DR. JOHN MANIOTES
COMPUTER TECHNOLOGY DEPT.
PURDUE UNIVERSITY
CALUMET CAMPUS
HAMMOND, IN 46323

DISCLAIMER

Although each program has been tested by its contributor, no warranty, express or implied, is made by the contributor or COMMON, as to the accuracy and functioning of the program and related program material, nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the contributor or COMMON, in connection therewith.

COMMON USERS GROUP PROGRAM REVIEW AND EVALUATION
(fill out in typewriter, ink or pencil)

Program No._____ Date_____

Program Name:_____

1. Does the abstract adequately describe what the program is and what it does? Yes____ No____
   Comment_____

2. Does the program do what the abstract says? Yes____ No____
   Comment_____

3. Is the description clear, understandable, and adequate? Yes____ No____
   Comment_____

4. Are the Operating Instructions understandable and in sufficient detail? Yes____ No____
   Comment_____
   Are the Sense Switch options adequately described (if applicable)? Yes____ No____
   Are the mnemonic labels identified or sufficiently understandable? Yes____ No____
   Comment_____

5. Does the source program compile satisfactorily (if applicable)? Yes____ No____
   Comment_____

6. Does the object program run satisfactorily? Yes____ No____
   Comment_____

7. Number of test cases run_____. Are any restrictions as to data, size, range, etc. covered adequately in description? Yes____ No____
   Comment_____

8. Does the Program meet the minimal standards of COMMON? Yes____ No____
   Comment_____

9. Were all necessary parts of the program received? Yes____ No____
   Comment_____

10. Please list on the back any suggestions to improve the usefulness of the program. These will be passed onto the author for his consideration.

Please return to:

Mr. Richard L. Pratt
Data Corporation
7500 Old Xenia Pike
Dayton, Ohio 45432

Your Name _____
Company _____
_____
Address _____
_____
Users Group Code _____

THIS REVIEW FORM IS PART OF THE COMMON ORGANIZATION'S PROGRAM REVIEW AND EVALUATION PROCEDURE. NONMEMBERS ARE CORDIALLY INVITED TO PARTICIPATE IN THIS EVALUATION.

a 11/1/65

LINEAR CORRELATION COEFFICIENT

ROBERT J. ROBINSON

MARQUETTE UNIVERSITY

MILWAUKEE, WISCONSIN

December 7, 1961

Modifications or revisions to this program, as they occur, will be announced in the appropriate Catalog of Programs for the IBM Data Processing Systems. If such announcement indicates a change to the program decks or tapes, a complete new program, if needed, should be requested from the Program Distribution Center.

MARQUETTE UNIVERSITY
COMPUTING CENTER
SUBROUTINE MANUAL
FOR THE IBM 1620

Program:	LINEAR CORRELATION COEFFICIENT
Date:	October 18, 1961
Programmer:	Mr. Robert J. Robinson
Description:	This program will compute the linear correlation coefficients between up to 50 variables. Input and output are via cards. Provision is made for computing the coefficients for partial data sets, merging following sets into the existing sums. Output includes the sum, sums of square, sum of cross products, the standard deviations, and the standard error of the coefficients.
Coding Language:	Fortran
Input/Output Format:	Notation:

N = number of observations.

J = number of variables in the data set.
(ie, the number of columns)

$X_1, X_2, \ldots, X_j$ = the J variables.

X_I = any particular variable of the J variables.

r_{IK} = correlation coefficient between variables X_I and X_K

σ_{X_I} = standard deviation of variables X_I

$\sigma_{r_{IK}}$ = standard error of the IK^{th} correlation coefficient.

Input Format:

Data cards must be punched row-wise. That is, if the data were:

X_1	X_2	X_3	X_4	X_5	X_6
66.3	5.886	43.95	-1058.2	47.18	-.0095
65.8	4.876	42.85	-1068.5	58.17	-.0105

The first row would be punched on a card (or cards-not beyond column 72, and a value must not be split into two parts) as:

66.3 5.886 43.95 -1058.2 47.18 -.0095

Since columns 73-80 are not used, a card or problem number may be punched there.

First card of the data set must contain the value of J in fixed point form (ie, no decimal point). Normally this will be the only value on the card, but one may begin punching the elements immediately after J (leaving a space). Hence, the first card of the deck for the above example might be:

6

The last data card of the deck must not contain data for more than one observation of the variables. That is, do not "double up" two data sets on the last card. (This may be done for all but the last card). This restriction is required because the Fortran statement IF (sense switch 9) is used to determine when the last card is fed.

Several Sets of Data:

If several data sets are to be run, they may not be placed one behind the other in the read hopper. They must be kept separate, and inserted (other than the first deck) only when the computer spaces the typewriter paper up three times and halts. (see operating instructions).

Subdividing a Single Data Set:

Sometimes it is desirable to be able to divide the data into sub-decks and compute the correlations for each sub-deck plus the one before it. That is, suppose that one has 1000 observations of several variables, and wishes to get the correlations, first on the first 200, then on the first 400, then on the first 600, etc. to the first 1000, or all data.

To do this, the complete data set should be punched according to the input format specifications, and then sub-divided into separate decks, the first containing the first 200 observations, the next the following 200, etc. (Note: the last card of each sub-deck must not contain data for more than one observation of the variables; same requirement as is placed on the last card of any deck). The card containing J is to be used only in front of the first deck. Next, load the program and the first data set, as described in the operating procedures. When the typewriter spaces 3 times and the computer halts, turn sense switch 2 on, insert the next data set in the read hopper, press reader start, and start on the console. Press reader start when the last card is partially fed. By turning switch 2 on, when start is depressed the computer is instructed to branch to the part of the program which reads in cards and forms necessary sums, skipping the portion which initally sets the sums to zero. This procedure may be repeated as often as desired, but the J value may not be the first value of any deck other than the first. (See operating instructions for reverting to normal operation by turning switch 2 off).

Output Format:

For every variable there will be two lines (two cards) output in the first group of output.

Line 1: I (the number of the variable) $\bar{X}_I$ σ_{X_I} ΣX_I

Line 2: ΣX_I^2

there will be J such pairs of lines, followed by one line containing the number of observations:

Line 2J+1: N

The next group of output again consists of two line pairs. They are:

Line 1: I K r_{IK} $\Sigma X_I X_K$

Line 2: $\sigma_{r_{IK}}$

Restrictions:

Not more than 50 variables nor more than 9,999 observations. All elements other than J must be punched with a decimal point.

Accuracy and Speed:

Accuracy variable with the number of observations.

Speed: Loading time per data set, $LT = .28 + .0975J + .0191hJ^2$

Compute and punch time, $CT = 21.0 - 2.7h5J + 1.118J^2$

Total running time, $TR = N.LT + CT$

Examples:

J	LT	CT	N	TR
5	1.2 sec.	36 sec.	200	h.6 min.
10	3.33	10h	100	7.3
18	7.7			
20	10.32	hlh	50	15.5
h0	3h.8	1700	25	h2.8

Operating Instructions:

1. Place program and first data set(including J) in read hopper, and blank cards in punch hopper. Press punch start.
2. Press instant stop and reset keys on console.
3. Switches 1, 2, 3, off (2 may be set at this point. See below). O'flow switch to program, all others to stop.
4. Press the load key on the card reader.
5. When ''load data'' types out, press start on the console.
6. When the last card is partially fed press reader start.
7. Computer will punch answers, and space typewriter 3 times and stop. Now if additional data is to be run do either the steps marked (a) or the steps marked (b):

If the next data set is entirely independent of the one just run (ie, has its own J value):

8a. If switch 2 is on, turn it off.

9a. Press start on the console.

10a. Insert the next data deck and press reader start. Go back to step 6.

If the next data set is a sub-deck of the one just read (ie, does not have its own J value):

8a. If switch 2 is off, turn it on.

9a. Press start on the console.

10a. Insert only the next sub-deck and press reader start. Go back to step 6.

Equipment Required:

May be used on any IBM 1620 with card I/O, but if memory size is 20K, program must be re-assembled to reduce the size of the dimensioned variable arrays.

Remarks:

The formulas used by the routine are:

$$r_{IK} = \frac{N\Sigma X_I X_K - (\Sigma X_I)(\Sigma X_K)}{\sqrt{N\Sigma X_I^2 - (\Sigma X_I)^2}\ \sqrt{N\Sigma X_K^2 - (\Sigma X_K)^2}}$$

If $N > 30$,

$$\sigma_{X_I} = \frac{1}{N}\sqrt{N\Sigma X_I^2 - (\Sigma X_I)^2}$$

If $N \leq 30$,

$$\sigma_{X_I} = \frac{1}{N(N-1)}\sqrt{N\Sigma X_I^2 - (\Sigma X_I)^2}$$

Hash Total:

3h05397808751731068h

(Using card hash total program developed at Marquette University).

```
       ONE POSSIBLE INPUT FORMAT (J=3) IS***
(J)
3
(X1) (X2) (X3)
2.0 6.0 3.0
3.0 4.0 8.0
1.0 5.0 9.0
4.0 8.0 4.0
5.0 1.0 7.0
       THE CORRESPONDING OUTPUT FORM IS(WITH LABELS HANDWRITTEN)***
   (I)            (X̄1)            (σX1)           (ΣX1)
    1       3.0000000E+00   1.5811388E+00   1.5000000E+01
  (ΣX1²)
 5.5000000E+01
                   (X̄2)            (σX2)           (ΣX2)
    2       4.8000000E+00   2.5884357E+00   2.4000000E+01
  (ΣX2²)
 1.4200000E+02
                   (X̄3)
    3       6.2000000E+00   2.5884357E+00   3.1000000E+01

 2.1900000E+02
  (N)
    5
   (I)          (K)            (r12)           (ΣX1X2)
    1            2       -3.6650836E-01   6.6000000E+01
  (σr12)
 3.8714016E-01
                                (r13)           (ΣX1X3)
    1            3       -1.8325418E-01   9.0000000E+01
  (σr13)
 4.3219528E-01
                                (r23)           (ΣX2X3)
    2            3       -5.5223888E-01   1.3400000E+02
  (σr23)
 3.1082790E-01
```



```
C     CORRELATION ROUTINE
C     J= NO.OF VARIABLES OBSERVED
DIMENSION SUM(50),X(50),SUMXY(50,50)
      IF(SENSE SWITCH 9)18,18
   18 N=0
      READ,J
C  CLEAR ALL ARRAYS TO 0.0
      DO 1 I=1,J
      SUM(I)=0.0
      DO 1 K=1,J
    1 SUMXY(I,K)=0.0
C  READ A CARD AND FORM SUMS,AND COUNT NO.OF OBSERVATIONS
    3 DO 2 I=1,J
      READ, X(I)
    2 SUM(I)= SUM(I) + X(I)
      DO 4 I=1,J
      DO 4 K=I,J
    4 SUMXY(I,K)= SUMXY(I,K)+ X(I)* X(K)
      N= N+1
      IF (SENSE SWITCH 9) 5,3
C  ALL CARDS READ AND ALL PRODUCTS FORMED
C  NOW COMPUTE CORRELATION COEFFICIENTS
    5 FN =N
      IF(N-30) 25,25,26
   25 FN1= SQR(FN*(FN-1.0))
      GO TO 27
   26 FN1= FN
   27 JM1 = J-1
      DO 9 I= 1,JM1
      AMEAN = SUM(I)/FN
      DENOM = SQR ( FN * SUMXY(I,I) -SUM(I)**2)
      STDEV = DENOM/ FN1
      IF (SENSE SWITCH 1) 12,13
   12 PRINT,I,AMEAN,STDEV,SUM(I), SUMXY(I,I)
   13 PUNCH,I,AMEAN,STDEV,SUM(I), SUMXY(I,I)
      KK= I+1
      DO 9 K=KK,J
      DENM2 = SQR(FN* SUMXY(K,K)-SUM(K)**2)
    9 SUMXY(K,I)= (FN*SUMXY(I,K)- SUM(I)*SUM(K))/(DENOM * DENM2)
C  VALUES OF MEAN,STANDARD DEVIATION,SUM,AND SUM OF SQUARES OUTPUT FOR
C     ALL BUT THE LAST VARIABLE.DO THAT NEXT.
      AMEAN= SUM(J)/FN
      STDEV= DENM2/FN1
      IF(SENSE SWITCH 1) 10,11
   10 PRINT,I,AMEAN,STDEV,SUM(J),SUMXY(J,J)
      PRINT
      PRINT
      PRINT
      PRINT,N
   11 PUNCH,I,AMEAN,STDEV,SUM(J),SUMXY(J,J)
      PUNCH,N
C  OUTPUT THE CORRELATION COEFFICIENT AND SUM OF CROSS PRODUCTS
      DO15I=1,JM1
      KK= I+1
```

```
      DO 15 K= KK,J
      SIGMAR = (1.0-SUMXY(K,I)**2)/SQR(FN)
      IF(SENSE SWITCH 1) 16,17
   16 PRINT, I,K, SUMXY(K,I), SUMXY(I,K) ,SIGMAR
   17 PUNCH, I,K, SUMXY(K,I), SUMXY(I,K) ,SIGMAR
   15 CONTINUE
      PRINT
      PRINT
      PRINT
      PAUSE
      IF(SENSE SWITCH 2) 3,18
      END
```

END OF LISTING