

A FORTRAN IV program for trend analysis with unequal intervals and unequal ns

WILLIAM P. DUNLAP

Tulane University, New Orleans, Louisiana 70118

When all levels of a scaled variable are separated by equal intervals (e.g., 0-, 1-, 2-, or 3-mg drug dosage) and when sample sizes at the various levels are equal, trend analysis is a fairly simple and well documented procedure (e.g., Edwards, 1968; Grant, 1956), for which tables of trend coefficients are widely available; a very complete set can be found in Fisher and Yates (1963). On the other hand, when the levels of the scaled variables are separated by unequal intervals (e.g., 0, 2, 5, 10 mg), the tabled coefficients may not be used; instead, coefficients must be computed for each particular problem. Also, when unequal sample sizes occur at the various levels of the scaled variable, the trend coefficients must be adjusted to maintain orthogonality, using the rule that

$$\sum_i n_i a_i b_i = 0,$$

where a_i and b_i are corresponding coefficients for level i of the scaled variable from two different sets of trend coefficients, and n_i is the sample size at level i .

Gaito (1965) has published a procedure for finding coefficients with unequal intervals and unequal ns; however, the solution involves sets of simultaneous equations that are quite involved for the higher order polynomials. In the present program, Gaito's equations are restated in matrix form so the solutions can be obtained using matrix inversion. The program solves and prints orthogonal trend coefficients that have been normalized to 1. Normalization to 1 simplifies the calculations involved if the coefficients are used in hand calculation when there are equal sample sizes. The user may preset the maximum order of the polynomials computed or all possible sets of coefficients may be calculated and printed. As a further option, the program will calculate sums of squares for the various trend components from either totals or means.

Computer and Language. The program is written in standard FORTRAN IV and runs on an IBM 7044. The matrix inversion subroutine used is MATINV, published by Cooley and Lohnes (1971); however, any matrix inversion subroutine that replaces a matrix by its inverse could be used. The program should run with little modification on any computer with a FORTRAN IV compiler.

Availability. A listing of the program may be obtained at no cost by writing William P. Dunlap, Department of Psychology, Tulane University, New Orleans, Louisiana 70118. Instructions for using the program are included as comment statements at the beginning of the program.

NOTE

1. This relationship insures orthogonality only when coefficients are applied to treatment totals. In the present program if means are input, they are converted to totals before calculating sums of squares. The correct restriction when coefficients are to be applied directly to means is $\sum_i a_i b_i / n_i = 0$. Therefore, the coefficients generated by the present program should not be applied to means, but only to totals if there is unequal n.

Preparation and testing of this program was supported by the Tulane University Computer Center.

Table 1

```
C THIS PROGRAM CALCULATES ORTHOGONAL TREND COEFFICIENTS AND SUMS OF SQUARES
C FOR UNEQUAL INTERVALS AND UNEQUAL SAMPLE SIZES. FOR A DESCRIPTION OF
C SOLUTION EQUATIONS IN NON MATRIX FORM AND A WORKED EXAMPLE SEE GAITO,
C PSYCHOLOGICAL BULLETIN, 1965, 63, 125-127.
C INPUT CARD 1
C COL 1 - 3 NUMBER OF LEVELS OF VARIABLE
C COL 4 - 6 MAXIMUM ORDER OF POLYNOMIAL
C COL 7 - 9 NUMBER OF SETS OF TOTALS (OR MEANS) TO FOLLOW
C COL 12 0 = MEANS, 1 = TOTALS
C INTERVAL CONSTANTS FOR THE SCALED VARIABLE ARE INPUT NEXT USING
C THE FORMAT (8F10.0)
C SAMPLE SIZES ARE INPUT NEXT AGAIN USING (8F10.0)
C OPTIONAL SETS OF TOTALS (OR MEANS) ARE INPUT LAST USING (8F10.0)
C THIS PROGRAM REQUIRES SUBROUTINE MATINV (COOLEY AND LOHNES, 1971)
C OR A SUBSTITUTE THAT WILL REPLACE A MATRIX BY ITS INVERSE
DIMENSION X(50,50),A(50,50),T(50),C(50),F(50),CL(50),AS(50)
1 READ(5,2)K,MORD,NT,MOT
2 FORMAT(4I3)
3 IF(NT.EQ.0)CALL EXIT
4 IF(MORD.GT.K-1.OR.MORD.EQ.0)MORD=K-1
5 READ(5,3)(C(I),I=1,K)
6 READ(5,3)(F(I),I=1,K)
7 FORMAT(8F10.0)
8 KQ=K+1
9 KT=K+2
10 DO 4 J=1,K
11 A(1,J)=1.0
12 DO 12 L=1,MORD
13 DO 5 I=1,K
14 DO 5 J=1,K
15 X(1,J)=0.0
16 IF(NT.EQ.0)X(I,J)=1.0
17 CONTINUE
18 KM=K+L
19 DO 6 I=KQ,KN
20 DO 6 J=KQ,KN
21 X(I,J)=0.0
22 DO 7 I=1,K
23 X(I,KQ)=1.0
24 IF(KQ.EQ.KN)GOTO9
25 DO 8 I=1,K
26 DO 8 J=KT,KN
27 X(I,J)=C(I)**(J-K-1)
28 DO 10 I=KQ,KN
29 DO 10 J=1,K
30 M=I-K
31 X(I,J)=F(J)**M
32 CALL MATINV(X,KN,DET)
33 DO 11 I=1,K
34 A(1,I)=0.0
35 DO 11 J=1,K
36 A(1+I,J)=A(1+I,J)+X(I,J)**C(J)**M
37 CONTINUE
38 DO 13 I=1,MORD
39 C(I)=0.0
40 DO 13 J=1,K
41 C(I)=C(I)+A(1+I,J)**2
42 DO 14 I=1,MORD
43 C(I)=SQRT(C(I))
44 DO 14 J=1,K
45 A(1+I,J)=A(1+I,J)/C(I)
46 WRITE THE COEFFICIENTS
47 WRITE(6,15)
48 FORMAT(1H0,20X,29HNORMALIZED TREND COEFFICIENTS/16X,37HROW 1 LINE
49 1AR, ROW 2 QUADRATIC, ETC.//)
50 DO 16 I=1,MORD
51 WRITE(6,17)1,(A(1+I,J),J=1,K)
52 FORMAT(1H ,12,8HOTH ORDER,8F15.6/11X,8F15.6/11X,8F15.6)
53 IF(NT.EQ.0)GOTO1
54 DO 24 M=1,NT
55 READ(5,3)(T(I),I=1,K)
56 IF(MOT.NE.0)GOTO19
57 DO 18 I=1,K
58 T(I)=T(I)**F(I)
59 WRITE(6,20)M
60 FORMAT(1H0,20X,23HTREND ANALYSIS FOR THE ,12,11HST DATA SET/21X,16
61 1HSUMC * TOTALS),5X,15HSUMS OF SQUARES//)
62 DO 22 I=1,MORD
63 AS(I)=0.0
64 CL(I)=0.0
65 DO 21 J=1,K
66 CL(I)=CL(I)+A(1+I,J)**T(J)
67 AS(I)=AS(I)+F(J)**T(J)*A(1+I,J)**2
68 C(I)=CL(I)*CL(I)/AS(I)
69 WRITE L AND SUMS OF SQUARES
70 WRITE(6,23)1,CL(I),C(I)
71 FORMAT(1H ,12,8HTH ORDER,2F20.3)
72 CONTINUE
73 GOTO1
74 END
```

Table 2
Output Example from Gaito (1965)

Normalized Trend Coefficients (Row 1 Linear, Row 2 Quadratic, Etc.)				
1th Order	-.446001	-.288185	-.130370	.185262 .816525
2th Order	.508404	.013240	-.341565	-.630100 .477135
3th Order	-.457724	.446626	.548754	-.524651 .120933
4th Order	.213946	-.652025	.713152	-.142630 .012735
Trend Analysis for the First Data Set				
	Sum (A * Totals)		Sums of Squares	
1th Order	2.566		1.526	
2th Order	-20.757		92.546	
3th Order	12.270		32.139	
4th Order	3.477		2.691	

REFERENCES

- COOLEY, W. W., & LOHNES, P. R. *Multivariate data analysis*. New York: Wiley, 1971.
- EDWARDS, A. L. *Experimental design in psychological research*. (3rd ed.) New York: Holt, Rinehart & Winston, 1968.
- FISHER, R. A., & YATES, F. *Statistical tables for biological, agricultural and medical research*. (6th ed.) New York: Hafner, 1963.
- GAITO, J. Unequal intervals and unequal n in trend analysis. *Psychological Bulletin*, 1965, **63**, 125-127.
- GRANT, D. A. Analysis of variance tests in the analysis and comparison of curves. *Psychological Bulletin*, 1956, **53**, 141-154.

Behavior Research Methods & Instrumentation
1975, Vol. 7 (5), 479

ITR-CLUSTERING: A LISP program to compute verbal recall intertrial repetitions

PAUL L. DERBY

State University of New York, Geneseo, New York 14454

Shuell (1969) reviewed a number of techniques used in quantifying verbal recall data using various clustering paradigms. One of these paradigms, subjective organization, assumes that subjects recall words in some repetitive fashion from trial to trial. As a measure of subjective organization, Bousfield and Bousfield (1966) suggested tabulating the frequency of occurrence of any given pair of words in one trial list that appear adjacent in the successive trial list as an intertrial repetition (ITR). The sum of ITRs for a given subject compared with the number of ITRs one might expect by chance provides a measure of the degree of word pair association a subject uses in memorizing word lists.

ITR-CLUSTERING was developed to assist verbal learning researchers in searching recall lists for occurrences of ITRs. For N recall lists the program recursively scans the second through Nth list for occurrence of any pair of words in the first through N - 1 lists. The presence of any adjacent pairs in two successive lists is scored as an ITR. The order of the words in any given pair is ignored in scoring ITRs.

Input. ITR-CLUSTERING reads recall lists in free field format. Words comprising each list are punched across cards with each word delimited by one or more blanks. A subject's recall trial is enclosed in parentheses. Recall lists for each subject may be submitted together as one deck for successive processing by the program.

Output. Data returned for each subject include a subject number, the number of words in each of the recall lists by trial, the number of common words in each successive pair of recall lists by pairs of trials, and the ITR count for each successive pair of trial recall lists.

Computer and Programming Language. ITR-CLUSTERING is written in LISP. All of the functions except the input/output

functions are written in "standard" LISP as described by Weissman (1967). The input/output function is coded according to the specifications of the University of Texas LISP (UT-LISP) interpreter/compiler, Version 3, Modification Level 7, which is implemented for the CDC 6000/Cyber 70, 170 series of computers. Under UT-LISP, ITR-CLUSTERING required 22 sec and 120,000 words of memory to compile. Execution time for 20 subjects and four 9 to 24 word recall lists required 15 sec and 70,000 words of memory. This same data required 94 sec and 100,000 words of memory to process under the interpreter version of UT-LISP. The above times and memory requirements were determined on a CDC 6400.

Restrictions. The number of words per recall list and the number of trials per subject that ITR-CLUSTERING can process is determined solely by the memory size of the available computer. The input/output function of the program is written to process four recall trials per subject. This part of the program may be easily modified by adding or deleting input/output directives to accommodate the desired number of trials per subject. Since the actual list scanning is accomplished recursively, there are no internal program limitations for the length of any of the recall trial lists. However, the program does not check to insure that all elements of any given recall list are unique. Also, each comparison made by the program is exact; various grammatical and spelling forms of the same word are treated as unique words.

Availability. A listing of the program with test data and sample output is available from the author at no cost.

REFERENCES

- BOUSFIELD, A. K., & BOUSFIELD, W. A. Measurement of clustering and of sequential constancies in repeated free recall. *Psychological Reports*, 1966, **19**, 135-142.
- SHUELL, T. J. Clustering and organization in free recall. *Psychological Bulletin*, 1969, **72**, 353-374.
- WEISSMAN, C. *LISP 1.5 primer*. Belmont: Dickenson, 1967.