Computer Methods and Programs in Biomedicine 24 (1987) 41–45
Elsevier

CPB 00809

Computer program for calculation of kinetic and pharmacologic parameters using a 'direct linear plot' derived algorithm

Zbigniew W. Kaminski* and Edward F. Domino

Department of Pharmacology, The University of Michigan, Ann Arbor, MI, U.S.A.

Data processing using nonparametric statistics was performed using a 'direct linear plot' algorithm of ECB. ECB (Eisenthal and Cornish-Bowden) is a computer program designed to calculate the K_m and V parameters in enzyme kinetics. It is also suitable for estimation of EC_{50} (ligand or drug concentration for 50% effect) in pharmacological studies.

BASIC; Nonparametric statistics; Enzyme kinetics; EC_{50}; Direct linear plot

1. Introduction

There are many biological systems which express a hyperbolic relationship between a measured response and a controlled variable. Such a relationship has been shown by Michaelis and Menten [10] for steady-state enzyme kinetics. The Michaelis-Menten equation describes the dependence between the velocity of an enzymatic reaction and concentration of the substrate as follows:

$$v = V \cdot S/(K_m + S)$$

where
v is the actual velocity at a substrate concentration equal to S,
K_m is the Michaelis-Menten constant and
V is maximum velocity.

The same equation may be applied to many non-enzymatic systems such as drug–receptor interactions [11]. Moreover, a number of pharmacological phenomena can be described in a similar way, for example, a drug effect at a given concentration or dose (ED_{50} calculations).

In most pharmacological as well as enzymological experiments, the form of the distribution of errors cannot easily be determined [14]. The assumption of a normal distribution of errors in such experiments frequently has not been proven [14]. It is well known that applications of least squares without proper knowledge of the error distribution of the data make the results invalid [15]. Therefore, a distribution free method of estimation is more appropriate.

In the computer program presented here we applied the 'direct linear plot' algorithm implemented by Eisenthal and Cornish-Bowden in 1976 [6]. For an extensive discussion of their algorithm see [1,3,4].

* *Current address:* Adjunct Professor, Department of Comparative Biochemistry, Institute of Biochemistry and Biophysics, Polish Academy of Sciences, 02-532 Warsaw, Rakowiecka 36, Poland.
Correspondence: E.F. Domino, Department of Pharmacology, M6414 Medical Science Bldg. I, University of Michigan, Ann Arbor, MI 48109-0010, U.S.A.

2. Methods

2.1. Algorithm description

The algorithm presented here is a modified version of one described by Porter and Trager [12]. Few changes were introduced according to the subsequent papers of Cornish-Bowden and Eisenthal [2] and Cornish-Bowden et al. [4]. The whole calculation procedure had been derived from the 'direct linear plot' of S/V against $1/V$ [2].

The data pairs (e.g., S_i/V_i, $1/V_i$) should be plotted not as points but rather as straight lines passing through (S_i/V_i, 0) and (0, $1/V_i$). Under such conditions, line intersections and coordinates of the intersection point, e.g., i-line with S_i/V_i and $1/V_i$ intersects with the j-line with S_j/V_j and $1/V_j$ to form the set of $K_{m(i,j)}/V_{(i,j)}$ and $1/V_{(i,j)}$ [2,4]. For n data pairs the total number of intersections (N) is described by the following equation:

$$N = [n(n-1)/2] - p_2 - p_3 - \ldots - [r(r-1)]\, p_r$$

where p_2 and p_3 are the number of duplications and triplications, and p_r the number of replicate observations of size r [12].

The $K_{m(i,j)}/V_{(i,j)}$ and $1/V_{(i,j)}$ values are calculated using the following equations:

$$K_{m(i,j)}/V_{(i,j)} = \left[(V_j - V_i)(S_j/V_j - S_i/V_i)\right] / \left[(S_j - S_i)(V_i/S_i - V_j/S_j)\right]$$

$$1/V_{(i,j)} = (S_j/V_j - S_i/V_i)/(S_j - S_i)$$

In the next step the sets of values mentioned above are sorted in ascending order. The best estimate of $X_{(ij)}$ (e.g., $K_{m(i,j)}/V_{(i,j)}$) is the median value taken from the $X_{(ij)}$ set [12]. Therefore, the best estimates of K_m and V are calculated from median $K_{m(i,j)}/V_{(i,j)}$ and $1/V_{(i,j)}$ values. The confidence limits (from lower to upper bound elements) are obtained from Kendall's [7] S distribution. According to Sen [13], lower and upper bound elements $X_{(ij)}$ have the ranks $(N - S^*)/2$ and $(N + S^*)/2 + 1$, respectively. The calculation of normal approximation of S* was published by Sen [13] and by Porter and Trager [12]. Lower and upper limits for K_m cannot be obtained from the lower and upper values of K_m/V and $1/V$ mentioned above. Therefore, a set of $K_{m(ij)}$ values is calculated for this purpose [4] from the equation:

$$K_{m(ij)} = (V_j - V_i)/(V_i/S_i - V_j/S_j)$$

2.2. Structure of the ECB program

The ECB program contains three main parts: data editor, calculation subroutine and file-creating routine. The data editor has been designed to allow interactive data editing with opportunities for changing, deleting, and inserting data pairs. After the data set is accepted by the user, S and V pairs are sorted in ascending order. This part of the program searches for replications, counts their size, and evaluates the confidence limits. The calculation subroutine computes $K_{m(i,j)}/V_{(i,j)}$, $1/V_{(i,j)}$ and $K_{m(i,j)}$ values. The next step consists of sorting the sets of the above elements by the QUICKSORT routine [5,8]. QUICKSORT seems to be the best possible algorithm for this purpose. Nevertheless, execution of this part is the most time consuming. The last part of the program contains a file-creating subroutine which allows permanent storage of the data files on a floppy disk.

```
NUMBER OF DATA POINTS  7
NUMBER OF INTERSECTIONS  21
KENDALL CONFIDENCE INTERVAL FROM  4  TO  18

                LOWER BOUND     MEDIAN          UPPER BOUND

Km              2.3017D+01      2.6726D+01      3.0697D+01
V               2.4272D+01      2.6290D+01      2.7300D+01
Km/V            9.5775D-01      1.0166D+00      1.1191D+00
V/Km            8.9357D-01      9.8369D-01      1.0441D+00
1/Km            3.2576D-02      3.7417D-02      4.3445D-02
1/V             3.6630D-02      3.8037D-02      4.1199D-02


#  [S]          v obs           v calc          ERR             REL ERR [%]

1  7.9000D+00   6.0000D+00      5.9982D+00      1.8492D-03       .03
2  1.0000D+01   7.0000D+00      7.1584D+00     -1.5845D-01     -2.27
3  1.3600D+01   9.0000D+00      8.8664D+00      1.3362D-01      1.48
4  2.6200D+01   1.2500D+01      1.3014D+01     -5.1441D-01     -4.12
5  6.7600D+01   1.9000D+01      1.8841D+01      1.5887D-01       .83
6  9.9500D+01   2.0500D+01      2.0724D+01     -2.2363D-01     -1.1
7  1.3700D+02   2.2000D+01      2.1999D+01      1.4339D-03      0
```

Fig. 1. Example hard copy of the results.

2.3. Example program run

Fig. 1 shows a printout of the results of K_m and V calculations for the data taken from Kuhn [9]. The table contains: the table file name, date and time of the file creation, the total number of S and V pairs, the number of intersections (N), and Kendall's confidence interval expressed as a lower and upper rank for the intersection number. Moreover, K_m, V, K_m/V, V/K_m, $1/K_m$ and $1/V$ values have been shown with their lower and upper confidence limits. The last part of the data table contains a list of all S and V data pairs with V values calculated respectively for 'median' K_m and V, and also absolute and relative errors for observed V values.

3. Hardware and software

The ECB program was written in MICROSOFT interpreted BASIC 2.0 for the Apple McIntosh (512K RAM). The program occupies 11K of memory. The second version in Microsoft interpreted BASIC 2.0 for IMB-PC is also available.

Acknowledgement

Z.W.K. and E.F.D. are supported in part by the Psychopharmacology Research Fund.

References

[1] G.L. Atkins and I.A. Nimo, *in* Kinetic Data Analysis: Design and Analysis of Enzyme and Pharmacokinetic Experiments, ed. L. Endrenyi, pp. 121–135 (Plenum Press, New York, 1980).

[2] A. Cornish-Bowden and R. Eisenthal, Biochim. Biophys. Acta, 523 (1978) 268–272.

[3] A. Cornish-Bowden, *in* Kinetic Data Analysis: Design and Analysis of Enzyme and Pharmacokinetic Experiments, ed. L. Endrenyi, pp. 105–119 (Plenum Press, New York, 1980).

[4] A. Cornish-Bowden, W.R. Porter and W.F. Trager, J. Theor. Biol., 74 (1978) 163–175.

[5] R. Dayton, (1985) MacIntosh Microsoft Basic (Reston Publishing, Reston VA, 1985).

[6] R. Eisenthal and A. Cornish-Bowden, Biochem. J. 139 (1974) 715–720.

[7] M.G. Kendall, Rank Correlation Methods, 4th edn. (Charles Griffin London, 1970).

[8] D.E. Knuth, The Art of Computer Programming (Addison-Wesley, Menlo Park CA, 1973).

[9] R. Kuhn, Hoppe-Seyler's Z. Physiol. Chem., 125 (1923) 28–92.

[10] L. Michaelis and M.L. Menten, Biochem. Z., 49 (1913) 333–369.

[11] J. Norman, *in* Pharmacokinetics of Anaesthesia, eds. C. Prys-Roberts and C.C. Hug Jr., pp. 25–37 (Blackwell Scientific Publications, Oxford, 1984).

[12] W.R. Porter and W.F. Trager, Biochem. J., 161 (1977) 293–302.

[13] P.K. Sen, J. Am. Stat. Assoc., 63 (1968) 1379–1389.

[14] A.C. Storer, M.G. Carlison and A. Cornish-Bowden, Biochem. J., 151 (1975) 361–367.

[15] D.G. Watts, *in* Kinetic Data Analysis: Design and Analysis of Enzyme and Pharmacokinetic Experiments, ed. L. Endrenyi, pp. 1–24 (Plenum Press, New York, 1980).

Program listing

```
10 ' ECB (from Eisenthal & Cornish-Bowden)
20 ' PROGRAM FOR Km AND V CALCULATION USING KENDALL RANK STATISTICS
30 ' AND EISENTHALL, CORNISH-BOWDEN PROCEDURE
40 ' FOR DETAILS see: Porter W.R., Trager W.F., (1977) Biochem.J.
[161],293-302
50 ' Cornish-Bowden A., (1978) Biochim.Biophys.Acta [523],268-272
60 ' Cornish-Bowden A., Porter W.R., Trager W.F., (1978) J.theor.Biol.
[74],163-175
70 DIM A(1,30),B(2,435),C(8),D(2,30),R%(4),L(435),M(435)
80 CLS
90 PRINT
100 PRINT SPC(20);"Select Command You Wish"
110 PRINT SPC(9);"Data = "DA" or "da" ; Read the File from disk = "RE" or "re""
120 INPUT"                    ***** COMMAND *****
";A$:A$=UCASE$(A$)
130 IF A$="DA" THEN 180
140 IF A$="RE" THEN 1920
150 IF LEN(A$)=0 THEN 80
160 CLS:PRINT:PRINT"              ******* ILLEGAL COMMAND !!! ******* 
*"
170 GOTO 90
180 CLS:GOTO 200
190 A1=1
200 PRINT"Input number of pairs [S],[v]  ";:CALL TEXTFACE(A1):PRINT"[not less
than 5 and not more than 30]";:CALL TEXTFACE(0):INPUT N%:IF N%<5 OR N%>30
THEN 190
210 FOR I%=0 TO N%-1
220 PRINT"input [S] and [v] for pair #";I%+1;:IF I%=0 THEN PRINT" ; for example
3.125,8.116";
230 IF I%=N%-1 THEN PRINT" ***** last data pair *****";
240 INPUT A(0,I%),A(1,I%):IF A(0,I%)<=0 OR A(1,I%)<=0 THEN GOSUB 2110
250 NEXT I%
260 CLS
270 'Preliminary sorting of the data set
280 FOR J%=1 TO N%-1
290 FOR I%=0 TO N%-1
300 IF A(0,I%)<A(0,J%) THEN 320
310 SWAP A(0,J%),A(0,I%):SWAP A(1,J%),A(1,I%)
320 NEXT I%,J%
330 FOR J%=1 TO N%-1
340 FOR I%=0 TO N%-1
350 IF A(0,I%)<>A(0,J%) THEN 380
360 IF A(1,I%)<A(1,J%) THEN 380
370 SWAP A(1,I%),A(1,J%)
380 NEXT I%,J%
390 PRINT TAB(2);"#";:PRINT TAB(21);"[S]";:PRINT TAB(42);"[v]"
400 FOR I%=0 TO N%-1
410 PRINT I%+1;:PRINT TAB(20);A(0,I%);:PRINT TAB(40);A(1,I%)
420 NEXT I%
430 'Searching of replicative observation and determination of their size
440 K1=0:FOR I%=0 TO 4:R%(I%)=0:NEXT I%
450 FOR I%=0 TO N%-2
```

```
460 IF A(0,I%)<>A(0,I%+1) THEN 490
470 K1=K1+1:IF K1>4 THEN 2120
480 GOTO 510
490  R%(K1)=R%(K1)+1
500 K1=0
510 NEXT I%
520  R%(K1)=R%(K1)+1
530 DATA editor
540 PRINT:PRINT SPC(13);:INPUT"***  CONTINUE  *** : YES="Y" or "y"
";X$:X$=UCASE$(LEFT$(X$,1)):IF X$<>"Y" THEN 100
550 PRINT:INPUT"Do you want to CHANGE the DATA ; YES = "Y" or "y"
";B$:B$=UCASE$(LEFT$(B$,1))
560 IF B$<>"Y" THEN 720
570 PRINT"In Which MANNER : change="C" or "c"; insert="I" or "i"; delete="D" or
"d" "
580 PRINT"                    *** ESCAPE *** = "E" or "e" ?"
590 C$=INKEY$:IF C$="" THEN 590
600 C$=UCASE$(C$):IF C$<>"I" AND C$<>"D" AND C$<>"C" AND C$<>"E" THEN 570
610 IF C$="E" THEN 540
620 IF C$="I" THEN 690
630 IF N%-1<5 AND C$="D" THEN PRINT"TOO FEW POINTS can not DELETE":GOTO
550
640 PRINT"Which pair ; input #";:INPUT J%:IF J%<1 OR J%>N% THEN 640
650 IF C$="C" THEN 700
660  SWAP  A(0,J%-1),A(0,N%-1):SWAP  A(1,J%-1),A(1,N%-1)
670 N%=N%-1
680 GOTO 260
690  N%=N%+1:J%=N%
700 PRINT"Type a NEW VALUES of [S] and [v] ; for example 2.081,3.145
";:INPUT"", A(0,J%-1),A(1,J%-1):IF A(0,J%-1)=<0 OR A(1,J%-1)=<0 THEN
GOSUB 2110:GOTO 700
710 GOTO 260
720 CLS
730 LOCATE 5,2
740 PRINT" I am working on calculation of Km(i,j), 1/V(i,j) and Km(i,j)/V(i,j)
sets "
750  I1%=1:A2%=-1
760 FOR J%=0 TO N%-2
770 FOR I%=I1% TO N%-1
780 IF A(0,J%)=A(0,I%) THEN 880
790 A2%=A2%+1:'Counting of the number of intersections 'N'
800  AA1=A(1,J%)-A(1,I%):'  v(j)-v(i)
810  AA2=A(0,J%)-A(0,I%):'  S(j)-S(i)
820  AA3=A(1,I%)/A(0,I%):'  v(i)/S(i)
830  AA4=A(1,J%)/A(0,J%):'  v(j)/S(j)
840 IF AA1=0 THEN AA1=1E+38
850 B(0,A2%)=AA1/(AA3-AA4): ' Km(i,j)
860 B(1,A2%)=(1/AA4-1/AA3)/AA2: ' 1/V(i,j)
870  B(2,A2%)=B(0,A2%)*B(1,A2%): '  Km(i,j)/V(i,j)
880 NEXT I%
890 I1%=I1%+1
900 NEXT J%
910 IF A2%<9 THEN 2130
920 CLS:LOCATE 7,2
930 PRINT" I am just sorting Km(i,j), 1/V(i,j) and Km(i,j)/V(i,j) sets ":IF
N%<10 THEN 990
940 CALL TEXTFACE(1):PRINT:PRINT"    PLEASE BE PATIENT YOUR CALCULATION IS
BEING PROCESSED":CALL TEXTFACE(0)
950 ' QUICKSORT 2 soting algorithm
960 ' It is 21 times faster than BUBBLESORT AND 3 times faster than SHELLSORT (at
500 data points)]
970 ' See Knuth D.E., in "The Art of Computer Programming; vol.3/Sorting and
Searching" Addison Publishing Company 1973 ; ISBN0-201-03803-X
980 ' See also Dayton R., in "MACINTOSH MICROSOFT BASIC" Reston Publishing
Company, Inc., Reston, Virginia, U.S.A. 1985 ; ISBN 0-8359-4158-2
990 FOR I1%=0 TO 2
1000  I%=0:L(0)=0:M(0)=A2%
1010  WHILE  I%>=0:L%=L(I%):J%=M(I%):I%=I%-1
1020 WHILE L%<J%
1030   M%=L%:K%=J%:X=B(I1%,INT((L%+J%)/2))
1040 WHILE NOT B(I1%,M%)>=X:M%=M%+1:WEND
1050 WHILE NOT X>=B(I1%,K%):K%=K%-1:WEND
1060 IF M%>K% THEN 1080
1070  SWAP  B(I1%,M%),B(I1%,K%):M%=M%+1:K%=K%-1
1080 IF M%<=K% THEN 1040
1090 IF M%>=J% THEN 1110
1100  I%=I%+1:L(I%)=M%:M(I%)=J%
1110 J%=K%
1120 WEND:WEND
1130 NEXT I1%
1140 CLS:LOCATE 9,2
1150 PRINT" I am very close to FINISH"
1160 P=0
1170 FOR J%=1 TO 4
1180  P=P+R%(J%)*J%*(2*(J%+1)+5)
1190 NEXT J%
1200  P=SQR((N%*(N%-1)*(2*N%+5)-P)/18)
1210 SK=1.945*P:'Kendall 'S'
1220 UB=.5*(A2%+1+SK)+1:'Upper Bound'
1230 LB=.5*(A2%+1-SK):GOSUB 2140:'Lower Bound'
1240 GOSUB 2190
1250 GOSUB 2270
1260 BEEP:CLS
1270 PRINT:PRINT TAB(10 );F1$;"  ";DA$;"  ";TI$:PRINT
1280 PRINT"NUMBER OF DATA POINTS ";N%
1290 PRINT"NUMBER OF INTERSECTIONS ";A2%+1
1300 PRINT"KENDALL CONFIDENCE INTERVAL FROM ";LB" TO ";UB
1310 PRINT
1320 PRINT TAB(12);"LOWER BOUND";TAB(30);"MEDIAN";TAB(42);"UPPER
BOUND"
1330 PRINT:Z$="    ##.####^^^^"
1340 PRINT"Km";:PRINT TAB(10):PRINT USING Z$;C(0);C(1);C(2)
1350 PRINT"V";:PRINT TAB(10):PRINT USING Z$;C(3);C(4);C(5)
1360 PRINT"Km/V";:PRINT TAB(10):PRINT USING Z$;C(6),C(7);C(8)
1370 PRINT"V/Km";:PRINT TAB(10):PRINT USING Z$;1/C(8);1/C(7);1/C(6)
1380 PRINT"1/Km";:PRINT TAB(10):PRINT USING Z$;1/C(2);1/1/C(1);1/C(0)
1390 PRINT"1/V";:PRINT TAB(10):PRINT USING Z$;1/C(5);1/C(4);1/C(3)
1400 D1%=0:FOR I%=0 TO 8:IF C(I%)=-1 THEN D1%=-1:NEXT I%
1410 IF D1%=0 THEN 1430
1420 PRINT:PRINT"     * * *  NEGATIVE SHOULD BE INTERPRETED AS INFINITE  *
* *"
1430 PRINT:PRINT"  Press ANY key to continue"
1440 IF INKEY$="" THEN 1440
1450 FOR I%=0 TO N%-1
1460    D(0,I%)=C(4)*A(0,I%)/(C(1)+A(0,I%)):D(1,I%)=A(1,I%)-D(0,I%):
D(2,I%)=D(1,I%)/A(1,I%)
1470 NEXT I%
1480 CLS:PRINT:PRINT TAB(10);F1$:PRINT
1490 PRINT"   #";TAB(10);"[S]";TAB(20);"v obs";TAB(30);"v
calc";TAB(41);"ERR";TAB(52);"REL ERR [%]"
1500 Z1$=" ##.####^^^^"
1510 FOR I%=0 TO N%-1:V%=2
1520 IF I%>8 THEN V%=1
1530 PRINT TAB(V%);I%+1;:PRINT USING
Z1$;A(0,I%);A(1,I%);D(0,I%);D(1,I%);:PRINT
SPC(4);INT(D(2,I%)*10000)/100
1540 NEXT I%
1550 PRINT:INPUT"Save the DATA ; Yes ="Y" or "y"; NO="N" or "n"
";B$:B$=UCASE$(LEFT$(B$,1))
1560 IF B$<>"Y" AND B$<>"N" THEN 1550 ELSE IF B$="Y" THEN 1810
1570 PRINT:INPUT"Printout the DATA set using line-printer ; Yes ="Y" or "y"
";B$:B$=UCASE$(LEFT$(B$,1))
1580 IF B$<>"Y" THEN 90
1590 LPRINT:LPRINT F1$;"  ";DA$;"  ";TI$:LPRINT
1600 LPRINT"NUMBER OF DATA POINTS ";N%
1610 LPRINT"NUMBER OF INTERSECTIONS ";A2%+1
1620 LPRINT"KENDALL CONFIDENCE INTERVAL FROM ";LB;" TO ";UB
1630 LPRINT
1640 LPRINT TAB(17);"LOWER BOUND";TAB(37);"MEDIAN";TAB(54);"UPPER
BOUND"
1650 LPRINT
1660 LPRINT"Km";:LPRINT TAB(10):LPRINT USING Z$;C(0);C(1);C(2)
1670 LPRINT"V";:LPRINT TAB(10):LPRINT USING Z$;C(3);C(4);C(5)
1680 LPRINT"Km/V";:LPRINT TAB(10):LPRINT USING Z$;C(6);C(7);C(8)
1690 LPRINT"V/Km";:LPRINT TAB(10):LPRINT USING Z$;1/C(8);1/C(7);1/C(6)
1700 LPRINT"1/Km";:LPRINT TAB(10):LPRINT USING Z$;1/C(2);1/C(1);1/C(0)
1710 LPRINT"1/V";:LPRINT TAB(10):LPRINT USING Z$;1/C(5);1/C(4);1/C(3)
1720 IF D1%=0 THEN 1740
1730 LPRINT:LPRINT"          * * *  NEGATIVE SHOULD BE INTERPRETED AS
INFINITE  * * *"
1740 LPRINT:LPRINT
1750 LPRINT"  #  ";TAB(10);"[S]";TAB(22);"v obs";TAB(35);"v
calc";TAB(50);"ERR";TAB(60);"REL ERR [%]":LPRINT
1760 FOR I%=0 TO N%-1:V%=2
1770 IF I%>8 THEN V%=1
1780 LPRINT TAB(V%);I%+1;:LPRINT USING
Z1$;A(0,I%);A(1,I%);D(0,I%);D(1,I%);:LPRINT
SPC(4);INT(D(2,I%)*10000)/100
```

```
1790 NEXT I%:LPRINT CHR$(12)
1800 GOTO 90
1810 GOSUB 2070
1820 DA$=DATE$:TI$=TIME$
1830 OPEN F$ FOR OUTPUT AS #1
1840 WRITE #1, DA$,TI$,N%
1850 FOR J%=0 TO N%-1
1860 WRITE #1, A(0,J%),A(1,J%)
1870 NEXT J%
1880 CLOSE #1
1890 CLS:PRINT
1900 PRINT TAB(5);"FILE * * *  ";F1$;"  * * *  DONE ! ! !  * * *"
1910 GOTO 1570
1920 GOSUB 2070
1930 ON ERROR GOTO 2030
1940 OPEN F$ FOR INPUT AS #2
1950 ON ERROR GOTO 2030
1960 INPUT #2, DA$,TI$,N%
1970 FOR J%=0 TO N%-1
1980 INPUT #2, A(0,J%),A(1,J%)
1990 NEXT J%
2000 CLOSE #2
2010 CLS:PRINT DA$,TI$,F1$:PRINT
2020  GOTO 390
2030 IF ERL=1940 THEN 2050
2040 GOTO 2400
2050 CLS:LOCATE 2,6:PRINT"TERRIBLY SORRY ; I WAS NOT ABLE TO FIND ***
";LEFT$(F$,(LEN(F$)-4));" ***"
2060 RESUME 90
2070 INPUT"Type Faile Name ; >3 & <9 Characters
";F$:F$=F$+".ECB":F1$=UCASE$(LEFT$(F$,(LEN(F$)-4)))
2080 IF LEN(F$)<7 OR LEN(F$)>13 THEN PRINT"BAD FILE NAME":GOTO 2070
2090 PRINT"""; F1$;""";:INPUT" * * * Are You Sure * * * ; YES = "Y" or "y"
";B$:B$=UCASE$(LEFT$(B$,1)):IF B$<>"Y" THEN 90
2100 RETURN
2110 PRINT"NOR [S] NEITHER [v] CAN BE ZERO OR LESS":I%=I%-1:RETURN
2120 PRINT"TOO BIG SIZE OF REPLICATION ( >5 )":GOTO 90
2130 CLS:PRINT SPC(7);"DATA  INSUFFICIENT FOR KENDALL STATISTIC
TREATMENT":GOTO 90
2140 IF LB-INT(LB)>.5 THEN LB=INT(LB)+1 ELSE 2170
2150 IF UB-INT(UB)>.5 THEN UB=INT(UB)+1 ELSE 2180
2160 RETURN
2170 LB=INT(LB):GOTO 2150
2180 UB=INT(UB):RETURN
2190 M=A2%+1
2200 IF M/2-INT(M/2)>0 THEN 2240
2210 M1=M/2-1:M2=M1+1:SM=(B(2,M1)+B(2,M2))/2:
SV=(B(1,M1)+B(1,M2))/2
2220 KM=SM/SV:V=1/SV
2230 RETURN
2240 M1=INT(M/2)
2250  KM=B(2,M1)/B(1,M1):V=1/B(1,M1)
2260 RETURN
2270 C(0)=B(0,LB-1)
2280 C(1)=KM
2290 C(2)=B(0,UB-1)
2300 C(3)=1/B(1,UB-1)
2310 C(4)=V
2320 C(5)=1/B(1,LB-1)
2330 C(6)=B(2,LB-1)
2340 C(7)=KM/V
2350 C(8)=B(2,UB-1)
2360 FOR I%=0 TO 8
2370 IF C(I%)=<0 THEN C(I%)=-1
2380 NEXT I%
2390 RETURN
2400 PRINT"ERROR #";ERR;"  IN THE PROGRAM LINE #";ERL
2410 STOP
```