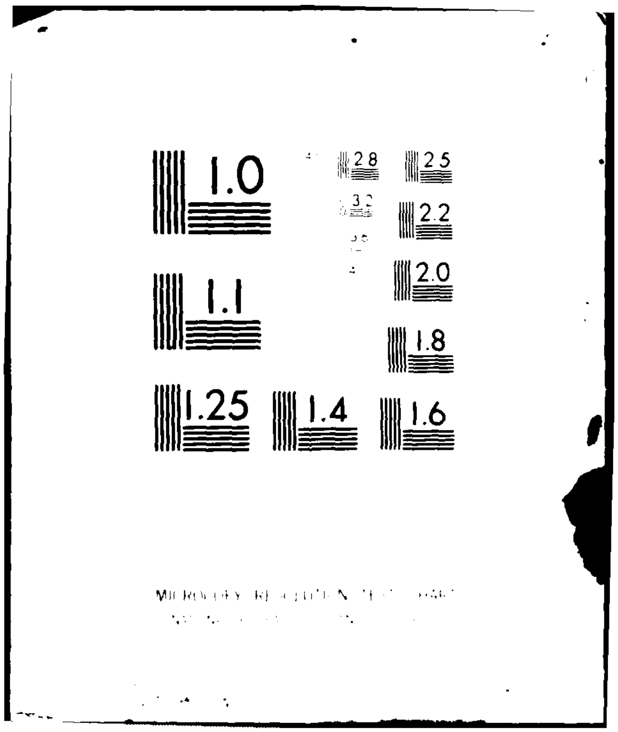
AD-A098 145   PITTSBURGH UNIV PA INST FOR STATISTICS AND APPLICATIONS   F/G 12/1
USER'S MANUAL: MONCOR--A PROGRAM TO COMPUTE CONCORDANT AND OTHE--ETC(U)
JAN 81   G KIMELDORF, J H MAY, A R SAMPSON   F49620-79-C-0161
UNCLASSIFIED   CPSS-TR-81   AFOSR-TR-81-0377   NL

1 of 1
AD A
098145

END
DATE
FILMED
5-81
DTIC


1.0
2.8
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

COMPUTER PROGRAMS FOR STATISTICAL SCIENTISTS (CPSS) SERIES

USER'S MANUAL: MONCOR--A Program to Compute Concordant and Other Monotone Correlations

by

George Kimeldorf[1]

Jerrold H. May

Allan R. Sampson[2]

University of Texas at Dallas

University of Pittsburgh

University of Pittsburgh

January 1981

CPSS Technical Report No. 81-1

Institute For Statistics and Applications

Department of Mathematics and Statistics

University of Pittsburgh

Pittsburgh, PA 15260

[1]The work of this author was supported by the National Science Foundation under Grant MCS-8002152.

[2]The work of this author is sponsored by the Air Force Office of Scientific Research, Air Force Systems Command, under Contract F49620-79-C-0161.

UNCLASSIFIED

(19) REPORT DOCUMENTATION PAGE

1. REPORT NUMBER

(18) AFOSR-TR-81-0377

2. GOVT ACCESSION NO.

AD-A098 145

4. TITLE (and Subtitle)

USER'S MANUAL: MONCOR--A Program to Compute Concordant and Other Monotone Correlations

(9) INTERIM

UP Stat. Report No. 81-1

7. AUTHOR(s)

(10) George Kimeldorf, Jerrold H. May and Allan R. Sampson

F49620-79-C-0161

NSF-M[illegible]-0[illegible]

9. PERFORMING ORGANIZATION NAME AND ADDRESS

University of Pittsburgh
Department of Mathematics & Statistics
Pittsburgh, PA 15260

(16) 2304/A5
61102F

11. CONTROLLING OFFICE NAME AND ADDRESS

AFOSR/NM
Bolling AFB, DC 20332

(12) 15

(11) Jan[redacted]81

10

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

(14) TSS-TR-81-1

UNCLASSIFIED

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

User's Manual, interactive program, correlation, monotone correlation, concordant monotone correlation, discordant monotone correlation, isotonic [illegible], ordinal contingency tables, nonlinear optimization.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A User's Manual is provided for the interactive FORTRAN program [illegible] which computes the concordant monotone correlation, discordant [illegible] correlation, isoconcordant monotone correlation, isodiscordant [illegible] correlation and their associated monotone variables. Data input are ordinal contingency tables in matrix form.

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

411315

USER'S MANUAL:

MONCOR--A Program

to Compute Concordant and Other

Monotone Correlations

by

George Kimeldorf, Jerrold H. May, and Allan R. Sampson

University of Texas at Dallas, University of Pittsburgh, and

University of Pittsburgh

ABSTRACT

A User's Manual is provided for the interactive FORTRAN program MONCOR which computes the concordant monotone correlation, discordant monotone correlation, isoconcordant monotone correlation, isodiscordant monotone correlation and their associated monotone variables. Data input are ordinal contingency tables in matrix form.

Key Words and Phrases: User's Manual, interactive program, correlation, monotone correlation, concordant monotone correlation, discordant monotone correlation, isoscaling, ordinal contingency tables, nonlinear optimization.

USER'S MANUAL: MONCOR--

A Program to Compute

Concordant and Other Monotone Correlations

by

George Kimeldorf[1], Jerrold H. May, and Allan R. Sampson[2]

University of Texas at Dallas, University of Pittsburgh, and

University of Pittsburgh

1. General Background

MONCOR is a set of routines to compute monotone correlation measures for two dimensional probability distributions of random variables X and Y given by matrices of the form

$$\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1J} \\ p_{21} & p_{22} & \cdots & p_{2J} \\ \vdots & \vdots & & \vdots \\ p_{I1} & p_{I2} & \cdots & p_{IJ} \end{bmatrix},$$

where $p_{ij} = \text{Prob}(X = x_i, Y = y_j)$, $i = 1, \ldots, I$ and $j = 1, \ldots, J$. These probabilities may be theoretical probability distributions or may be estimates derived from ordinal contingency tables.

The general approach followed in MONCOR is based upon the algorithm given in Kimeldorf, May and Sampson (1980) with a more theoretical development in

[1]The work of this author was supposed by the National Science Foundation under Grant MCS-8002152.

[2]The work of this author is sponsored by the Air Force Office of Scientific Research, Air Force Systems Command, under Contract F49620-79-C-0161.

Kimeldorf and Sampson (1978). MONCOR computes the concordant and discordant monotone correlations, and their isoscale versions, by formulating a linearly constrained nonlinear programming problem whose optimum value is the desired monotone correlation measure. A specially modified form of the nonderivative nonlinear programming algorithm QRMNEW (see May (1979)) is used to solve the nonlinear program.

2. Program Structure and Coding Considerations

2.1 Structure

MONCOR consists of two basic components: routines specific to monotone correlation and routines adapted from the QRMNEW program. A brief overview of the calling sequences is given in Figure 2.1. The primary goal throughout the coding of MONCOR was simplicity of presentation. The components of QRMNEW adapted for MONCOR are written in part in a non-structured style, but with readability enhanced by comment lines.

Since MONCOR was designed to be used in an interactive mode for small problems, dense matrix storage is utilized throughout. Array storage is made available to all subroutines through labelled common only. All dimensions are explicitly set in each subroutine. MONCOR version 2.3 is dimensioned so as to handle probability matrices of dimension up to 20 x 20. The source listing comments indicate how to redimension the arrays for larger problems.

The original source code for MONCOR consists of 1932 statements written in DEC FORTRAN-10 and is implemented on the University of Pittsburgh's DEC 1099.

2.2 Programming Considerations

(a) Types. Double precision arithmetic is used throughout, and both variables and arrays are set by an IMPLICIT statement at the beginning of each routine. Without exception, the type of a variable is given by the first character of its name, as follows:

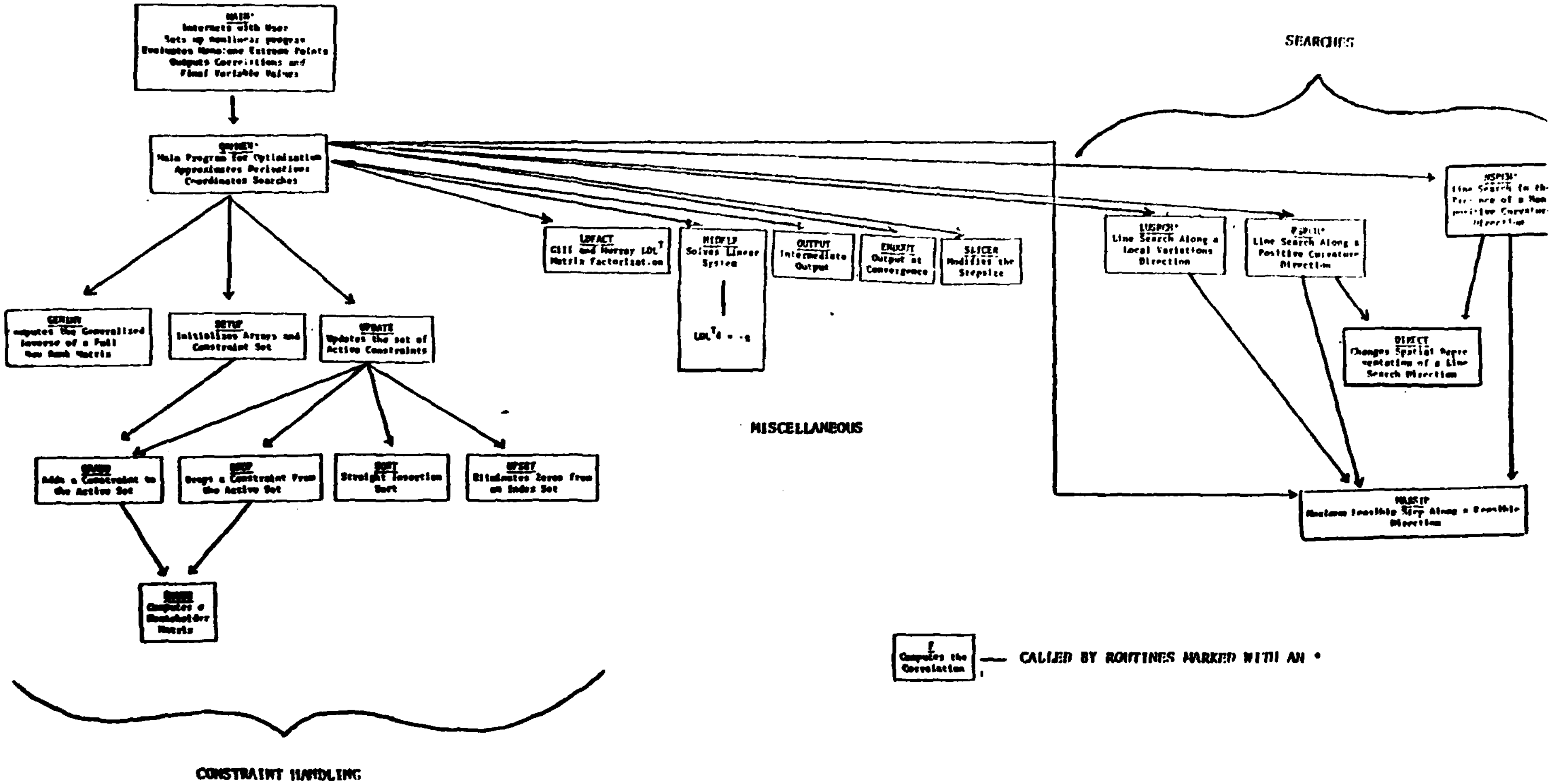


Figure 2.1

Calling Structure For MONCOR Routines

A-H, O-Z	Double Precision
I-N	Integer.

No half word integer (INTEGER *2) variables are used. No logical variables appear in MONCOR.

(b) COMMON. Blank COMMON is not used. There are 21 labelled common blocks, containing all the arrays and important scalar variables.

(c) EQUIVALENCE. No EQUIVALENCE statements are used.

(d) DATA. No DATA statements are used.

(e) Arrays. There are both one-dimensional and two-dimensional arrays, none of variable length. Declarations of the form REAL A(1) are never used.

(f) Subscripts. Subscript expressions of the form X(M + J) as well as X(INDEX (J)) are often used.

(g) Avoidance of Trouble Spots. The following FORTRAN syntactical items are not used:

Arithmetic IF; Assigned GO TO; BACKSPACE; Blank COMMON; BLOCK DATA; COMPLEX; ENDFILE; ENTRY; EQUIVALENCE; EXTERNAL; FIND; FUNCTION; NAMELIST; Nonstandard RETURN; PAUSE; PRINT; PUNCH; TIME; Uncommitted READ and WRITE.

2.3 Conversion of MONCOR to Other Machines

If double precision is not required, the IMPLICIT statements should be deleted, and the names of the library functions (DABS, DMAX1, DMIN1, DLOG, DSQRT, etc.) should be changed to their real equivalents (ABS, AMAX1, AMIN1, ALOG, SQRT, etc.).

If double precision is required but the IMPLICIT statement is not allowed, all variables starting with A-H and O-Z must be typed explicitly. Such a conversion will require many lines of otherwise unnecessary code, and will also be prone to error, since FORTRAN compilers do not usually issue warnings for undeclared variables.

In FORMAT statements, apostrophes are used to delimit strings. Certain strings in the main program contain the character *.

The main program calls the FORTRAN-10 basic external function RAN, which returns a [0,1] random number. The call to RAN may be modified to accomodate any other random number generator.

3. Program Description

In all sample MONCOR outputs, those items inputted by the user are underlined. On this output, references to material in Section 3 are typed in the left margin.

3.1 Example

To describe the use of MONCOR, the British Mobility data example is employed. (See Kimeldorf, May and Sampson; the original data are in Glass and Hall (1954, p. 183).) The estimated probability matrix $\{p_{ij}\}$ derived from Table 5.1 of Kimeldorf, May and Sampson is given in Table 3.1.

Table 3.1

Estimated Probability Matrix
British Mobility Data

Father's Occupational Status		Son's Occupational Status: S1	S2	S3	S4	S5
	S1	.014	.013	.002	.005	.002
	S2	.008	.050	.024	.044	.016
	S3	.003	.022	.031	.064	.027
	S4	.004	.043	.053	.205	.128
	S5	.001	.012	.021	.091	.117

In this example I = J = 5.

3.2 Data Input

The program requires input in the following order:

(a) the number of rows of $\{p_{ij}\}$: I,

(b) the number of columns of $\{p_{ij}\}$: J,

(c) the device number that the matrix is to be read in from (entering a 5 allows reading from terminal),

(d) the values for $\{p_{ij}\}$ read in a row at a time (it is assumed $\max(I,J) \leq 20$).

After $\{p_{ij}\}$ is inputted,

(e) the matrix is printed by MONCOR under the heading INPUT MATRIX.

An example of this procedure is given for the British Mobility data.

```
3.2(a) HOW MANY ROWS IN YOUR MATRIX?           5
3.2(b) HOW MANY COLUMNS IN YOUR MATRIX?        5
3.2(c) ON WHAT DEVICE IS YOUR MATRIX?  ENTER 5 IF YOU  WISH TO TYPE IT IN NOW.      5
       ENTER THE MATRIX A ROW AT A TIME. MAX 20 NOS. PER LINE
      >.014 .013 .002 .005 .002
      >.008 .050 .024 .044 .016
3.2(d)>.003 .022 .031 .064 .027
      >.004 .043 .053 .205 .128
      >.001 .012 .021 .091 .117

                          INPUT MATRIX

       .0140 .0130 .0020 .0050 .0020
       .0080 .0500 .0240 .0440 .0160
3.2(e) .0030 .0220 .0310 .0640 .0270
       .0040 .0430 .0530 .2050 .1280
       .0010 .0120 .0210 .0910 .1170

                          END OF INPUT MATRIX
```

3.3 Output

(a) After the matrix $\{p_{ij}\}$ has been output by MONCOR, the user must specify which <u>one</u> measure of monotone correlation is desired. The user needs only to enter one number (1, 2, 3 or 4) according to the MONCOR prompts.

(b) For each monotone measure calculation, the user has the <u>option</u> of calling for

(i) intermediate point output during the optimization process

(ii) inputting a starting point for the optimization process.

Responses to MONCOR questions concerning (i) or (ii) are YES or NO.

For the remainder of the output description in Subsection 3.3, it is assumed that neither option (i) nor (ii) is requested.

(c) MONCOR then evaluates the correlation at the monotone extreme points (see Section 4, Kimeldorf, May and Sampson). The number of such monotone extreme points is given by MONCOR (if the concordant or discordant monotone correlation is requested, the number of monotone extreme points is (I-1)(J-1); if the iso-concordant or isodiscordant monotone correlation is requested, the number of monotone extreme points is (I-1)).

(d) MONCOR computes the largest correlation among these monotone extreme points, and

(e) prints one monotone extreme point at which this maximum is achieved.

(f) MONCOR then prompts the user to enter a five digit number which is used to warm up the FORTRAN-10 basic external function random number generator RAN. RAN is called that number of times before being used to generate the ten random starting points (see Kimeldorf, May and Sampson (Section 4)).

(g) The estimated monotone correlation measure is given based upon the 10 random starting point calculations as well as the monotone extreme points (Kimeldorf, May and Sampson (Section 4)).

(h) A point (monotone variable) at which the monotone correlation is achieved is printed by MONCOR. MONCOR, without loss of generality, sets X(1) = Y(1) = 0 and X(I) = Y(J) = 1. If an iso-measure is selected, only the X monotone variable is printed. (The Y monotone variable is identical by definition.)

(i) MONCOR then asks if the user wants to use the inputted $\{p_{ij}\}$ matrix to calculate other measures of monotone correlation. If the user responds YES, it begins at 3.3(a) again. If the user responds NO, an opportunity is given to input another $\{p_{ij}\}$, (see 3.2(a)) or to terminate MONCOR.

For illustration, the calculation of the isoconcordant monotone correlation for the British Mobility data is given below.

```
3.3(a)      WHICH ONE OF THE FOLLOWING DO YOU WISH TO SEE?
              1=CONCORDANT MONOTONE CORRELATION
              2=DISCORDANT MONOTONE CORRELATION
              3=ISOCONCORDANT MONOTONE CORRELATION
              4=ISODISCORDANT MONOTONE CORRELATION
            >3


            *********************************************** ISOCONCORDANT MONOTONE CORRELATION ******
3.3(b)(i)   DO YOU WISH TO SEE INTERMEDIATE POINT OUTPUT?          NO
3.3(b)(ii)  DO YOU WISH TO INPUT A STARTING POINT?                 NO
3.3(c)      BASED ON     4  MONOTONE EXTREME POINTS, THE
3.3(d)      MAXIMUM OF CORRELATIONS AMONG MONOTONE EXTREME POINTS IS        0.49626150     AT
3.3(e)       X( 1)= .00000   X( 2)=1.00000   X( 3)=1.00000   X( 4)=1.00000   X( 5)=1.00000  ..


3.3(f)      ENTER A 5 DIGIT INTEGER       >48147
3.3(g)      ESTIMATED ISOCONCORDANT MONOTONE CORRELATION IS       0.49682067   AT
3.3(h)       X( 1)= .00000   X( 2)= .62728   X( 3)= .84644   X( 4)= .92409   X( 5) 1.00000  ..
3.3(i)       ANOTHER RUN WITH THE SAME MATRIX?        >YES
```

3.4 Option: Inputting a Starting Point

Unless otherwise specified, MONCOR generates 10 random starting points from which the optimization procedure starts. However, the user has the option of inputting a monotone starting point by replying YES to the question "DO YOU WISH TO INPUT A STARTING POINT?" The user must then input X(2), ..., X(I-1) and Y(2), ..., Y(J-1). (If an iso-measure is requested, then only X(2), ..., X(I-1) must be input.) MONCOR sets S(1) = Y(1) = 0 and X(I) = Y(J) = 1. If this option is employed, no monotone extreme points are evaluated (see 3.3(c)) and no random starting points are used (see 3.3(g), 3.3(f)).

For the British Mobility data, the following illustrates inputting a pair of starting points for the concordant monotone correlation.

```
       DO YOU WISH TO INPUT A STARTING POINT?            YES
       X( 2)=======>.25
       X( 3)=======>.5
3.4    X( 4)=======>.75
       Y( 2)=======>.25
       Y( 3)=======>.5
       Y( 4)=======>.75
```

3.5 Option: Requesting Intermediate Point Output

Unless otherwise specified MONCOR does not print out the intermediate steps in the iterations of the nonlinear optimization (see Section 4 of Kimeldorf, May and Sampson). However, the user has the option of requesting intermediate output by replying YES to the question "DO YOU WISH TO SEE INTERMEDIATE POINT OUTPUT?"

If the user has <u>not</u> inputted a starting point, MONCOR will print the intermediate evaluations for each of the 10 random starting points. If the user has inputted a starting point, MONCOR will print out the intermediate evaluations based upon the given starting point.

For each iteration, MONCOR outputs

(a) the iteration number (with iteration number 1 corresponding to the random or inputted starting point),

(b) the number of evaluations, which is the number of times QRMNEW has evaluated the objective function cumulatively up to the given iteration.

(c) the correlation based upon the current iteration,

(d) the gradient which measures the degree of the local steepness at that point (see a description of QRMNEW in May for details),

(e) the coordinates of the current point (X(1), X(I), Y(1), Y(J) for the current point are suppressed, and for iso-measures, Y(2), ..., Y(J-1) are not printed).

(f) When the algorithm stops, MONCOR prints "OPTIMAL SOLUTION FOUND" and provides the optimizing point, the value for the final iteration, and the total cumulative number of function evaluations (see 3.5(b)).

Again for the British Mobility Data, the intermediate point output for the isoconcordant monotone correlation is given below for random point number 2 (no starting point was input).

RANDOM POINT NUMBER 2

3.5(a)	ITN	EVAL	CORR	GRAD	X(2)	X(3)	X(4)
3.5(b)	1	10	0.4597	3.8D-01	.529	.644	.775
3.5(c)	2	20	0.4796	1.7D-01	.776	.977	1.000
3.5(d)	3	30	0.4966	3.8D-02	.626	.834	.913
3.5(e)	4	40	0.4960	7.6D-04	.628	.846	.924

3.5(f)

XXXXXXXX OPTIMAL SOLUTION FOUND

X(1)= .00000 X(2)= .62778 X(3)= .84644 X(4)= .92409 Y(5) 1.00000 Y(

CORRELATION COEFFICIENT= 0.496821 50 FUNCTION EVALUATIONS

3.6 Comment

A User who wishes to compute the (Pearson) correlation for a given set of values of X(2), ..., X(I-1) and Y(2), ..., Y(J-1) can use Options 3.4 and 3.5, so that the first iteration evaluation will provide the desired correlation. For example, to compute Spearman's rank correlation, set X(i) = (i-1)/(I-1) for i = 2, ..., I-1, and Y(j) = (j-1)/(J-1) for j = 2, ..., J-1.

REFERENCES

Glass, D.V. and Hall, J.R., (1954). "Social Mobility in Britain: A study of inter-generation changes in status." In Social Mobility in Britain. Ed. by D.V. Glass. Routledge and Kogan Paul Ltd., London.

Kimeldorf, G., May, J.H., and Sampson, A.R., (1980). "Concordant and discordant monotone correlations and their evaluation by nonlinear optimization." Technical Report.

Kimeldorf, G., and Sampson, A.R., (1978). "Monotone dependence." Annals of Statistics 6, 895-903.

May, J.H., (1979). "Solving nonlinear programs without using analytic derivatives." Operations Research 27, 547-484.

DATE
FILMED
-8