeneous Poisson process.
RNOPG	1311	Retrieves the indicator of the type of uniform random number generator.

RNOPT	1311	Selects the uniform (0, 1) multiplicative congruential pseudorandom number generator.
RNPER	1398	Generates a pseudorandom permutation.
RNPOI	1338	Generates pseudorandom numbers from a Poisson distribution.
RNSEF	1167	Initializes the array used in the IMSL GFSR random number generator.
RNSES	1167	Initializes the table in the IMSL random number generators that use shuffling.
RNSET	1167	Initializes a random seed for use in the IMSL randomnumber generators.
RNSPH	1381	Generates pseudorandom points on a unit circle or κ -dimensional sphere.
RNSRI	1399	Generates a simple pseudorandom sample of indices.
RNSRS	1401	Generates a simple pseudorandom sample from a finite population.
RNSTA	1362	Generates pseudorandom numbers from a stable distribution.
RNSTT	1364	Generates pseudorandom numbers from a Student's t distribution.
RNTAB	1383	Generates a pseudorandom two-way table.
RNTRI	1366	Generates pseudorandom numbers from a triangular distribution on the interval (0,1).
RNUN	1316	Generates pseudorandom numbers from a uniform (0,1) distribution.
RNUND	1340	Generates pseudorandom numbers from a discrete uniform distribution.
RNUNF	1317	Generates a pseudorandom number from a uniform (0, 1) distribution.
RNUNO	1388	Generates pseudorandom order statistics from a uniform (0, 1) distribution.
RNVMS	1367	Generates pseudorandom numbers from a von Mises distribution.
RNWIB	1368	Generates pseudorandom numbers from a Weibull distribution.
RONE	89	Analyzes a simple linear regression model.
RORDM	1433	Reorders rows and columns of a symmetric matrix.

ROREX	472	Reorders the responses from a balanced complete experimental design.
ROTIN	215	Computes diagnostics for detection of outliers and influential data points given residuals and the R matrix for a fitted general linear model.
RPOLY	260	Analyzes a polynomial regression model.
RSTAP	279	Computes summary statistics for a polynomial regression model given the fit based on orthogonal polynomials.
RSTAT	152	Computes statistics related to a regression fit given the coefficient estimates $\hat{\beta}$ and the R matrix.
RSTEP	237	Builds multiple linear regression models using forward selection, backward selection, or stepwise selection.
RSUBM	251	Retrieves a symmetric submatrix from a symmetric matrix.
RUNS	657	Performs a runs up test.
SCIPM	460	Computes simultaneous confidence intervals on all pairwise differences of means.
SCOLR	1444	Sorts columns of a real rectangular matrix using keys in rows.
SCTP	1216	Prints a scatterplot of several groups of data.
SDPLC	602	Performs the Cox and Stuart sign test for trends in dispersion and location.
SIGNT	592	Performs a sign test of the hypothesis that a given value is a specified quantile of a distribution.
SMPPR	1026	Computes statistics for inferences regarding the population proportion and total, given proportion data from a simple random sample.
SMPPS	1029	Computes statistics for inferences regarding the population proportion and total, given proportion data from a stratified random sample.
SMPRR	1032	Computes statistics for inferences regarding the population mean and total using ratio or regression estimation, or inferences regarding the population ratio, given a simple random sample.
SMPRS	1039	Computes statistics for inferences regarding the population mean and total using ratio or regression estimation, given continuous data from a stratified random sample.

SMPSC	1046	Computes statistics for inferences regarding the population mean and total using single-stage cluster sampling with continuous data.
SMPSR	1050	Computes statistics for inferences regarding the population mean and total, given data from a simple random sample.
SMPSS	1054	Computes statistics for inferences regarding the population mean and total, given data from a stratified random sample.
SMPST	1058	Computes statistics for inferences regarding the population mean and total, given continuous data from a two-stage sample with equisized primary units.
SNKMC	467	Performs Student-Newman-Keuls multiple comparison test.
SNRNK	595	Performs a Wilcoxon signed rank test.
SPWF	749	Computes the Wiener forecast operator for a stationary stochastic process.
SPWLK	644	Performs a Shapiro-Wilk W -test for normality.
SRCH	1452	Searches a sorted vector for a given scalar and return its index.
SROWR	1448	Sorts rows of a real rectangular matrix using keys in columns.
SSRCH	1456	Searches a character vector, sorted in ascending ASCII order, for a given string and return its index.
SSWD	831	Estimates the nonnormalized spectral density of a stationary time series using a spectral window given the time series data.
SSWP	840	Estimates the nonnormalized spectral density of a stationary time series using a spectral window given the periodogram.
STBLE	1114	Estimates survival probabilities and hazard rates for various parametric models.
STMLP	1221	Prints a stem-and-leaf plot.
SVGLM	1095	Analyzes censored survival data using a generalized linear model.
SVIGN	1442	Sorts an integer array by algebraic value.
SVIGP	1443	Sorts an integer array by algebraic value and returns the permutations.
SVRGN	1439	Sorts a real array by algebraic value.

SVRGP	1440	Sorts a real array by algebraic value and returns the permutations.
SWED	845	Estimation of the nonnormalized spectral density of a stationary time series based on specified periodogram weights given the time series data.
SWEP	851	Estimation of the nonnormalized spectral density of a stationary time series based on specified periodogram weights given the periodogram.
TCSCP	301	Transforms coefficients from a quadratic regression model generated from squares and crossproducts of centered variables to a model using uncentered variables.
TDATE	1467	Gets today's date.
TDF	1288	Evaluates the Student's t distribution function.
TETCC	370	Categorizes bivariate data and compute the tetrachoric correlation coefficient.
TFPE	762	Computes preliminary estimates of parameters for a univariate transfer function model.
TIMDY	1466	Gets time of day.
TIN	1289	Evaluates the inverse of the Student's t distribution function.
TNDF	1291	Evaluates the noncentral Student's t distribution function.
TNIN	1293	Evaluates the inverse of the noncentral Student's t distribution function.
TREEP	1236	Prints a binary tree.
TRNBL	1073	Computes Turnbull's generalized Kaplan-Meier estimates of survival probabilities in samples with interval censoring.
TWFRQ	7	Tallies observations into a two-way frequency table.
TWOMV	41	Computes statistics for mean and variance inferences using samples from two normal populations.
UMACH	1510	Sets or retrieves input or output device unit numbers.
UVSTA	18	Computes basic univariate statistics.
VERSL	1472	Obtains STAT/LIBRARY-related version, system and license numbers.
VHS2P	1210	Prints a vertical histogram with every bar subdivided into two parts.
VHSTP	1208	Prints a vertical histogram.

WRIRL	1418	Prints an integer rectangular matrix with a given format and labels.
WRIRN	1416	Prints an integer rectangular matrix with integer row and column labels.
WROPT	1421	Sets or retrieves an option for printing a matrix.
WRRRL	1412	Prints a real rectangular matrix with a given format and labels.
WRRRN	1410	Prints a real rectangular matrix with integer row and column labels.

Appendix C: References

Abraham and Ledolter

Abraham, Bovas, and Johannes Ledolter (1983), *Statistical Methods for Forecasting*, John Wiley & Sons, New York.

Abramowitz and Stegun

Abramowitz, Milton, and Irene A. Stegun (editors) (1964), *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, National Bureau of Standards, Washington.

Afifi and Azen

Afifi, A.A. and S.P. Azen (1979), *Statistical Analysis : A Computer Oriented Approach*, second edition, Academic Press, New York.

Agresti, Wackerly, and Boyette

Agresti, Alan, Dennis Wackerly, and James M. Boyette (1979), Exact conditional tests for cross-classifications: Approximation of attained significance levels, *Psychometrika*, **44**, 75-83.

Ahrens and Dieter

Ahrens, J.H., and U. Dieter (1974), Computer methods for sampling from gamma, beta, Poisson, and binomial distributions, *Computing*, **12**, 223–246.

Ahrens, J.H., and U. Dieter (1985), Sequential random sampling, *ACM Transactions on Mathematical Software*, **11**, 157–169.

Aird and Howell

Aird, Thomas J., and Byron W. Howell (1991), IMSL Technical Report 9103, IMSL, Houston.

Akaike

Akaike, H. (1980) Seasonal Adjustment by Bayesian Modeling. *Journal of Time Series Analysis*, Vol 1, 1-13.

Akaike, H., et al

Akaike, H., Kitagawa, G., Arahata, E., and Tada, F. (1979) Computer Science Monographs, No. 13, Institute of Statistical Mathematics, Tokyo.

Akaike and Nakagawa

Akaike, H. and Nakagawa, T. (1972) *Statistical Analysis and Control of Dynamic Systems*, KTK Scientific Publishers, Tokyo.

Akima

Akima, Hirosha (1970), A new method of interpolation and smooth curve fitting based on local procedures, *Journal of the ACM*, **17**, 589–602.

Anderberg

Anderberg, Michael R. (1973), *Cluster Analysis for Applications*, Academic Press, New York.

Anderson

Anderson, T.W. (1971), *The Statistical Analysis of Time Series*, John Wiley & Sons, New York.

Anderson and Bancroft

Anderson, R.L., and T.A. Bancroft (1952), *Statistical Theory in Research*, McGraw-Hill Book Company, New York.

Anderson and Rubin

Anderson, T., and H. Rubin (1956), Statistical inference in factor analysis, *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, Volume 5, University of California Press, Berkeley, 111–150.

Atkinson

Atkinson, A.C. (1973), Testing transformations to normality, *Journal of the Royal Statistical Society, Series B: Methodological*, **35**, 473–479.

Atkinson, A.C. (1979), A family of switching algorithms for the computer generation of beta random variates, *Biometrika*, **66**, 141–145.

Atkinson, A.C. (1985), *Plots, Transformations, and Regression*, Clarendon Press, Oxford.

Atkinson, A.C. (1986), Diagnostic tests for transformations, *Technometrics*, **28**, 29–37.

Baker, Clarke, and Lane

Baker, R.J., M.R.B. Clarke, and P.W. Lane (1985). Zero entries in contingency tables, *Computational Statistics and Data Analysis*, **3**, 33-45.

Bartlett

Bartlett, M.S. (1935), Contingency table interactions, *Journal of the Royal Statistics Society Supplement*, **2**, 248–252.

Bartlett, M. (1937), The statistical conception of mental factors, *British Journal of Psychology*, **28**, 97–104.

Bartlett, M.S. (1946), On the theoretical specification and sampling properties of autocorrelated time series, *Supplement to the Journal of the Royal Statistical Society*, **8**, 27–41.

Bartlett, M.S. (1978), *Stochastic Processes*, 3rd. ed., Cambridge University Press, Cambridge.

Barrodale and Roberts

Barrodale, I., and F.D.K. Roberts (1973), An improved algorithm for discrete L_1 approximation, *SIAM Journal on Numerical Analysis*, **10**, 839–848.

Barrodale, I., and C. Phillips (1975), Algorithm 495. Solution of an overdetermined system of linear equations in the Chebyshev norm, *ACM Transactions on Mathematical Software*, **1**, 264–270.

Barrodale, I., and F.D.K. Roberts (1974), Solution of an overdetermined system of equations in the l_1 norm, *Communications of the ACM*, **17**, 319–320.

Barlow et al.

Barlow, R.E., D.J. Bartholomew, J.M. Bremner, and H.D. Brunk (1972), *Statistical Inference under Order Restrictions*, John Wiley & Sons, London.

Bendel and Mickey

Bendel, Robert B., and M. Ray Mickey (1978), Population correlation matrices for sampling experiments, *Communications in Statistics*, **B7**, 163–182.

Berk

Berk, Kenneth. N. (1976), Tolerance and condition in regression computations, *Proceedings of the Ninth Interface Symposium on Computer Science and Statistics*, Prindle, Weber and Schmidt, Boston, 202–203.

Best and Fisher

Best, D.J., and N.I. Fisher (1979), Efficient simulation of the von Mises distribution, *Applied Statistics*, **28**, 152–157.

Bhapkar

Bhapkar, V.P. (1961), A nonparametric test for the problem of several samples, *Annals of Mathematical Statistics*, **32**, 1108–1117.

Bishop, Feinberg, and Holland

Bishop, Yvonne M. M., Stephen E. Feinberg, and Paul W. Holland (1975), *Discrete Multivariate Analysis*, The MIT Press, Cambridge, Mass.

Bjorck and Golub

Bjorck, Ake, and Gene H. Golub (1973), Numerical Methods for Computing Angles Between Subspaces, *Mathematics of Computation*, **27**, 579–594.

Blackman and Tukey

Blackman, R.B., and J. W. Tukey (1958), *The Measurement of Power Spectra from the Point of View of Communications Engineering*, Dover Publications, New York.

Blom

Blom, Gunnar (1958), *Statistical Estimates and Transformed Beta-Variables*, John Wiley & Sons, New York.

Boisvert, Howe, and Kahaner

Boisvert, Ronald F., Sally E. Howe, and David K. Kahaner (1985), GAMS: A framework for the management of scientific software, *ACM Transactions on Mathematical Software*, **11**, 313-355.

Boisvert, Howe, Kahaner, and Springmann

Boisvert, Ronald F., Sally E. Howe, David K. Kahaner, and Jeanne L. Springmann (1990), *Guide to Available Mathematical Software*, NISTIR 90-4237, National Institute of Standards and Technology, Gaithersburg, Maryland.

Bosten and Battiste

Bosten, Nancy E., and E.L. Battiste (1974), Incomplete beta ratio, *Communications of the ACM* **17**, 156–157.

Box and Cox

Box, G.E.P., and D.R. Cox (1964), An analysis of transformations, *Journal of the Royal Statistical Society, Series B: Methodological*, **26**, 211–243.

Box and Jenkins

Box, George E.P., and Gwilym M. Jenkins (1976), *Time Series Analysis: Forecasting and Control*, rev. ed., Holden-Day, Oakland, Calif.

Box and Pierce

Box, G.E.P., and David A. Pierce (1970), Distribution of residual autocorrelations in autoregressive-integrated moving average time series models, *Journal of the American Statistical Association*, **65**, 1509–1526.

Box and Tidwell

Box, G.E.P., and P.W. Tidwell (1962), Transformation of the independent variables, *Technometrics*, **4**, 531–550.

Boyette

Boyette, James M. (1979), Random RC tables with given row and column totals, *Applied Statistics*, **28**, 329–332.

Bradley

Bradley, J.V. (1968), *Distribution-Free Statistical Tests*, Prentice-Hall, New Jersey.

Bradley, J.V. (1968), *Distribution-Free Statistical Inference*, Prentice-Hall, New Jersey.

Breslow

Breslow, N.E. (1974), Covariance analysis of censored survival data, *Biometrics*, **30**, 89–99.

Brillinger

Brillinger, David R. (1981), *Time Series: Data Analysis and Theory*, expanded ed., Holden-Day, San Francisco.

Bross

Bross, I. (1950), Fiducial intervals for variance components, *Biometrics*, **6**, 136–144.

Brown

Brown, Morton B. (1983), BMDP4F, two-way and multiway frequency tables measures of association and the log-linear model (complete and incomplete tables), in *BMDP Statistical Software, 1983 Printing with Additions*, (edited by W. J. Dixon), University of California Press, Berkeley.

Brown and Benedetti

Brown, Morton B., and Jacqueline K. Benedetti (1977), Sampling behavior and tests for correlation in two-way contingency tables, *Journal of the American Statistical Association*, **42**, 309–315.

Brown and Fuchs

Brown, Morton B., and C. Fuchs (1983), On maximum likelihood estimation in sparse contingency tables, *Computational Statistics and Data Analysis*, **1**, 3–15.

Bryson and Johnson

Bryson, Maurice C. and Mark E. Johnson (1981), The incidence of monotone likelihood in the Cox model, *Technometrics*, **23**, 381–384.

Cantor

Cantor, Alan B. (1979), A computer algorithm for testing significance in M K contingency tables, *Proceedings of the Statistical Computing Section, American Statistical Association*, Washington, D.C., 220–221.

Carroll and Chang

Carroll, J.D., and J.J. Chang (1970), Analysis of individual differences in multidimensional scaling via an n -way generalization of “Eckart-Young” decomposition, *Psychometrika*, **35**, 283–319.

Chambers et al.

Chambers, J.M., C.L. Mallows, and B.W. Stuck (1976), A method for simulating stable random variates, *Journal of the American Statistical Association*, **71**, 340–344.

Chambers, John M., William S. Cleveland, Beat Kleiner, and Paul A. Tukey (1983), *Graphical Methods for Data Analysis*, Wadsworth, Belmont, Calif.

Chatfield

Chatfield, C. (1980), *The Analysis of Time Series: An Introduction*, 2d ed., Chapman and Hall, London.

Chiang

Chiang, Chin Long (1968), *Introduction to Stochastic Processes in Statistics*, John Wiley & Sons, New York.

Chiang, Chin Long (1972), On constructing current life tables, *Journal of the American Statistical Association*, **67**, 538–541.

Cheng

Cheng, R.C.H. (1978), Generating beta variates with nonintegral shape parameters, *Communications of the ACM*, **21**, 317–322.

Christensen

Christensen, Ronald (1989), Lack-of-fit tests based on near or exact replicates, *Annals of Statistics*, **17**, 673–683.

Clarke

Clarke, M.R.B. (1982), The Gauss-Jordan sweep operator with detection of collinearity, *Applied Statistics*, **31**, 166–168.

Clarkson

Clarkson, Douglas B. (1988a), Remark on Algorithm AS 211: The F-G diagonalization algorithm, *Applied Statistics*, **38**, 147–151.

Clarkson, Douglas B. (1988b), A least-squares version of AS 211: The F-G diagonalization algorithm, *Applied Statistics*, **38**, 317–321.

Clarkson and Fan

Clarkson, Douglas B. and Yuan-An Fan (1989), *Some improvements to the network algorithm for exact probabilities in contingency tables*, IMSL Technical Report 8903, IMSL, Houston.

Clarkson and Gentle

Clarkson, Douglas B. and James E. Gentle (1986), Methods for multidimensional scaling, in *Computer Science and Statistics, Proceedings of the Seventeenth Symposium on the Interface*, (D.M. Allen, editor), North-Holland, Amsterdam, 185–192.

Clarkson and Jennrich

Clarkson, Douglas B. and Robert I. Jennrich (1988), Computing extended maximum likelihood estimates for linear parameter models, *IMSL Technical Report 8804*, IMSL, Houston.

Clarkson, Douglas B. and Robert I. Jennrich (1988), Quartic rotation criteria and algorithms, *Psychometrika*, **53**, 251–259.

Clarkson, Douglas B. and Robert I. Jennrich (1991), Computing extended maximum likelihood estimates for linear parameter models, submitted to *Journal of the Royal Statistical Society, Series B*, **53**, 417-426.

Cochran

Cochran, William G. (1977), *Sampling Techniques*, 3rd ed., John Wiley & Sons, New York.

Conover

Conover, W.J. (1980), *Practical Nonparametric Statistics*, 2d ed., John Wiley & Sons, New York.

Conover and Iman

Conover, W.J., and Ronald L. Iman (1983), *Introduction to Modern Business Statistics*, John Wiley & Sons, New York.

Cook and Weisberg

Cook, R. Dennis, and Sanford Weisberg (1982), *Residuals and Influence in Regression*, Chapman and Hall, New York.

Cooper

Cooper, B.E. (1968), Algorithm AS4, An auxiliary function for distribution integrals, (*Applied Statistics*), **17**, 190–192.

Coveyou and MacPherson

Coveyou, R.R., and R.D. MacPherson (1967), Fourier analysis of uniform random number generators, *Journal of the ACM*, **14**, 100–119.

Cox

Cox, David R. (1970), *The Analysis of Binary Data*, Methuen, London.

Cox, D.R. (1972), Regression models and life tables (with discussion), *Journal of the Royal Statistical Society, Series B, Methodology*, **34**, 187–220.

Cox and Lewis

Cox, D.R., and P.A.W. Lewis (1966), *The Statistical Analysis of Series of Events*, Methuen, London.

Cox and Oakes

Cox, D.R., and D. Oakes (1984), *Analysis of Survival Data*, Chapman and Hall, London.

Cox and Stuart

Cox, D.R., and A. Stuart (1955), Some quick sign tests for trend in location and dispersion, *Biometrika*, **42**, 80–95.

Craddock

Craddock, J.M. (1969), *Statistics in the Computer Age*, American Elsevier, New York.

Crawford and Ferguson

Crawford, C.B. and G.A. Ferguson (1970), A general rotation criteria and its use in orthogonal rotation, *Psychometrika*, **35**, 321–332.

D'Agostino and Stevens

D'Agostino, Ralph B. and Michael A. Stevens (1986) *Goodness-of-Fit Techniques*, Marcel Dekker, New York.

Dahlquist and Bjorck

Dahlquist, Germund, and Ake Bjorck (1974), *Numerical Methods*, translated by Ned Anderson, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Dallal and Wilkinson

Dallal, Gerald E. and Leland Wilkinson (1986), An analytic approximation to the distribution of Lilliefors's test statistic for normality, *The American Statistician*, **40**, 294–296.

Davison

Davison, Mark L. (1983), *Multidimensional Scaling*, John Wiley & Sons, New York.

De Leeuw and Pruzansky

De Leeuw, Jan and Sandra Pruzansky (1978), A new computational method to fit the weighted Euclidean distance model, *Psychometrika*, **43**, 479–490.

Deming and Stephan

Deming, W.E., and F.F. Stephan (1940), On the least-squares adjustments of a sampled frequency table when the expected marginal totals are known, *Annals of Mathematical Statistics*, **11**, 427–444.

Demspter, Nan, and Rubin

Demspter, Arthur P., Nan, Laird, and Donald B. Rubin (1977), Maximum likelihood from incomplete data via the EM algorithm (with discussion), *Journal of the Royal Statistical Society, Serie B*, **39**, 1–38.

Dennis and Schnabel

Dennis, John E., Jr., and Robert B. Schnabel (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, New Jersey.

Dewey

Dewey, Michael E. (1984), A remark on Algorithm AS 169: An improved algorithm for scatter plots, *Applied Statistics*, **33**, 370–372.

Draper and Smith

Draper, N.R., and H. Smith (1981), *Applied Regression Analysis*, 2d ed., John Wiley & Sons, New York.

Durbin

Durbin, J. (1960), The fitting of time series models, *Revue Institute Internationale de Statistics*, **28**, 233–243.

Efroymson

Efroymson, M.A. (1960), Multiple regression analysis, in *Mathematical Methods for Digital Computers*, Volume 1, (edited by A. Ralston and H. Wilf), John Wiley & Sons, New York, 191–203.

Eklblom

Eklblom, Hakan (1973), Calculation of linear best L_p -approximations, *BIT*, **13**, 292–300.

Eklblom, Hakan (1987), The L_1 -estimate as limiting case of an L_p or Huber-estimate, in *Statistical Data Analysis Based on the L_1 -Norm and Related Methods* (edited by Yadolah Dodge), North-Holland, Amsterdam, 109–116.

Elandt-Johnson and Johnson

Elandt-Johnson, Regina C., and Norman L. Johnson (1980), *Survival Models and Data Analysis*, John Wiley & Sons, New York, 172—173.

Emerson and Hoaglin

Emerson, John D., and David C. Hoaglin (1983), Analysis of two-way tables by medians, in *Understanding Robust and Exploratory Data Analysis* (edited by David C. Hoaglin, Frederick Mosteller, and John W. Tukey), John Wiley & Sons, New York, 166–210.

Emmett

Emmett W.G. (1949), Factor analysis by Lawley's method of maximum likelihood, *British Journal of Psychology*, **2**, 90-97.

Fisher

Fisher, R.A. (1936), The use of multiple measurements in taxonomic problems, *The Annals of Eugenics*, **7**, 179–188.

Fishman

Fishman, George S. (1978), *Principles of Discrete Event Simulation*, John Wiley & Sons, New York.

Fishman et al.

Fishman, George F., and Louis R. Moore, III (1982), A statistical evaluation of multiplicative random number generators with modulus 2311, *Journal of the American Statistical Association*, **77**, 129–136.

Fishman, George F., and Louis R. Moore, III (1986), An exhaustive analysis of multiplicative congruential random number generators with modulus $2^{31} - 1$, *SIAM Journal on Scientific and Statistical Computing*, **7**, 24–45.

Flury

Flury, Bernard H. (1984), Common principal components in k groups, *Journal of the American Statistical Association*, **79**, 892–898.

Flury, Bernard H. (1988), *Common Principal Components & Related Multivariate Models*, John Wiley & Sons, New York.

Flury and Constantine

Flury, Bernard H. and Gregory Constantine (1985), The F-G diagonalization algorithm, *Applied Statistics*, **35**, 177–183.

Flury and Gautschi

Flury, Bernard H. and Walter Gautschi (1986), An algorithm for simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly diagonal form, *SIAM Journal of Scientific and Statistical Computing*, **7**, 169–185.

Forsythe

Forsythe, G.E. (1957), Generation and use of orthogonal polynomials for fitting data with a digital computer, *SIAM Journal on Applied Mathematics*, **5**, 74–88.

Forthofer and Koch

Forthofer, Ronald N., and Gary G. Koch (1973), An analysis of compounded functions of categorical data, *Biometrics*, **29**, 143–157.

Fox, Hall, and Schryer

Fox, P.A., A.D. Hall, and N.L. Schryer (1978), The PORT mathematical subroutine library, *ACM Transactions on Mathematical Software*, **4**, 104–126.

Frane

Frane, James W. (1977), A note on checking tolerance in matrix inversion and regression, *Technometrics*, **19**, 513–514.

Freeman and Halton

Freeman, G.H., and J.H. Halton (1951), Note on the exact treatment of contingency, goodness of fit, and other problems of significance, *Biometrika*, **38**, 141–149.

Friedman, Bentley, and Finkel

Friedman, Jerome H., Jon Louis Bentley, and Raphael Ari Finkel (1977), An algorithm for finding best matches in logarithmic expected time, *ACM Transactions on Mathematical Software*, **3**, 209–226.

Fuller

Fuller, Wayne A. (1976), *Introduction to Statistical Time Series*, John Wiley & Sons, New York.

Furnival and Wilson

Furnival, G.M., and R.W. Wilson, Jr. (1974), *Regressions by leaps and bounds*, *Technometrics*, **16**, 499–511.

Fushimi

Fushimi, Masanori (1990), Random number generation with the recursion $X_t = X_{t-3p} \oplus X_{t-3q}$, *Journal of Computational and Applied Mathematics*, **31**, 105–118.

Gentle

Gentle, James E. (1981), Portability considerations for random number generators, in *Computer Science and Statistics: The Interface*, (edited by William F. Eddy), SpringerVerlag, New York, 158–161.

Gentle, James E. (1990), Computer implementation of random number generators, *Journal of Computational and Applied Mathematics*, **31**, 119–125.

Gentleman

Gentleman, W. Morven (1974), Basic procedures for large, sparse or weighted linear least squares problems, *Applied Statistics*, **23**, 448–454.

Gibbons

Gibbons, J.D. (1971), *Nonparametric Statistical Inference*, McGraw-Hill, New York.

Girshick

Girshick, M.A. (1939), On the sampling theory of roots of determinantal equations, *Annals of Mathematical Statistics*, **10**, 203–224.

Golub

Golub, Gene H. (1969), Matrix computations and statistical calculations, in *Statistical Computation*, (edited by Roy C. Milton and John A. Nelder), Academic Press, New York. 365–398.

Golub and Van Loan

Golub, Gene H. and Charles F. Van Loan (1983), *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Maryland.

Gonin and Money

Gonin, Rene, and Arthur H. Money (1989), *Nonlinear L_p -Norm Estimation*, Marcel Dekker, New York.

Goodnight

Goodnight, James H. (1979), A tutorial on the SWEEP operator, *The American Statistician*, **33**, 149–158.

Granger and Newbold

Granger, C.W.J., and Paul Newbold (1977), *Forecasting Economic Time Series*, Academic Press, Orlando, Florida.

Graybill

Graybill, Franklin A. (1976), *Theory and Application of the Linear Model*, Duxbury Press, North Scituate, Mass.

Griffin and Redish

Griffin, R., and K.A. Redish (1970), Remark on Algorithm 347: An efficient algorithm for sorting with minimal storage, *Communications of the ACM*, **13**, 54.

Grizzle, Starmer, and Koch

Grizzle, J.E., C.F. Starmer, and G.G. Koch, (1969), Analysis of categorical data by linear models, *Biometrics*, **28**, 489-504.

Gross and Clark

Gross, Alan J., and Virginia A. Clark (1975), *Survival Distributions: Reliability Applications in the Biomedical Sciences*, John Wiley & Sons, New York.

Gruenberger and Mark

Gruenberger, F., and A.M. Mark (1951), The d^2 test of random digits, *Mathematical Tables and Other Aids in Computation*, **5**, 109–110.

Guerra et al.

Guerra, Victor O., Richard A. Tapia, and James R. Thompson (1976), A random number generator for continuous random variables based on an interpolation procedure of Akima, in *Proceedings of the Ninth Interface Symposium on Computer Science and Statistics*, (edited by David C. Hoaglin and Roy E. Welsch), Prindle, Weber & Schmidt, Boston, 228–230.

Haberman

Haberman, S.J. (1972), Log-linear fit for contingency tables, *Applied Statistics*, **21**, 218–225.

Haldane

Haldane, J.B.S. (1939), The mean and variance of χ^2 when used as a test of homogeneity, when expectations are small, *Biometrika*, **31**, 346.

Hancock

Hancock, T.W. (1975), Remark on Algorithm 434: Exact probabilities for $R \times C$ contingency tables, *Communications of the ACM*, **18**, 117–119.

Hand

Hand, D.J. (1981), *Discrimination and Classification*, John Wiley & Sons, New York.

Harman

Harman, Harry H. (1976), *Modern Factor Analysis*, 3rd. ed. revised, University of Chicago Press, Chicago.

Harris and Kaiser

Harris, C., and H. Kaiser (1964), Oblique factor analysis solutions by orthogonal transformations, *Psychometrika*, **29**, 347–362.

Hart, et al.

Hart, John F., E.W. Cheney, Charles L. Lawson, Hans J. Maehly, Charles K. Mesztenyi, John R. Rice, Henry G. Thacher, Jr., and Christoph Witzgall (1968), *Computer Approximations*, John Wiley & Sons, New York.

Hartigan

Hartigan, John A. (1975), *Clustering Algorithms*, John Wiley & Sons, New York.

Hartigan and Wong

Hartigan, J.A., and M.A. Wong (1979), Algorithm AS 136: A *K*-means clustering algorithm, *Applied Statistics*, **28**, 100–108.

Harvey

Harvey, A.C. (1981a), *The Econometric Analysis of Time Series*, Philip Allen Publishers, Deddington, England.

Harvey, A.C. (1981b), *Time Series Models*, John Wiley & Sons, New York.

Hayter

Hayter, Anthony J. (1984), A proof of the conjecture that the Tukey-Kramer multiple comparisons procedure is conservative, *Annals of Statistics*, **12**, 61–75.

Heiberger

Heiberger, Richard M. (1978), Generation of random orthogonal matrices, *Applied Statistics*, **27**, 199–206.

Hemmerle

Hemmerle, William J. (1967), *Statistical Computations on a Digital Computer*, Blaisdell Publishing Company, Waltham, Mass.

Hendrickson and White

Hendrickson, A., and P. White (1964), PROMAX: A quick method for rotation to oblique simple structure, *British Journal of Statistical Psychology*, **17**, 65–70.

Herraman

Herraman, C. (1968), Sums of squares and products matrix, *Applied Statistics*, **17**, 289–292.

Hill

Hill, G.W. (1970), Student's t -distribution, *Communications of the ACM*, **13**, 617–620.

Hinkley

Hinkley, David (1977), On quick choice of power transformation, *Applied Statistics*, **26**, 67–69.

Hoaglin

Hoaglin, David C. (1983), Letter values: A set of selected order statistics, in *Understanding Robust and Exploratory Data Analysis* (edited by David C. Hoaglin, Frederick Mosteller, and John W. Tukey), John Wiley & Sons, New York, 33–57.

Hoaglin et al.

Hoaglin, David C., Frederick Mosteller, and John W. Tukey (editors) (1983), *Understanding Robust and Exploratory Data Analysis*, John Wiley & Sons, New York.

Hoaglin and Welsh

Hoaglin, D.C., and R. Welsh (1978), The hat matrix in regression and ANOVA, *American Statistician*, **32**, 17–22.

Hocking

Hocking, R.R. (1972), Criteria for selection of a subset regression: Which one should be used?, *Technometrics*, **14**, 967–970.

Hocking, R.R. (1973), A discussion of the two-way mixed model, *The American Statistician*, **27**, 148–152.

Hocking, R.R. (1985), *The Analysis of Linear Models*, Brooks/Cole Publishing Company, Monterey, California.

Huber

Huber, Peter J. (1977), Robust covariances, in *Statistical Decision Theory and Related Topics*, S.S. Gupta and D.S. Moore (editors), Academic Press, New York.

Huber, Peter J. (1981), *Robust Statistics*, John Wiley & Sons, New York.

Hughes and Saw

Hughes, David T., and John G. Saw (1972), Approximating the percentage points of Hotelling's generalized T_0^2 statistic, *Biometrika*, **59**, 224–226.

Hurley and Cattell

Hurley, J., and R. Cattell (1962), The Procrustes program: Producing direct rotation to test a hypothesized factor structure, *Behavioral Science*, **7**, 258–262.

IEEE

ANSI/IEEE Std 754-1985 (1985), *IEEE Standard for Binary Floating-Point Arithmetic*, The IEEE, Inc., New York.

Iman and Davenport

Iman, R.L., and J.M. Davenport (1980), Approximations of the critical region of the Friedman statistic, *Communications in Statistics*, **A9(6)**, 571–595.

Isogai

Isogai, Takafumi (1983), On measures of multivariate skewness and kurtosis, *Mathematica Japonica*, **28**, 251–261.

Jenkins and Watts

Jenkins, Gwilym M., and Donald G. Watts (1968), *Spectral Analysis and Its Applications*, Holden-Day, Oakland, Calif.

Jennrich

Jennrich, Robert I. (1973), Standard errors for obliquely rotated factor loadings, *Psychometrika*, **38**, 593–604.

Jennrich and Robinson

Jennrich, R.I., and S.M. Robinson (1969), A Newton-Raphson algorithm for maximum likelihood factor analysis, *Psychometrika*, **34**, 111–123.

Jennrich and Sampson

Jennrich, R.I. and P.F. Sampson (1966), Rotation for simple loadings, *Psychometrika*, **31**, 313–323.

John (1971)

John, Peter W.M. (1971), *Statistical Design and Analysis of Experiments*, Macmillan Company, New York.

Johnk

Johnk, M.D. (1964), Erzeugung von Beta- und Gamma-verteiltern Zufallszahlen, *Metrika*, **8**, 5–15.

Johnson and Kotz

Johnson, Norman L., and Samuel Kotz (1969), *Discrete Distributions*, Houghton Mifflin Company, Boston.

Johnson, Norman L., and Samuel Kotz (1970a), *Continuous Univariate Distributions-1*, John Wiley & Sons, New York.

Johnson, Norman L., and Samuel Kotz (1970b), *Continuous Univariate Distributions-2*, John Wiley & Sons, New York.

Johnson and Welch

Johnson, D.G., and W.J. Welch (1980), The generation of pseudo-random correlation matrices, *Journal of Statistical Computation and Simulation*, **11**, 55–69.

Jonckheere

Jonckheere, A.R. (1954), A distribution-free k -sample test against ordered alternatives, *Biometrika*, **41**, 133–143.

Joreskog

Joreskog, K.G. (1977), Factor analysis by least squares and maximum-likelihood methods, in *Statistical Methods for Digital Computers*, (edited by Kurt Enslein, Anthony Ralston, and Herbert S. Wilf), John Wiley & Sons, New York, 125–153.

Kachitvichyanukul

Kachitvichyanukul, Voratas (1982), *Computer generation of Poisson, binomial, and hypergeometric random variates*, Ph.D. dissertation, Purdue University, West Lafayette, Indiana.

Kaiser

Kaiser, H.F. (1958), The varimax criterion for analytic rotation in factor analysis, *Psychometrika*, **23**, 187–200.

Kaiser, H.F. (1963), Image analysis, in *Problems in Measuring Change*, (edited by C. Harris), University of Wisconsin Press, Madison, Wisconsin.

Kaiser and Caffrey

Kaiser, H.F., and J. Caffrey (1965), Alpha factor analysis, *Psychometrika*, **30**, 1–14.

Kalbfleisch and Prentice

Kalbfleisch, John D., and Ross L. Prentice (1980), *The Statistical Analysis of Failure Time Data*, John Wiley & Sons, New York.

Kalman

Kalman, R. E. (1960), A new approach to linear filtering and prediction problems, *Journal of Basic Engineering*, **82**, 35–45.

Kelly

Kelly, L.G. (1967), *Handbook of Numerical Methods and Applications*, Addison-Wesley, Reading, Mass.

Kemp

Kemp, A.W., (1981), Efficient generation of logarithmically distributed pseudo-random variables, *Applied Statistics*, **30**, 249–253.

Kempthorne

Kempthorne, Oscar (1975), *The Design and Analysis of Experiments*, Robert E. Krieger Publishing Company, Huntington, New York.

Kendall

Kendall, Maurice G. (1962), *Rank Correlation Methods*, Charles Griffin & Company, 94–100.

Kendall, Stuart, and Ord

Kendall, Maurice G., Alan Stuart, and J. Keith Ord (1983), *The Advanced Theory of Statistics*, Volume 3: *Design and Analysis, and Time Series*, 4th ed., Oxford University Press, New York.

Kendall, Maurice G., Alan Stuart, and J. Keith Ord (1987), *The Advanced Theory of Statistics*, Volume 1: *Distribution Theory*, 5th ed., Oxford University Press, New York.

Kendall and Stuart

Kendall, Maurice G., and Alan Stuart (1979), *The Advanced Theory of Statistics*, Volume 2: *Inference and Relationship*, 4th ed., Oxford University Press, New York.

Kennedy and Gentle

Kennedy, William J., and James E. Gentle (1980), *Statistical Computing*, Marcel Dekker, New York.

Kim and Jennrich

Kim, P.J., and R.I. Jennrich (1973), Tables of the exact sampling distribution of the two sample Kolmogorov-Smirnov criterion D_{mn} ($m < n$), in *Selected Tables in Mathematical Statistics*,

Volume 1, (edited by H. L. Harter and D.B. Owen), American Mathematical Society, Providence, Rhode Island.

Kinderman

Kinderman, A.J., and J.G. Ramage (1976), Computer generation of normal random variables, *Journal of the American Statistical Association*, **71**, 893–896.

Kinderman, A.J., J.F. Monahan, and J.G. Ramage (1977), Computer methods for sampling from Student's *t* distribution, *Mathematics of Computation* **31**, 1009–1018.

Kinnucan and Kuki

Kinnucan, P., and H. Kuki (1968), A single precision inverse error function subroutine, Computation Center, University of Chicago. Strecok, Anthony J. (1968), On the calculation of the inverse of the error function, *Mathematics of Computation*, **22**, 144–158.

Kirk

Kirk, Roger E. (1982), *Experimental Design: Procedures for the Behavioral Sciences*, 2d. ed., Brooks/Cole Publishing Company, Monterey, Calif.

Kitagawa and Akaike

Kitagawa, G. and Akaike, H. (1978) A procedure for the modeling of non-stationary time series. *Ann. Inst. Statist. Math.*, Vol 30, B, 351-363.

Knuth

Knuth, Donald E. (1973), *The Art of Computer Programming*, Volume 3: *Sorting and Searching*, Addison-Wesley Publishing Company, Reading, Mass.

Knuth, Donald E. (1981), *The Art of Computer Programming*, Volume 2: *Seminumerical Algorithms*, 2d ed., Addison-Wesley, Reading, Mass.

Koch, Amara, and Atkinson

Koch, G.G., I.A. Amara, and S.S. Atkinson (1983), Mantel-Haenszel and related methods in analyzing ordinal categorical data with concomitant information, 39th Annual Conference on Applied Statistics, Newark, New Jersey.

Kotz and Johnson

Kotz, Samuel, and Norman L. Johnson (Editors) (1986), *Encyclopedia of Statistical Sciences*, **7**, John Wiley & Sons, New York.

Kronmal and Peterson

Kronmal, Richard A., and Arthur J. Peterson, Jr. (1979), On the alias method for generating random variables from a discrete distribution, *The American Statistician*, **33**, 214–218.

Kruskal

Kruskal, J.B. (1964), Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, *Psychometrika*, **29**, 1–27.

Kruskal, Young, and Seery

Kruskal J.B., F.W. Young, and J.B. Seery (1977), How to use KYST, a very flexible program to do multidimensional scaling and unfolding, Unpublished manuscript, Bell Telephone Laboratories, Murray Hill, New Jersey.

Kshirsagar

Kshirsagar, Anant M. (1972), *Multivariate Analysis*, Marcel Dekker, New York.

Lachenbruch

Lachenbruch, Peter A. (1975), *Discriminant Analysis*, Hafner Press, London.

Landis, Cooper, Kennedy, and Koch

Landis, J. Richard, Murray M. Cooper, Thomas Kennedy, and Gary G. Koch (1979), A computer program for testing average partial association in three-way contingency tables (PARCAT), *Computer Programs in Biomedicine*, **9**, 223–246.

Landis, Stanish, Freeman, and Koch

Landis, J. Richard, William M. Stanish, Jean L. Freeman, and Gary G. Koch (1976), A computer program for the generalized chi-square analysis of categorical data using weighted least squares (GENCAT), *Computer Programs in Biomedicine*, **6**, 196–231.

Lawless

Lawless, J.F. (1982), *Statistical Models and Methods for Lifetime Data*, John Wiley & Sons, New York.

Lawley and Maxwell

Lawley, D.N., and A.E. Maxwell (1971), *Factor Analysis as a Statistical Method*, 2d ed., Butterworth, London.

Learmonth et al.

Learmonth, G.P., and P.A. W. Lewis (1973a), *Naval Postgraduate School Random Number Generator Package LLRANDOM, NPS55LW73061A*, Naval Postgraduate School, Monterey, Calif.

Learmonth, G. P., and P. A. W. Lewis (1973b), Statistical tests of some widely used and recently proposed uniform random number generators, in *Computer Science and Statistics: 7th Annual Symposium on the Interface*, (edited by William J. Kennedy), Statistical Laboratory, Iowa State University, Ames, Iowa, 163–171.

Lee (1977)

Lee, S. Keith (1977), On the asymptotic variances of $\hat{\mu}$ terms in log-linear models of multidimensional contingency tables, *Journal of the American Statistical Association*, **72**, 412–419.

Lee (1980)

Lee, Elisa T. (1980), *Statistical Methods for Survival Data Analysis*, Lifetime Learning Publications, Belmont, Calif.

Lehmann

Lehmann, E.L. (1975), *Nonparametrics: Statistical Methods Based on Ranks*, Holden-Day, San Francisco.

Levenberg

Levenberg, K. (1944), A method for the solution of certain problems in least squares, *Quarterly of Applied Mathematics*, **2**, 164–168.

Levin and Marascuilo

Levin, J.R., and L.A. Marascuilo (1983), *Multivariate Statistics in the Social Sciences: A Researcher's Guide*, Wadsworth, Inc., California.

Lewis et al.

Lewis, P.A.W., A. S. Goodman, and J.M. Miller (1969), A pseudorandom number generator for the System/360, *IBM Systems Journal*, **8**, 136–146.

Lewis, P.A.W., and G.S. Shedler (1979), Simulation of nonhomogeneous Poisson processes by thinning, *Naval Logistics Quarterly*, **26**, 403–413.

Lilliefors

Lilliefors, H.W. (1967), On the Kolmogorov-Smirnov test for normality with mean and variance unknown, *Journal of the American Statistical Association*, **62**, 534–544.

Lilliefors, H.W. (1969), On the Kolmogorov-Smirnov test for the exponential distribution with mean unknown, *Journal of the American Statistical Association*, **64**, 387–389.

Lin and Bendel

Lin, Shang P., and Robert B. Bendel (1985), Generation of population correlation matrices with specified eigenvalues, *Applied Statistics*, **34**, 193–198.

Longley

Longley, James W. (1967), An appraisal of least-squares programs for the electronic computer from the point of view of the user, *Journal of the American Statistical Association*, **62**, 819–841.

Ljung and Box

Ljung, G.M., and G.E.P. Box (1978), On a measure of lack of fit in time series models, *Biometrika*, **65**, 297–303.

McCormack

McCormack, R.L. (1965), Extended tables of the Wilcoxon matched pair signed rank test, *Journal of the American Statistical Association*, **60**, 96–102.

McCullagh and Nelder

McCullagh, P., and J.A. Nelder, (1983), *Generalized Linear Models*, Chapman and Hall, London.

McKean and Schrader

McKean, Joseph W., and Ronald M. Schrader (1987), Least absolute errors analysis of variance, in *Statistical Data Analysis Based on the L_1 -Norm and Related Methods* (edited by Yadolah Dodge), North-Holland, Amsterdam, 297–305.

McKeon

McKeon, James J. (1974), F approximations to the distribution of Hotelling's T_0^2 , *Biometrika*, **61**, 381–383.

Maindonald

Maindonald, J.H. (1984), *Statistical Computation*, John Wiley & Sons, New York.

Mandel

Mandel, J. (1961), Non-additivity in two-way analysis of variance, *Journal of the American Statistical Association*, **69**, 859–866.

Marazzi

Marazzi, Alfio (1985), Robust affine invariant covariances in ROBETH, ROBETH-85 document No. 6, Division de Statistique et Informatique, Institut Universitaire de Medecine Sociale et Preventive, Lausanne.

March

March, D.L. (1972), Algorithm 434: *Exact probabilities for $R \times C$ contingency tables*, *Communications of the ACM*, **15**, 991–992.

Mardia et al.

Mardia, K.V. (1970), Measures of multivariate skewness and kurtosis with applications, *Biometrics*, **57**, 519–530.

Mardia, K.V., J.T. Kent, J.M. Bibby (1979), *Multivariate Analysis*, Academic Press, New York.

Mardia and Foster

Mardia, K.V. and K. Foster (1983), Omnibus tests of multinormality based on skewness and kurtosis, *Communications in Statistics A, Theory and Methods*, **12**, 207–221.

Marquardt

Marquardt, D. (1963), An algorithm for least-squares estimation of nonlinear parameters, *SIAM Journal on Applied Mathematics*, **11**, 431–441.

Marsaglia

Marsaglia, George (1964), Generating a variable from the tail of a normal distribution, *Technometrics*, **6**, 101–102.

Marsaglia, G. (1968), Random numbers fall mainly in the planes, *Proceedings of the National Academy of Sciences*, **61**, 25–28.

Marsaglia, G. (1972), The structure of linear congruential sequences, in *Applications of Number Theory to Numerical Analysis*, (edited by S. K. Zaremba), Academic Press, New York, 249–286.

Marsaglia, George (1972), Choosing a point from the surface of a sphere, *The Annals of Mathematical Statistics*, **43**, 645–646.

Marsaglia and Bray

Marsaglia, G. and T.A. Bray (1964), A convenient method for generating normal variables, *SIAM Review*, **6**, 260–264.

Marsaglia et al.

Marsaglia, G., M.D. MacLaren, and T.A. Bray (1964), A fast procedure for generating normal random variables, *Communications of the ACM*, **7**, 4–10.

Martinson and Hamdan

Martinson, E.O., and M.A. Hamdan (1972), Maximum likelihood and some other asymptotically efficient estimators of correlation in two way contingency tables, *Journal of Statistical Computation and Simulation*, **1**, 45–54.

McLeod and Bellhouse

McLeod, A.I., and D.R. Bellhouse (1983), A convenient algorithm for drawing a simple random sample, *Applied Statistics*, **32**, 182–184.

Mehta and Patel

Mehta, Cyrus R., and Nitin R. Patel (1983), A network algorithm for performing Fisher's exact test in $r \times c$ contingency tables, *Journal of the American Statistical Association*, **78**, 427–434.

Mehta, C.R. and N.R. Patel (1986a), Algorithm 643: FEXACT: A FORTRAN subroutine for Fisher's exact test on unordered $r \times c$ contingency tables, *ACM Transactions on Mathematical Software*, **12**, 154–161.

Mehta, C.R. and N.R. Patel (1986b), A hybrid algorithm for Fisher's exact test in unordered $r \times c$ contingency tables, *Communication in Statistics, Series A*, **15**, 387–404.

Merle and Spath

Merle, G., and H. Spath (1974), Computational experiences with discrete L_p approximation, *Computing*, **12**, 315–321.

Meyers

Meyers, Raymond H. (1971), *Response Surface Methodology*, Allyn and Bacon, Boston.

Miller

Miller, Rupert G., Jr. (1980), *Simultaneous Statistical Inference*, 2d ed., SpringerVerlag, New York.

Milliken and Johnson

Milliken, George A., and Dallas E. Johnson (1984), *Analysis of Messy Data: Volume 1*, Designed Experiments, Van Nostrand Reinhold, New York.

Moran

Moran, P.A.P. (1947), Some theorems on time series I, *Biometrika*, **34**, 281–291.

More and Hillstrom

More, J.J., B.S. Garbow, and K. E. Hillstrom (1980), *User Guide for MINPACK-1*, Argonne National Labs Report ANL-80-74, Argonne, Ill.

Morrison

Morrison, Donald F. (1976), *Multivariate Statistical Methods*, 2nd. ed. McGraw-Hill Book Company, New York.

Mosier

Mosier, C. (1939), Determining a simple structure when loadings for certain tests are known, *Psychometrika*, **4**, 149–162.

Muller

Muller, M.E. (1959), A note on a method for generating points uniformly on N-dimensional spheres, *Communications of the ACM*, **2**, 19–20.

Neter and Wasserman

Neter, John, and William Wasserman (1974), *Applied Linear Statistical Models*, Richard D. Irwin, Homewood, Ill.

Neter, Wasserman, and Kutner

Neter, John, William Wasserman, and Michael H. Kutner (1983), *Applied Linear Regression Models*, Richard D. Irwin, Homewood, Ill.

Noether

Noether, G.E. (1956), Two sequential tests against trend, *Journal of the American Statistical Association*, **51**, 440–450.

Null and Sarle

Null, Cynthia H., and Warren S. Sarle (1982), Multidimensional Scaling by Least Squares, in *Proceedings of the Seventh Annual SAS Users Group International Conference*, SAS Institute Inc., Cary, North Carolina.

Owen

Owen, D.B. (1962), *Handbook of Statistical Tables*, Addison-Wesley Publishing Company, Reading, Mass.

Owen, D.B. (1965), A special case of the bivariate non-central t -distribution, *Biometrika*, **52**, 437–446.

Pagano and Halvorsen

Pagano, Marcello, and Katherine Taylor Halvorsen (1981), An algorithm for finding the exact significance levels in $r \times c$ contingency tables, *Journal of the American Statistical Association*, **76**, 931–934.

Park and Miller

Park, Stephen K., and Keith W. Miller (1988), Random number generators: good ones are hard to find, *Communications of the ACM*, **31**, 1192–1201.

Patefield

Patefield, W.M. (1981), An efficient method of generating $R \times C$ tables with given row and column totals, *Applied Statistics*, **30**, 91–97.

Patefield and Tandy

Patefield, W.M. (1981), and Tandy D. (2000) Fast and Accurate Calculation of Owen's T-Function, *J. Statistical Software*, **5**, Issue 5.

Peixoto

Peixoto, Julio L. (1986), Testable hypotheses in singular fixed linear models, *Communications in Statistics: Theory and Methods*, **15**, 1957–1973.

Peto

Peto, R. (1973), Experimental survival curves for interval-censored data, *Applied Statistics*, **22**, 86–91.

Petro

Petro, R. (1970), Remark on Algorithm 347: An efficient algorithm for sorting with minimal storage, *Communications of the ACM*, **13**, 624.

Pike

Pike, M.C. (1966), A method of analysis of a certain class of experiments in carcinogenesis, *Biometrics*, **22**, 1–39.

Pillai

Pillai, K.C.S. (1985), Pillai's trace, in *Encyclopedia of Statistical Sciences, Volume 6*, (edited by Samuel Kotz and Norman L. Johnson), John Wiley & Sons, New York, 725–729.

Pregibon

Pregibon, Daryl (1981), Logistic regression diagnostics, *The Annals of Statistics*, **9**, 705–724.

Priestley

Priestley, M.B. (1981), *Spectral Analysis and Time Series*, Volumes 1 and 2, Academic Press, New York.

Prentice

Prentice, Ross L. (1976), A generalization of the probit and logit methods for dose response curves, *Biometrics*, **32**, 761–768.

Ramsey

Ramsey, James O. (1977), Maximum likelihood estimation in multidimensional scaling, *Psychometrika*, **42**, 241–266.

Ramsey, J.O. (1978), Confidence regions for multidimensional scaling analysis, *Psychometrika*, **43**, 145–160.

Ramsey, J.O. (1983), *Multiscale II Manual*, Unpublished manuscript, McGill University, Montreal, Quebec, Canada.

Rao

Rao, C. Radhakrishna (1973), *Linear Statistical Inference and Its Applications*, 2d ed., John Wiley & Sons, New York.

Robinson

Robinson, Enders A. (1967), *Multichannel Time Series Analysis with Digital Computer Programs*, Holden-Day, San Francisco.

Romesburg and Marshall

Romesburg, C., and K. Marshall (1974), *LIFE: A computer program for stochastic life table analysis*, US/IBP Desert Research Memorandum 74-68, Utah State University, Logan, Utah.

Royston

Royston, J.P. (1982a), An extension of Shapiro and Wilk's W test for normality to large samples, *Applied Statistics*, **31**, 115–124.

Royston, J.P. (1982b), The W test for normality, *Applied Statistics*, **31**, 176–180.

Royston, J.P. (1982c), Expected normal order statistics (exact and approximate), *Applied Statistics*, **31**, 161–165.

Sallas

Sallas, William M. (1988), Some Remarks on Algorithm AS 164. Least squares subject to linear constraints, *Applied Statistics*, **37**, 484–489.

Sallas, William M. (1990), An algorithm for an L_p norm fit of a multiple linear regression model, *American Statistical Association 1990 Proceedings of the Statistical Computing Section*, 131–136.

Sallas and Harville

Sallas, William M., and David A. Harville (1981), Best linear recursive estimation for mixed linear models, *Journal of American Statistical Association*, **76**, 860–869.

Sallas, William M., and David A. Harville (1988), Noninformative priors and restricted maximum likelihood estimation in the Kalman filter, in *Bayesian Analysis of Time Series and Dynamic Models* (edited by James C. Spall), Marcel Dekker, New York, 477–508.

Sallas and Lioni

Sallas, William M. and Abby M. Lioni (1988), Some useful computing formulas for the nonfull rank linear model with linear equality restrictions, IMSL Technical Report 8805, IMSL, Houston.

Satterthwaite

Satterthwaite, F.E. (1946), An approximate distribution of estimates of variance components, *Biometrics Bulletin*, **2**, 110–114.

Savage

Savage, I. Richard (1956), Contributions to the theory of rank order statistics|the twosample case, *Annals of Mathematical Statistics*, **27**, 590–615.

Scheffe

Scheffe, Henry (1959), *The Analysis of Variance*, John Wiley & Sons, New York.

Schiffman, Reynolds, and Young

Schiffman, Susan S., M. Lance Reynolds, and Forrest W. Young (1981), *Introduction to Multidimensional Scaling: Theory, Methods, and Applications*, Academic Press, New York.

Schmeiser et al.

Schmeiser, Bruce W., and A.J.G. Babu (1980), Beta variate generation via exponential majorizing functions, *Operations Research*, **28**, 917–926.

Schmeiser, Bruce W., and Ram Lal (1980), Squeeze methods for generating gamma variates, *Journal of the American Statistical Association*, **75**, 679–682.

Schmeiser, Bruce, and Voratas Kachitvichyanukul (1981), *Poisson Random Variate Generation*, Research Memorandum 81-4, School of Industrial Engineering, Purdue University, West Lafayette, Indiana.

Schmeiser, Bruce (1983), Recent advances in generating observations from discrete random variates, in *Computer Science and Statistics: Proceedings of the Fifteenth Symposium on the Interface*, (edited by James E. Gentle), North-Holland Publishing Company, Amsterdam, 154–160.

Schoneman

Schoneman, P.H. (1966), A generalized solution of the orthogonal Procrustes problem, *Psychometrika*, **31**, 1–10.

Scott

Scott, David W. (1976), Nonparametric probability density estimation by optimization theoretic techniques, *Technical Report 476* 131-1, Rice University, Houston, Texas.

Scott, Tapia, and Thompson

Scott, D.W., R.W. Tapia, and J.R. Thompson (1980), Nonparametric probability density estimation by discrete penalized-likelihood criteria, *The Annals of Statistics*, **8**, 820–832.

Searle

Searle, S.R. (1971), *Linear Models*, John Wiley & Sons, New York.

Seber

Seber, G.A.F. (1984), *Multivariate Observations*, John Wiley & Sons, New York.

Shampine

Shampine, L.F. (1975), Discrete least-squares polynomial fits, *Communications of the ACM*, **18**, 179–180.

Siegel

Siegel, Sidney (1956), *Nonparametric Statistics for the Behavioral Sciences*, McGraw-Hill, New York.

Silverman

Silverman, Bernard W. (1982), Kernel density estimation using the fast Fourier transform, *Applied Statistics*, **31**, 93–99.

Silverman, Bernard W. (1986), *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London.

Singleton

Singleton, R.C. (1969), Algorithm 347: An efficient algorithm for sorting with minimal storage, *Communications of the ACM*, **12**, 185–187.

Smirnov

Smirnov, N.V. (1939), Estimate of deviation between empirical distribution functions in two independent samples (in Russian), *Bulletin of Moscow University*, **2**, 3–16.

Snedecor and Cochran

Snedecor, George W., and William G. Cochran (1967), *Statistical Methods*, 6th. ed., Iowa State University Press, Ames, Iowa.

Sposito

Sposito, Vincent A. (1989), Some properties of L_p -estimators, in *Robust Regression: Analysis and Applications* (edited by Kenneth D. Lawrence and Jeffrey L. Arthur), Marcel Dekker, New York, 23–58.

Spurrier and Isham

Spurrier, John D. and Steven P. Isham (1985), Exact simultaneous confidence intervals for pairwise comparisons of three normal means, *Journal of the American Statistical Association*, **80**, 438–442.

Stablein, Carter, and Novak

Stablein, D.M, W.H. Carter, and J.W. Novak (1981), Analysis of survival data with nonproportional hazard functions, *Controlled Clinical Trials*, **2**, 149–159.

Stahel

Stahel, W. (1981), Robuste Schatzugen: Infinitesimale Opimalitat und Schatzugen von Kovarianzmatrizen, Dissertation no. 6881, ETH, Zurich.

Stephens

Stephens, M.A. (1974), EDF statistics for goodness of fit and some comparisons, *Journal of the American Statistical Association*, **69**, 730–737.

Stirling

Stirling, W.D. (1981), Algorithm AS 164. Least squares subject to linear constraints, *Applied Statistics*, **30**, 204–212 (see correction, page 357).

Stirling, W.D. (1981), Algorithm AS 169: An improved algorithm for scatter plots, *Applied Statistics*, **30**, 345–349.

Stoline

Stoline, Michael R. (1981), The status of multiple comparisons: simultaneous estimation of all pairwise comparisons in one-way ANOVA designs, *The American Statistician*, **35**, 134–141.

Swan

Swan, A.V. (1969a), Computing maximum-likelihood estimates for parameters of the normal distribution from grouped and censored data, *Applied Statistics*, **18**, 65–69.

Swan, A.V. (1969b), Maximum likelihood estimation from grouped and censored normal data, *Applied Statistics*, **18**, 110–114.

Takane and Carroll

Takane, Yoshio, and J. Douglas Carroll (1981), Nonmetric maximum likelihood multidimensional scaling from directional ranking of similarities, *Psychometrika*, **46**, 389–405.

Takane, Young, and De Leeuw

Takane, Y., F.W. Young, and J. De Leeuw (1977), Nonmetric individual differences multidimensional scaling: An alternating least-squares method with optimal scaling features, *Psychometrika*, **42**, 7–67.

Tanner

Tanner, Martin A. (1983), A note on the variable kernel estimator of the hazard function from censored data, *Annals of Statistics*, **11**, 994–998.

Tanner and Thisted

Tanner, Martin A., and Ronald A. Thisted (1982), Generation of random orthogonal matrices, *Applied Statistics*, **31**, 190–192.

Tanner and Wong

Tanner, Martin A., and Wing H. Wong (1983), The estimation of the hazard function from randomly censored data by the kernel method, *Annals of Statistics*, **11**, 989–993.

Tanner, Martin A., and Wing H. Wong (1984), Data-based nonparametric estimation of the hazard function with applications to model diagnostics and exploratory analysis, *Journal of the American Statistical Association*, **79**, 123–456.

Tapia

Tapia, R.A. (1974), A stable approach to Newton's method for general mathematical programming problems in R^n , *Journal of Optimization Theory and Applications*, **14**, 453–476.

Tapia and Thompson

Tapia, Richard A., and James R. Thompson (1978), *Nonparametric Probability Density Estimation*, Johns Hopkins University Press, Baltimore.

Tatsuoka

Tatsuoka, Maurice M. (1971), *Multivariate Analysis: Techniques for Educational and Psychological Research*, John Wiley & Sons, New York.

Taylor and Thompson

Taylor, Malcolm S., and James R. Thompson (1986), Data based random number generation for a multivariate distribution via stochastic simulation, *Computational Statistics & Data Analysis*, **4**, 93–101.

Thompson

Thompson, James R. (1989), *Empirical Model Building*, John Wiley & Sons, New York.

TIMSAC

TIMSAC, *TIME Series Analysis and Control* (TIMSAC-72, TIMSAC-74, TIMSAC-78, TIMSAC-84). Developed by the Institute of Statistical Mathematics, Japan.

Tucker and Lewis

Tucker, Ledyard, and Charles Lewis (1973), A reliability coefficient for maximum likelihood factor analysis, *Psychometrika*, **38**, 1–10.

Tukey

Tukey, J.W. (1949), One degree of freedom for nonadditivity, *Biometrics*, **5**, 232.

Tukey, John W. (1962), The future of data analysis, *Annals of Mathematical Statistics*, **33**, 1–67.

Tukey, John W. (1977), *Exploratory Data Analysis*, Addison-Wesley Publishing Company, Reading, Mass.

Turnbull

Turnbull, Bruce W. (1976), The empirical distribution function with arbitrary grouped, censored, and truncated data, *Journal of the Royal Statistical Society, Series B: Methodology*, **38**, 290–295.

Van de Geer

Van de Geer, John P. (1971), *Introduction to Multivariate Analysis for the Social Sciences*, W.H. Freeman and Company, San Francisco.

Velleman and Hoaglin

Velleman, Paul F., and David C. Hoaglin (1981), *Applications, Basics, and Computing of Exploratory Data Analysis*, Duxbury Press, Boston.

Verdooren

Verdooren, L.R. (1963), Extended tables of critical values for Wilcoxon's test statistic, *Biometrika*, **50**, 177–186.

Walker

Walker, A.J. (1974), New fast method for generating discrete random numbers with arbitrary frequency distributions, *Electronics Letters*, **10**, 127–128.

Wallace

Wallace, D.L. (1959), Simplified Beta-approximations to the Kruskal-Wallis H-test, *Journal of the American Statistical Association*, **54**, 225–230.

Weisberg

Weisberg, S. (1985), *Applied Linear Regression*, 2d ed., John Wiley & Sons, New York.

Weeks and Bentler

Weeks, David G., and P.M. Bentler (1982), Restricted multidimensional scaling models for asymmetric proximities, *Psychometrika*, **47**, 201–208.

Wilks

Wilks, S.S. (1935), On the independence of k sets of normally distributed statistical variables, *Econometrika*, **3**, 309–326.

Williams

Williams, J.S. (1962), A confidence interval for variance component, *Biometrika*, **49**, 278– 281.

Woodfield

Woodfield, Terry J. (1990), Some notes on the Ljung-Box portmanteau statistic, *American Statistical Association 1990 Proceedings of the Statistical Computing Section*, 155–160.

Young and Lewyckyj

Young, F.W., Y. Takane, and R. Lewyckyj (1978), Three notes on ALSCAL, *Psychometrika*, **43**, 433–435.

Young, Forrest W., and Rostyslaw Lewyckyj (1979), *ALSCAL-4 Users Guide*, second edition, Data Analysis and Theory Associates, Chapel Hill, North Carolina.

Product Support

Contacting Visual Numerics Support

Users within support warranty may contact Visual Numerics regarding the use of the IMSL Libraries. Visual Numerics can consult on the following topics:

- Clarity of documentation
- Possible Visual Numerics-related programming problems
- Choice of IMSL Libraries functions or procedures for a particular problem
- Evolution of the IMSL Libraries

Not included in these consultation topics are mathematical/statistical consulting and debugging of your program.

Consultation

Contact Visual Numerics Product Support by faxing 713/781-9260 or by emailing:

`support@houston.vni.com`.

The following describes the procedure for consultation with Visual Numerics.

1. Include your serial (or license) number
2. Include the product name and version number: IMSL Fortran Library Version 5.0
3. Include compiler and operating system version numbers
4. Include the name of the routine for which assistance is needed and a description of the problem

Index

A

acceptance/rejection method 1358
Akaike's Information Criterion
 AIC 677, 791, 797
alias method 1321
arguments, optional subprogram
 xv
ARMA model 674, 720, 723, 727,
 733, 741, 752, 822, 1390
ARMA Spectrum Estimation 822
array permutation 1429
ASCII collating sequence 1462
ASCII values 1458, 1459, 1460
autocorrelation function 697, 702
autocovariance function 679, 681
automatic workspace allocation 1522
autoregressive and moving average
 parameters 727
Autoregressive Integrated Moving
 Average 675
Autoregressive Moving Average
 Model 674
autoregressive parameters 720

B

backward difference operator 697
backward selection 237, 536
balanced complete experimental
 design 434, 469, 472
balanced incomplete block design
 415
balanced lattice design 415
balanced n -way classification model
 427
band Hermitian storage mode 1515
band storage mode 1513
band symmetric storage mode 1514
band triangular storage mode 1516
Bartlett-Priestley 684
basic statistics 56

basic uniform (0, 1) generators 1308
basic univariate statistics 18
Bayesian Time Series Analysis 803
beta distribution 1341
beta distribution function 1268
beta probability distribution function
 1265
Bhapkar V test 619
binary tree 1236
binomial distribution function 1246
binomial distributions 48, 1319
binomial probability function 1248
biserial correlation coefficient 376,
 377
biserial correlation coefficients 374
bivariate data 370
bivariate normal correlation
 coefficient 367
bivariate normal distribution
 function 1270
block design 415
Blom normal scores 30
Box-Cox (power) transformation 689
Box-Jenkins 673
Box-Jenkins forecasts 752
boxplots 1219
Bross' method 470

C

canonical correlation analysis 957,
 971
case diagnostics 83
case statistics 285
categorical data 557
Cauchy distribution 1344
cell
 frequencies 60
 means 60
 sums of squares 60
censored normal data 53
censored survival data 1095
centered 687
centered and padded realization 680
centered variables 296
character arguments 1459
character sequence 1463
character string 1464
chi-squared analysis 480, 491
chi-squared distribution 1343
chi-squared distribution function
 1271, 1274, 1275
chi-squared goodness-of-fit test 638
chi-squared statistic 497

chi-squared test 633
 Cholesky decomposition 353
 Cholesky factorization 479, 1483
 class of interest 1028
 classification variable 222, 374, 377
 classification variables 75
 cluster analysis 1005
 cluster membership 1015
 cluster sampling 1046
 Cochran Q test 625
 codiagonal band Hermitian storage
 mode 1519
 codiagonal band symmetric storage
 mode 1517
 coefficient of variation 24
 coefficients 301
 coherency spectrum 687
 cohort life tables 1121
 communalities 952
 confidence band information 1223
 confidence interval 469, 685
 confidence intervals 460
 contingency table 367, 480, 491,
 501, 509, 528, 548, 574
 continuous data 2, 1039, 1046, 1058,
 1257, 1260
 continuous distributions 634
 continuous variables 75
 contrast estimates 458
 correlation matrix 340, 353, 901,
 971, 1370
 cospectrum and quadrature spectrum
 686
 covariance matrix 353, 358
 covariates 395
 Cox and Stuart sign test 602
 CPU time 1465
 Crawford-Ferguson rotation 937
 cross periodogram 673, 687, 855,
 873
 cross-amplitude spectrum 687
 cross-correlation function 706, 711
 crossproducts 79, 176, 183, 296, 301
 cross-spectral analysis 686
 cross-spectral density 673, 862, 873,
 880, 889
 cross-spectral density function 688
 cubic interpolation 1185
 cumulative distribution function
 1295, 1298
 cumulative distribution function
 (CDF) 1223
 cumulative distribution functions
 1226
 cyclical trend 599

D

Daniell 683
 data set 1474
 data structures 477
 data tapering 687
 date 1467, 1468, 1470, 1471
 deleted residual 84
 diagnostic checking 674
 diagnostics 215
 dichotomous variable 374, 377
 differences of means 460
 direct oblimin rotation 924
 direct oblique rotation 936
 Dirichlet kernel 818
 discrete Fourier transform 680
 discrete uniform distribution 1340
 dissimilarity matrices 1154
 dissimilarity/similarity matrices 1150
 distances 1146
 domain of study 1051
 double precision xi
 DOUBLE PRECISION types xiii
 dummy variables 138
 d^2 test 664

E

empirical quantiles 38
 empirical tests 1310
 equamax rotation 918
 error handling xv, 1501
 errors
 informational 1500
 severity 1499
 terminal 1499
 estimation 673
 exact probabilities 501
 expected value 1487
 exponential distribution 646
 exponential distributions 1347
 exponential scores 26

F

F distribution function 1280, 1282
 F statistic 46
 factor analysis 909
 factor loading matrix 924, 927, 932
 factor score coefficient matrix 949
 factor score coefficients 944
 factor scores 949
 factor structure 942
 factorization 1480
 factor-loading matrix 921

fast Fourier transform 825, 855,
1180
Fast Fourier Transform 673
Faure 1407, 11, 5
Faure sequence 1306, 1405, 1406,
11, 5
Fejer kernel 820
filtering 687
Final Prediction Error
 FPE 794, 801
finite population 1401
Fisher exact probability 503
Fisher's exact test 486
Fisher's exact test probability 504
Fisher's linear discriminant analysis
 method 993
fitted general linear model 215
fitted regression model 189, 195
fixed interval 1180
fixed model 434
forecast 673
forecasting 674
forecasts
 GARCH 671, 745, 18, 6
forward selection 237, 536
fourth-degree polynomial criterion
 936
frequency distribution 387
frequency domain 679
frequency scale 685
frequency tables 3, 7, 14
 multiway 14
 one-way 3
 two-way 7
frequency tabulations 1
Friedman's test 620

G

gamma distribution function 1284,
1286
GARCH
 (Generalized Autoregressive
 Conditional Heteroskedastic)
 671, 745, 18, 6
Gaussian kernel estimates 1180
general continuous cumulative
 distribution function 1298,
 1301
general continuous distribution 1351,
 1354
general discrete distribution 1321,
 1324, 1328
general distributions 633

general linear model 75, 225
generalized feedback shift register
 method 1307, 1310
generalized inverse 1477
generalized linear model 1095
generalized linear models 557
generalized orthomax criterion 918
geometric distribution 1331
getting started xv
GFSR generator 1307
GFSR method 1307, 1310
Givens rotations 353
Givens transformations 117
Goodman and Kruskal coefficient
 489, 500
goodness-of-fit tests 633
Gray code 1408
Graybill's method 470
grouped data 2, 56
grouped normal data 53

H

Harris-Kaiser method 932
hazard rate estimation 1203
hazard rates 1114
HAZRD 1195
hierarchical cluster analysis 1005,
 1011
hierarchical cluster tree 1015
histogram 1208, 1210, 1213
horizontal histogram 1213
Hotelling's trace 79, 187
Huber's conjugate-gradient
 algorithm 360
hypergeometric distribution 1333
hypergeometric distribution function
 1250
hypergeometric probability function
 1252

I

identical population medians 616
image transformation matrix 940
impulse response weights 758
includance test 613
independence 954
initial estimates 1159
INTEGER types xiii
interval censoring 1073
inverse CDF method 1361
inverse prediction 103

iterative proportional-fitting
algorithm 512

J

jackknife residual 84

K

K cluster means 1023
Kalman filtering 780
Kaplan-Meier estimates 1063, 1068,
1073
Kappa analysis 478
Kappa statistic 485, 490, 496, 501
k-d tree 1490, 1494
K-dimensional sphere 1381
Kendall coefficient of concordance
379
Kendall test 384
Kendall's rank correlation
coefficient 383, 387
kernel function 1203
kernel functions 1188, 1195
kernel method 1176
K-means cluster analysis 1005, 1019
Kolmogorov-Smirnov goodness of
fit 1257, 1260
Kolmogorov-Smirnov test 633, 634,
654, 656
Kruskal-Wallis statistic 490, 500
Kruskal-Wallis test 616
k-sample trends test 627
kurtosis 24, 649

L

lack of fit 82
lack of fit test 189, 195, 815
Latin square design 422
least absolute values criterion 317
least squares 86, 107, 255, 673
least-squares estimates 733, 768,
774, 905
Lebesgue measure 1407
left censored 56
letter value summary 32
library subprograms xiv
Lilliefors test 646
linear discriminant function analysis
979
linear interpolation 1185
linear least-squares analysis 574

linear regression 72, 73, 99, 103,
107, 113, 117, 142, 231, 237,
317, 322, 333
linear regression model 89, 768
linear systems 1477, 1480
logarithmic distribution 1335
logistic linear model 557
loglinear model 509
log-linear models 513, 522, 528, 536
lognormal distribution 1356
low-discrepancy 1408

M

machine-dependent constants 1506
Mantel-Haenszel statistics 479
Mantel-Haenszel test 548
Mardia's multivariate measures 649,
653
matrices 1006, 1410, 1412, 1416,
1418, 1421
general 1512
Hermitian 1513
permutation 1431
printing 1410, 1412, 1416, 1418,
1421
real
rectangular 1410, 1412
rectangular 1512
symmetric 251, 1512
triangular 1513
matrix of dissimilarities 1006
matrix permutation 1431
matrix storage modes 1512
maximum 24
maximum likelihood 905
maximum likelihood estimate 741
McNemar test 485, 490, 496, 501
mean 23, 41, 53
mean vector 358
measures of association 487, 498
median 66
method of moments 673
method of moments estimates 720,
723
Mill's ratio 1489
minimal standard generator 1307
minimax criterion 333
minimum 24
missing value code 1435
missing values xvi, 86, 1150
mixed model 434
model estimates 513, 522
modified Bartlett 683

Monte Carlo applications 1310
 moving average parameters 723
 multichannel 673
 multichannel cross-correlation
 function 677
 multichannel time series 677, 711,
 768, 774
 multidimensional scaling 1166
 multidimensional scaling model
 1146
 multidimensional scaling models
 1159
 multinomial distribution 1377
 multiple linear regression model 317,
 322, 333
 multiplicative congruential generator
 1307
 multiplicative generator 1307
 multivariate data 60
 multivariate distribution 1373
 Multivariate Final Prediction Error
 MFPE 801
 multivariate general linear
 hypothesis 170, 176, 183
 multivariate general linear model 75,
 77, 128
 multivariate normal distribution
 1379
 multivariate normal variables 954
 multivariate time series 677
 multiway frequency tables 14

N

naming conventions xiii
 NaN xvi, 86, 1435
 nearest neighbor 1494
 nearest neighbor discrimination 997
 negative binomial distribution 1336
 nested random model 449
 network algorithm 504
 Newton-Raphson iterations 56
 Noether test 599
 noncentral chi-squared function 1278
 nonhomogeneous Poisson process
 1394
 nonlinear regression 304
 nonlinear regression model 79
 nonmissing observations 23
 nonnormalized spectral density 831,
 840, 845
 nonparametric hazard rate estimation
 1188, 1195

nonparametric probability density
 function estimation 1172, 1176
 nonseasonal ARMA 673
 nonseasonal ARMA model 733, 752
 nonuniform generators 1309
 normal distribution 646
 normal order statistic 1487
 normal populations 41
 normal scores 26
 normalized product-moment
 matrices 1154

O

oblique Promax rotation 927
 oblique rotation 932
 observations 83
 one-way classification model 392,
 395
 one-way frequency tables 3
 Optimal Controller Design 810
 optional argument xv
 optional data xv
 optional subprogram arguments
 xv
 order statistics 34
 ordinates of the density 1295, 1298
 orthogonal central composite design
 268
 orthogonal polynomials 272, 279,
 285, 292
 orthogonal Procrustes rotation 921
 orthogonal rotation 918
 outliers 215
 overflow xiv

P

padded 687
 padding 681
 page length 1428
 page width 1428
 pairs test 661
 parameter estimation 674
 parametric estimates 2
 parametric models 1114
 partial association statistics 528
 partial correlations 353
 Parzen 684
 Pearson chi-squared statistic 486
 penalized likelihood method 1172
 periodogram 673, 681, 825, 840,
 845, 851, 880, 889
 phase spectrum 687

Pillai's trace 79
 pivot 927
 Poisson distribution 50
 Poisson distribution function 1254
 Poisson linear model 557
 Poisson probability function 1255
 polar form 686
 polynomial curve 255
 polynomial model 74
 polynomial regression model 260,
 279, 285
 pooled variance-covariance matrix
 348
 population 1121
 population mean 1032, 1039, 1046,
 1050, 1054, 1058
 population proportion 1026, 1029
 power vector option 927
 preliminary estimates 727, 762
 prewhitening 687
 primary unit 1061
 principal components 901, 905
 printing 1428, 1501
 printing results xvi
 probability density function 1185
 probability plot 1229
 Probit linear model 557
 Procrustes rotation 927
 product-moment correlation 487, 498
 programming conventions xiv
 proportional fitting 509
 proportional hazards model 1078
 pseudorandom number generators
 633
 pseudorandom numbers 1316, 1317,
 1319, 1321, 1324, 1328, 1331,
 1333, 1335, 1336, 1340, 1341,
 1343, 1344, 1346, 1347, 1349,
 1351, 1354, 1356, 1358, 1359,
 1361, 1362, 1364, 1366, 1367,
 1368, 1373, 1377, 1379, 1394
 pseudorandom order statistics 1386,
 1388
 pseudorandom orthogonal matrix
 1370
 pseudorandom permutation 1398
 pseudorandom points 1381
 pseudorandom sample 1401
 pseudorandom sample of indices
 1399
 pseudorandom two-way table 1383

Q

quadratic discriminant function
 analysis 979
 quartimax rotation 918
 quasi-likelihoods 1188

R

random model 434, 449
 random number generators 1310
 random sample 1026, 1029, 1032,
 1039, 1050, 1054
 randomized block design 409
 randomized complete block design
 620
 range 24
 ranks 26, 30
 ranks and order statistics 2
 real rectangular matrix 1444, 1448
 REAL types xiii
 regression coefficients 164
 regression estimation 1032, 1039
 regression fit 152
 regression models 77
 regression parameters 142
 regressors 225
 related observations 625
 reordering matrices 1433
 replicates 189, 195
 required arguments xv
 reserved names 1521
 residuals 215
 response control 99
 right censored 56
 robust estimate 358
 Roy's maximum root 79, 187
 runs up test 657

S

sample correlation functions 674
 Savage scores 31
 scatter plot 1216
 search 1452, 1454, 1456
 second order response surface model
 301
 serial number 1472
 sets of points 1232
 Shapiro-Wilk W -test 644
 sign test 592
 simultaneous confidence intervals
 460
 single precision xi
 skewness 23, 649

- sorting 1439, 1440, 1442, 1443,
1444, 1448, 1452, 1454, 1456
- sparse matrix storage mode 1520
- Spearman correlation 487, 498
- specified weights 292
- spectral analysis 679
- spectral density 673, 681, 831, 851,
873, 889
- spectral window 673, 683, 831, 840,
862, 873
- squares 296
- stable distribution 1362
- Stahel's algorithm 360
- standard errors 487, 498
- standard exponential distribution
1346
- standard gamma distribution 1349
- standard normal (Gaussian)
distribution function 1262,
1264
- standard normal distribution 1358,
1359, 1361, 1386
- standardized factor residual
correlation matrix 952
- starting values 1311
- statespace model 780
- stationary stochastic process 749
- stationary time series 697, 702, 706,
825, 831, 840, 845, 851, 855,
862
- statistics
 - basic univariate 18
 - univariate summary 2
- statistics for inferences 1026, 1029,
1032, 1039, 1046, 1050, 1054,
1058
- stem-and-leaf plot 1221
- stepwise selection 237, 536
- stratified random sample 1029
- stratified samples 1063, 1068
- stress criteria 1166
- Student's t distribution 1364
- Student's t distribution function
1288, 1289, 1291, 1293
- Student-Newman-Keuls method 469
- Student-Newman-Keuls multiple
comparison test 467
- subprograms
 - library xiv
 - optional arguments xv
- summary statistics 279
- sums of squares 79, 176, 183, 458
- survival probabilities 1063, 1068,
1073, 1114
- symmetric submatrix 251

T

- t* statistic 46
- table lookup method 1328
- target matrix 927
- tests for randomness 633
- tetrachoric correlation coefficient
370
- theoretical CDF 1223
- tie statistics 607
- time 1466
- time domain methodology 674
- time event data 1078
- time interval 684
- time series 673, 674, 693, 711, 768,
774, 815, 825, 831, 840, 845,
851, 855, 862, 873, 880, 889,
1390
- transfer function 673
- transfer function model 675, 758,
762
- transformation 674
- transformations 84
- trends in dispersion and location 602
- triangular distribution 1366
- triplets test 667
- Tukey 684
- Tukey normal scores 31
- Turnbull's generalized Kaplan-Meier
estimates 1073
- two-way balanced design 409
- two-way frequency tables 7
- two-way table 66

U

- uncentered variables 301
- uncertainty coefficient 489, 500
- underflow xiv
- uniform (0, 1) distribution 1316,
1317, 1388
- uniform (0, 1) numbers 1310
- unique values 222
- unit circle 1381
- univariate density 1180
- univariate summary statistics 2
- univariate time series 677, 815
- user errors 1499
- user interface xi
- using library subprograms xiv

V

- Van der Waerden normal scores 31
- variance 23, 41, 53, 942

variance-covariance matrix 113, 164,
340, 348, 901, 971
varimax rotation 918
version 1472
vertical histogram 1208, 1210
von Mises distribution 1367

W

Weibull distribution 1368
weights 673
Wiener filter coefficients 774
Wiener forecast function 677
Wiener forecast operator 749
Wilcoxon rank sum test 609
Wilcoxon signed rank test 595
Wilcoxon two-sample test 618
Wilks' lambda 79, 186