บรรณานุกรม

ภาษาไทย

จรัญ จันทลักขณา. สถิติ-วิธีวิเคราะห์และวางแผนวิจัย. พิมพ์ครั้งที่ 4. กรุงเทพมหานคร : สำนักพิมพ์ไทยวัฒนาพานิช, 2523.

เอกชัย ชัยประเสริฐสิทธิ. สถิติที่ไม่ใช่พารามิเตอร์. กรุงเทพมหานคร : โรงพิมพ์มหาวิทยาลัยธรรมศาสตร์, 2521.

ภาษาอังกฤษ

B.J. Winer. Statistical Principles in Experimental Design. 2nd ed. Tokyo : McGraw-Hill Kogakusha, 1971.

Davies, O.L. Design and Analysis of Industial Experiments. London : Oliver and Boyd, 1956.

F.J. Anscombe, and Joseph B.Kruskal. "Statistical Analysis, Special Problems of Outlier and Transformations of Data." International Encyclopedia of the Social Sciences 15 (1968) : 178-193.

George W. Snedecor. "Queries." Biomet ics 11 (March 1955) : 111-113.

Myles Hollander, and Douglas A. Wolfe. Nonparametric Statistical Methods. New York : John Wiley & Sons, 1973.

Snedecor, G.W. Statistical Methods. 4th ed. Ames : Iowa State College Press, 1946.

S. Siegel. Nonparametric Statistics for the Behavioral Sciences. New York : McGraw-Hill Book Co., 1956.

```
C
C       PROGRAM  'GAUSS'
C
C
C
C          PURPOSE
C
C             COMPUTE A NORMALLY DISTRIBUTED RANDOM NUMBER WITH A
C             GIVEN MEAN AND STANDARD DEVIATION.
C
C
C          DESCRIPTION OF PARAMETERS
C
C             IC      -  IC MUST CONTAIN AN ODD INTEGER NUMBER WITH
C                        NINE OR LESS DIGITS ON THE FIRST ENTRY TO
C                        GAUSS.  THEREAFTER IT WILL CONTAIN A UNIFORMLY
C                        DISTRIBUTED INTEGER RANDOM NUMBER GENERATED
C                        BY THE SUBROUTINE FOR USE ON THE NEXT ENTRY
C                        TO THE SUBROUTINE.  (INPUT)
C             AMEAN   -  THE DESIRED MEAN OF THE NORMAL DISTRIBUTION.
C                        (INPUT = 0.0)
C             SD      -  THE DESIRED STANDARD DEVIATION OF THE NORMAL
C                        DISTRIBUTION.  (INPUT)
C             E       -  THE VALUE OF THE COMPUTED NORMAL RANDOM
C                        VARIABLE.  (THE OUTPUT OF THE EXPERIMENTAL
C                        ERROR WHICH IS A NORMAL DISTRIBUTION WITH
C                        MEAN = 0.0  AND VARIANCE = SD(K)**2 IN THE
C                        ADDITIVE MODEL.)
C             EE      -  THE VALUE OF ANTILOG(E).  (THE OUTPUT OF THE
C                        EXPERIMENTAL ERROR WHICH IS A NORMAL
C                        DISTRIBUTION WITH MEAN = 0.0  AND VARIANCE =
C                        SD(K)**2 IN THE MULTIPLICATIVE MODEL.)
C
C
C          REMARK
C
C             THIS PROGRAM USES  RANDU  WHICH IS MACHINE SPECIFIC.
C
C
C          SUBROUTINE AND FUNCTION SUBPROGRAM REQUIRED
C
C               RANDU
```

```
C              METHOD
C
C                   USE 12 UNIFORM RANDOM NUMBERS TO COMPUTE NORMAL RANDOM
C                   NUMBERS BY CENTRAL LIMIT THEOREM.  THE RESULT IS THEN
C                            ADJUSTED TO MATCH THE GIVEN MEAN AND STANDARD
C                            DEVIATION.  THE UNIFORM RANDOM NUMBERS COMPUTED
C                            WITHIN THE SUBROUTINE ARE FOUND BY THE POWER
C                            RESIDUE METHOD.
C
C
C.......................................................................
C
C
      DIMENSION IC(9,9), E(9,9), EE(9,9)
      DOUBLE PRECISION EE
C
C              INITIALIZATION
C
    5 DO 10 I = 1,9
      DO 10 J = 1,9
   10 IC(I,J) = 0.0
C
C              READ INPUT DATA
C
      READ (1,20) SD
   20 FORMAT (9X,F9.0)
      IF (SD.EQ.0.0) STOP
      READ (1,30) ((IC(I,J),J=1,9),I=1,9)
   30 FORMAT (9I10)
C
      AMEAN    = 0.0
      DO 60 I  = 1,9
      DO 50 J  = 1,9
      A        = 0.0
      DO 40 L  = 1,9
      CALL RANDU (IC,IY,Y)
      IC(I,J)  = IY
   40 A        = A+Y
      E(I,J)   = (A-6.0)*SD+AMEAN
      EEE      = E(I,J)
      EE(I,J)  = 10**EEE
   50 CONTINUE
   60 CONTINUE
C
C              PRINT OUTPUT
C
      WRITE (3,70) SD
   70 FORMAT ('1'//9X,F10.2//)
      DO 90 I = 1,22
      WRITE (3,80) (E(I,J),J=1,9)
   80 FORMAT ((6X,9F14.2)/)
   90 CONTINUE
```

```
      WRITE (3,100)
  100 FORMAT ('1')
      DO 120 I = 1,9
      WRITE (3,110) (EE(I,J),J=1,9)
  110 FORMAT ((2X,5(F26.5)/2X,4(F26.5),26X)/)
  120 CONTINUE
C
      GO TO 5
      END
C
C
C.......................................................................
      SUBROUTINE RANDU (IC,IY,Y)
      DIMENSION IC(22,3)
C
C.......................................................................
C
C
C           PURPOSE
C
C              COMPUTE UNIFORMLY DISTRIBUTED RANDOM REAL NUMBERS
C              BETWEEN 0 AND 1.0 AND RANDOM INTEGERS BETWEEN ZERO AND
C              2**31.  EACH ENTRY USES AS INPUT AN INTEGER RANDOM
C              NUMBER AND PRODUCES A NEW INTEGER AND REAL RANDOM
C              NUMBER.
C
C
C           DESCRIPTION OF PARAMETERS
C
C              IC   -  FOR THE FIRST ENTRY THIS MUST CONTAIN ANY ODD
C                      INTEGER NUMBER WITH NINE OR LESS DIGITS.  AFTER
C                      THE FIRST ENTRY, IC SHOULD BE THE PREVIOUS VALUE
C                      OF IY COMPUTED BY THIS SUBROUTINE.
C              IY   -  A RESULTANT INTEGER RANDOM NUMBER REQUIRED FOR
C                      THE NEXT ENTRY TO THIS SUBROUTINE.  THE RANGE OF
C                      THIS NUMBER IS BETWEEN ZERO AND 2**31.
C              Y    -  THE RESULTANT UNIFORMLY DISTRIBUTED, FLOATING
C                      POINT, RANDOM NUMBER IN THE RANGE 0 TO 1.0 .
C
C           REMARK
C
C              THIS SUBROUTINE IS SPECIFIC TO SYSTEM/360 AND WILL
C              PRODUCE 2**29 TERMS BEFORM REPEATING THE REFERENCE BELOW
C              DISCUSSES SEEDS (65539 HERE), RUN PROGRAMS, AND PROBLEMS
C              CONCEARNING RANDOM DIGITS USING THIS GENERATION SCHEME.
C              MACLAREN AND MARSAGLIA, JACM 12, P. 83-89, DISCUSS
C              CONGRUENTIAL GENERATION METHOD AND TESTS.  THE USE OF
C              TWO GENERATORS OF THE RANDU TYPE, ONE FILLING A TABLE
C              AND ONE PICKING FROM THE TABLE, IS OF BENEFIT IN SOME
C              CASES.  65549 HAS BEEN SUGGESTED AS A SEED WHICH HAS
```

```
C                  BETTER STATISTICAL PROPERTIES FOR HIGH ORDER BITS OF THE
C                  GENERATED DEVIATE. SEEDS SHOULD BE CHOSEN IN ACCORDANCE
C                  WITH THE DISCUSSION GIVEN IN THE REFERENCE BELOW. ALSO,
C                  IT SHOULD BE NOTED THAT IF FLOATING POINT RANDOM NUMBERS
C                  ARE DESIRED, AS ARE AVAILABLE FROM RANDU, THE RANDOM
C                  CHARACTERISTICS OF THE FLOATING  POINT DEVIATES ARE
C                  MODIFIED AND IN FACT THESE DEVIATES HAVE HIGH PROBABILITY
C                  OF HAVING A TRAILING LOW ORDER ZERO BIT IN THEIR
C                  FRACTIONAL PART.
C
C               METHOD
C
C                  POWER RESIDUE METHOD DISCUSSED IN IBM MANUL C20-8011,
C                  RANDOM NUMBER GENERATION AND TESTING.
C
C
C.......................................................................
C
C
      IY = IC(I,J)*65539
      IF (IY) 10,20,20
   10 IY = IY+2147483647+1
   20 Y  = IY
      Y  = Y*.4656613E-9
      RETURN
      END
```

```
C ..............................................................
C
C
C       PROGRAM  'TUKEY'S TEST FOR NON-ADDITIVITY'
C          PURPOSE
C              COMPUTE TUKEY'S TEST FOR NON-ADDITIVITY
C              BEFORE AND AFTER TRANSFORMING BY ***
C                 (1)   THE LOGARITHMIC TRANSFORMATION
C                 (2)   THE ARCSINE TRANSFORMATION
C                 (3)   THE SQUARE ROOT TRANSFORMATION
C          DESCRIPTION OF PARAMETERS
C              N    -   THE INPUT NUMBER OF ROWS (#TREATMENTS)
C              M    -   THE INPUT NUMBER OF COLUMNS (#BLOCKS)
C              U    -   THE INPUT POPULATION MEAN
C              T    -   THE INPUT TREATMENT EFFECT
C              B    -   THE INPUT BLOCK EFFECT
C              E    -   THE INPUT EXPERIMENTAL ERROR
C              X    -   THE OUTPUT MATRIX, N BY M OF THE MULTIPLICATIVE
C                       INPUT DATA FOR TESTING TUKEY'S TEST
C              Y    -   THE OUTPUT TRANSFORMATION MATRIX, N BY N
C             X1    -   X-I-DOT          SUM OF I-TH ROW
C             X2    -   X-DOT-J      -   SUM OF J-TH COLUMN
C         X1MEAN    -   X-BAR-I-DOT  -   MEAN OF I-TH ROW
C         X2MEAN    -   X-BAR-DOT-J  -   MEAN OF J-TH COLUMN
C             Y1    -   Y-I-DOT      -   SUM OF I-TH ROW
C             Y2    -   Y-DOT-J      -   SUM OF J-TH COLUMN
C         Y1MEAN    -   Y-BAR-I-DOT  -   MEAN OF I-TH ROW
C         Y2MEAN    -   Y-BAR-DOT-T  -   MEAN OF J-TH COLUMN
C          TOTAL    -   SUM OF X1 OR Y1
C          GMEAN    -   POPULATION MEAN
C             DI    -   (X1MEAN-GMEAN) OR (Y1MEAN-GMEAN)
C             DJ    -   (X2MEAN-GMEAN) OR (Y2MEAN-GMEAN)
C           SUMW    -   SUM OF (X*DJ) OR (Y*DJ)
C             S1    -   SUM OF (DI**2)
C             S2    -   SUM OF (DJ**2)
C              D    -   S1*S2
C             SN    -   SUM OF (X*DI*DJ) OR (Y*DI*DJ)
C           SNSS    -   SN**2
C             SS    -   NON-ADDITIVITY SUM OF SQUARES
C           SSTT    -   TOTAL SUM OF SQUARES
C            SST    -   TREATMENT SUM OF SQUARES
C            SSB    -   BLOCK SUM OF SQUARES
C            SSE    -   ERROR SUM OF SQUARES
C            SSR    -   REAMINDER SUM OF SQUARES
C           NSDF    -   DEGREES OF FREEDOM OF TOTAL
C           NTDF    -   DEGREES OF FREEDOM OF TREATMENT
C           NBDF    -   DEGREES OF FREEDOM OF BLOCK
C           NEDF    -   DEGREES OF FREEDOM OF ERROR
C           NNDF    -   DEGREES OF FREEDOM OF  NON-ADDITIVITY
```

```
C              NRDF   -   DEGREES OF FREEDOM OF REMAINDER
C              SMST   -   MEAN SQUARES TREATMENT
C              SMSB   -   MEAN SQUARES BLOCK
C              SMSE   -   MEAN SQUARES ERROR
C              SMSN   -   MEAN SQUARES NON-ADDITIVITY
C              SMSR   -   MEAN SQUARES REMAINDER
C                FT   -   F VALUE OF TREATMENT
C                FB   -   F VALUE OF BLOCK
C                FN   -   F VALUE OF NON-ADDITIVITY
C
C
C             SUBROUTINE AND FUNCTION SUBPROGRAM REQUIRED
C                 'TUKEY'
C
C
C             METHOD
C
C                 SECTION 1 - THE CALCULATION OF MULTIPLICATIVE INPUT DATA
C                 SECTION 2 - TUKEY'S TEST BEFORE TRANSFORMATION
C                 SECTION 3 - TUKEY'S TEST AFTER TRANSFORMATION
C
C
C ..............................................................................
C
      DIMENSION T(9), B(9), E(9,9), X(9,9),X1(9), X2(9), X1MEAN(9),
     1          X2MEAN(9), Y(9,9), PX(9,9), PPX(9,9), Y1(9), Y2(9),
     2          Y1MEAN(9), Y2MEAN(9), DI(9), DJ(9)
C
      DOUBLE PRECISION SUMW,S1, S2, D, ASN, SNSS, SS, SSTT, SST, SSB, SS
     1E, SSR, SMST, SMSB, SMSE, SMSN, SMSR
C
C
C             SECTION 1 - THE CALCULATION OF MULTIPLICATIVE INPUT DATA
C                             X(I,J)  =  U*T(I)*B(J)*E(I,J)
C
C
C
C             INITIALIZATION
C
   10 U = 0.0
      DO 20 I = 1,9
      DO 20 J = 1,9
      T(I)     = 0.0
      B(J)     = 0.0
      E(I,J)   = 0.0
      X(I,J)   = 0.0
   20 CONTINUE
```

```
C
C          READ INPUT DATA
C
      READ (1,30) N,M
   30 FORMAT (I1,2X,I1)
      IF (N.EQ.0) STOP
      READ (1,40) U,(T(I),I=1,N),(B(J),J=1,M),((E(I,J),J=1,M),I=1,N)
   40 FORMAT (16F5.2)
C
C          COMPUTE THE MULTIPLICATIVE INPUT DATA
C
      DO 50 I = 1,N
      DO 50 J = 1,M
      X(I,J)  = U*T(I)*B(J)*E(I,J)
   50 CONTINUE
C
C
C          SECTION 2 -TUKEY'S TEST BEFORE TRANSFORMATION
C
C
      DO 790 JJ = 0,6
      IF (JJ.GE.1) GO TO 60
      CALL TUKEY (N,M,X,X1,X2,X1MEAN,X2MEAN,TOTAL,GMEAN, DI,DJ,SUMW,S1,S2
     1,D,SN,SNSS,SS,SSTT,SST,SSB,SSE,SSR,NSDF,NTDF,NEDF,NNDF,NRDF,S
     2MST,SMSS,SMSE,SMSN,SMSR,FT,FB,FN)
      GO TO 200
C
C
C          SECTION 3 - TUKEY'S TEST AFTER TRANSFORMATION
C
C
   60 IF (JJ.EQ.3)    GO TO 90
      IF (JJ.GE.4) GO TO 120
C
C          LOG(X) AND LOG(X+1) TRANSFORMATION
C
      DO 80 I = 1,N
      DO 80 J = 1,M
      C       = X(I,J)
      IF (JJ.EQ.2) GO TO 70
      Y(I,J)  = ALOG10(C)
      GO TO 80
   70 Y(I,J)  = ALOG10(C+1)
   80 CONTINUE
      GO TO 170
C
C          ARCSIN(SQRT(X)) TRANSFORMATION
C
   90 TOTAL1   = 0.0
      DO 100 I = 1,N
      DO 100 J = 1,M
  100 TOTAL1   = TOTAL1+X(I,J)
```

```
      DO 110 I = 1,N
      DO 110 J = 1,M
      PX(I,J)  = X(I,J)/TOTAL1
      PPX(I,J) = PX(I,J)*100.0
      PP       = PX(I,J)
      CC       = SQRT(PPX)
      Y(I,J)   = ARSIN(CC)*100.0
  110 CONTINUE
      GO TO 170

            SQRT(X%),SQRT((X+1)%),SQRT(X%)+SQRT((X+1)%) TRANSFOR MATION

  120 DO 160 I = 1,N
      DO 160 J = 1,M
      PPP      = PPX(I,J)
      IF (PPP.LT.80.0) GO TO 130
      PPP      = 100.0-PPP
  130 IF (JJ.EQ.5) GO TO 140
      IF (JJ .EQ.6) GO TO 150
      Y(I,J)   = SQRT(PPP)
      GO TO 160
  140 Y(I,J)   = SQRT(PPP+1.0)
      GO TO 160
  150 Y(I,J)   = SQRT(PPP)+SQRT(PPP+1.0)
  160 CONTINUE

  170 IF (M.EQ.9) GO TO 190
      MM = M+1
      DO 180 I =  1,N
      DO 180 J = MM,9
  180 Y(I,J) = 0.0

  190 CALL TUKEY (N,M,Y,Y1,Y2,Y1MEAN,Y2MEAN,TOTAL,GMEAN,DI,DJ,SUMW,S1,S2
     1,D,SN,SNSS,SS,SSTT,SST,SSB,SSE,SSR,NSDF,NTDF,NBDF,NEDF,NNDF,NRDF,S
     2MST,SMSB,SMSE,SMSN,SMSR,FT,FB,FN)

            PRINT OUTPUT

  200 IF (JJ.EQ.1) GO TO 220
      IF (JJ.EQ.2) GO TO 240
      IF (JJ.EQ.3) GO TO 260
      IF (JJ.EQ.4) GO TO 280
      IF (JJ.EQ.5) GO TO 300
      IF (JJ.EQ.6) GO TO 320
      WRITE (3,210) JJ
  210 FORMAT ('1'/T63, '(',I1,')'/T53,'THE MULTIPLICATIVE DATA '/T54,'BE
     1FORE TRANSFORMATION'/)
      GO TO 340
  220 WRITE (3,230) JJ
  230 FORMAT ('1'/T63,'(',I1,')'/T44,'THE LOGARITHMIC TRANSFORMATION - A
     1LOG10(X)'/)
      GO TO 340
```

```
240 WRITE (3,250) JJ
250 FORMAT ('1'/T63,'(',I1,')'/T43,'THE LOGARITHMIC TRANSFORMATION - A
   1LOG10(X+1)'/)
    GO TO 340
260 WRITE (3,270) JJ
270 FORMAT ('1'/T63,'(',I1,')'/T44,'THE ARCSINE TRANSFORMATION - ARCSI
   1N(SQRT(X))'/)
    GO TO 340
280 WRITE (3,290) JJ
290 FORMAT ('1'/T63,'(',I1,')'/T44,'THE SQUARE ROOT TRANSFORMATION - S
   1QRT(X%)'/)
    GO TO 340
300 WRITE (3,310) JJ
310 FORMAT ('1'/T63,'(',I1,')'/T43,'THE SQUARE ROOT TRANSFORMATION - S
   1QRT((X+1)%)'/)
    GO TO 340
320 WRITE (3,330) JJ
330 FORMAT ('1'/T63,'(',I1,')'/T39,'THE SQUARE ROOT TRANSFORMATION - S
   1QRT(X%)+SQRT((X+1)%('/)
340 WRITE (3,350)
350 FORMAT (T46,'****NON-ADDITIVITY SUM OF SQUARES****'//1X,131('-')/)
    WRITE (3,360)
360 FORMAT (T52,'BLOCK (J)'/)
    WRITE (3,370)
370 FORMAT (T2,'TREATMENT',T11,89('-'),T109,'SUM',T118,'MEAN',T130,'DI
   1'/)
    WRITE (3,380) (J,J = 1,9)
380 FORMAT (T5,'(I)',T15,I1,8(9X,I1)//1X,131('-')//)
    DO 420 I = 1,N
    IF (JJ.GE.1) GO TO 400
    WRITE (3,390) I,(X(I,J),J=1,9),X1(I),X1MEAN(I),DI(I)
390 FORMAT (T6,I1,T11,F9.2,8(1X,F9.2),T102,3(F10.2)/)
    GO TO 420
400 WRITE (3,410) I,(Y(I,J),J=1,9),Y1(I),Y1MEAN(I),DI(I)
410 FORMAT (T6,I1,T11,F9.2,8(1X,F9.2),T102,3(F10.2)/)
420 CONTINUE
    WRITE (3,430)
430 FORMAT (1X,131('-')/)
    IF (JJ.GE.1) GO TO 460
    WRITE (3,440) (X2(J),J = 1,9), TOTAL
440 FORMAT (2X,'SUM',T10,9(F10.2),T101,F11.2/)
    WRITE (3,450) (X2MEAN(J),J = 1,9), GMEAN
450 FORMAT (2X,'MEAN',T10,9(F10.2),T103,'POPULATION MEAN =',F12.2/)
    GO TO 490
460 WRITE (3,470) (Y2(J),J = 1,9), TOTAL
470 FORMAT (2X,'SUM',T10,9(F10.2),T101,F11.2/)
    WRITE (3,480) (Y2MEAN(J),J = 1,9), GMEAN
480 FORMAT (2X,'MEAN',T10,9(F10.2),T103,'POPULATION MEAN = ',F12.2/)
490 WRITE (3,500) (DJ(J),J = 1,9)
500 FORMAT (2X,'DJ',T10,9(F10.2)//1X,131('-')/)
    WRITE (3,510) SUMW, S1, S2, D, SN, SNSS
510 FORMAT (2X,'SUMW =',F50.3//T2,'S1    =',F50.3,5X,'S2    =',F50.3//T2
   1,'D    =',F50.3//2X,'SN    =',F50.3,5X,'SNSS =',F50.3//)
```

```
      WRITE (3,520) SS
  520 FORMAT (T8,'SS NON-ANDDITIVITY  =',F30.3)
      WRITE (3,530)
  530 FORMAT ('1'////T21,'THE ANALYSIS OF VARIANCE FOR RANDOMIZED BLOCKS
     1 DESIGN AND TUKEY''S TEST FOR NON-ADDITIVITY'//)
      WRITE (3,540)
  540 FORMAT (T8,118('*'))
      WRITE (3,550)
  550 FORMAT (T8,'*',T38,'*',T51,'*',T79,'*',T106,'*',T125,'*')
      WRITE (3,560)
  560 FORMAT (T8,'*',T38,'*',T51,'*',T79,'*',T106,'*',T125,'*')
      WRITE (3,570)
  570 FORMAT (T8,'*',T14,'SOURCE OF VARIATION',T38,'*',T44,'DF',T51,'*',
     1T56,'SUM OF SQUARES (SS)',T79,'*',T85,'MEAN SQUARES (MS)',T106,'*'
     2,T115,'F',T125,'*')
      WRITE (3,580)
  580 FORMAT (T8,'*',T38,'*',T51,'*',T79,'*',T206,'*',T125,'*')
      WRITE (3,590)
  590 FORMAT (T8,'*',T38,'*',T51,'*',T79,'*',T106,'*',T125,'*')
      WRITE (3,600)
  600 FORMAT (T8,118('*'))
      WRITE (3,610)
  610 FORMAT (T8,'*',T38,'*',T51,'*',T79,'*',T106,'*',T125,'*')
      WRITE (3,620) NSDF, SSTT
  620 FORMAT (T8,'*',T14,'TOTAL',T38,'*',T44,I2,T51,'*',T56,F19.3,T79,'*
     1' T106,'*',T125,'*')
      WRITE (3,630)
  630 FORMAT (T8,'*',T38,'*',T51,'*',T79,'*',T106,'*',T125,'*')
      WRITE (3,640)
  640 FORMAT (T8,'*',29('-'),'*',12('-'),'*',27('-'),'*',26('-'),'*',18(
     1'-'),'*')
      WRITE (3,650)
  650 FORMAT (T8,'*',T38,'*',T51,'*',T79,'*',T106,'*',T125,'*')
      WRITE (3,660) NTDF, SST,SMST, FT
  660 FORMAT (T8,'*',T14,'TREATMENT',T38,'*',T44,I2,T51,'*',T56,F19.3,T7
     19,'*',T83,F19.3,T106,'*',T112,F13.3,T125,'*')
      WRITE (3,670)
  670 FORMAT (T8,'*',T38,'*',T51,'*',T79,'*',T106,'*',T125,'*')
      WRITE (3,680) NBDF, SSB, SMSB, FB
  680 FORMAT (T8,'*',T14,'BLOCK',T38,'*',T44,I2,T51,'*',T56,F19.3,T79,'*
     1',T83,F19.3,T106,'*',T112,F13.3,T125,'*')
      WRITE (3,690)
  690 FORMAT (T8,'*',T38,'*',T51,'*',T79,'*',T106,'*',T125,'*')
      WRITE (3,700) NEDF, SEE, SMSE
  700 FORMAT (T8,'*',T14,'ERROR',T38,'*',T44,I2,T51,'*',T56,F19.3,T79,'*
     1',T83,F19.3,T106,'*',T125,'*')
      WRITE (3,710)
  710 FORMAT (T8,'*',T38,'*',T51,'*',T79,'*',T106,'*',T125,'*')
      WRITE (3,720)
  720 FORMAT (T8,'*',29('-'),'*',12('-'),'*',27('-'),'*',26('-'),'*',18(
     1'-'),'*')
      WRITE (3,730)
  730 FORMAT (T8,'*',T38,'*',T51,'*',T79,'*',T106,'*',T125,'*')
```

```
      WRITE (3,740) NNDF, SS,SMSN, FN
  740 FORMAT (T8,'*',T14,'NON-ADDITIVITY',T38,'*',T45,I1,T51,'*',T56,F19
     1.3,T79,'*',T83,F19.3,T106,'*',T112,F13.3,T125,'*')
      WRITE (3,750)
  750 FORMAT (T8,'*',T38,'*',T51,'*',T79,'*',T106,'*',T125,'*')
      WRITE (3,760) NRDF, SSR, SMSR
  760 FORMAT (T8,'*',T14,'REMAINDER',T38,'*',T44,I2,T52,'*',T56,F19.3,T7
     19,'*',T83,F19.3,T106,'*',T125,'*')
      WRITE (3,770)
  770 FORMAT (T8,'*',T38,'*',T51,'*',T79,'*',T106,'*',T125,'*')
      WRITE (3,780)
  780 FORMAT (T8,118('*'))
  790 CONTINUE
C
      GO TO 10
      END

      SUBROUTINE TUKEY (N,M,X,X1,X2,X1MEAN,X2MEAN,TOTAL,GMEAN,DI,DJ,SUMW
     1,S1,S2,D,SN,SNSS,SS,SSTT,SST,SSB,SSE,SSR,NSDF,NTDF,NBDF,NEDF,NNDF,
     2NRDF,SMST,SMSB,SMSE,SMSN,SMSR,FT,FB,FN)
C
      DIMENSION X(9,9),X1(9),X2(9),X1MEAN(9),X2MEAN(9),DI(9),DJ(9),W(9)
C
      DOUBLE PRECISION W, SUMW, S1, S2, D, SN, SNSS, SS, CT, TT, SSTT, S
     1X1, SX2, SSX1, SSX2, SST, SSB, SSE, SSR, SMST, SMSB, SMSE, SMSN, S
     2MSR
C
C ......................................................................
C
C
C          PURPOSE
C
C               COMPUTE TUKEY'S TEST FOR NON-ADDITIVITY
C
C
C          DESCRITION OF PARAMETERS
C
C               N   -  THE INPUT NUMBER OF ROWS (#TREATMENTS)
C               M   -  THE INPUT NUMBER OF COLUMNS (#BLOCKS)
C               X   -  THE INPUT MATRIX, N BY M
C              X1 -  X-I-DOT         -  SUM OF I-TH ROW
C              X2  -  X-DOT-J        -  SUM OF J-TH COLUMN
C          X1MEAN  -  X-BAR-I-DOT   -  MEAN OF I-TH ROW
C          X2MEAN  -  X-BAR-DOT-J   -  MEAN OF J-TH COLUMN
C           TOTAL  -  SUM OF X1
C           GMEAN  -  POPULATION MEAN
C              DI  -  (X1MEAN-GMEAN)
C              DJ  -  (X2MEAN-GMEAN)
C            SUMW  -  SUM OF (X*DJ)
C              S1  -  SUM OF (DI**2)
C              S2  -  SUM OF (DJ**2)
C               D  -  S1*S2
C              SN  -  SUM OF (X*DI*DJ)
C            SNSS  -  SN**2
```

```
C                 SS    -   NON-ADDITIVITY SUM OF SQUARES
C               SSTT    -   TOTAL SUM OF SQUARES
C                SST    -   TREATMENT SUM OF SQUARES
C                SSB    -   BLOCK SUM OF SQUARES
C                SSE    -   ERROR SUM OF SQUARES
C                SSR    -   REMAINDER SUM OF SQUARES
C               NSDF    -   DEGREES OF FREEDOM OF TOTAL
C               NTDF    -   DEGREES OF FREEDOM OF TREATMENT
C               NBDF    -   DEGREES OF FREEDOM OF BLOCK
C               NEDF    -   DEGREES OF FREEDOM OF ERROR
C               NNDF    -   DEGREES OF FREEDOM OF NON-ADDITIVITY
C               NRDF    -   DEGREES OF FREEDOM OF REMAINDER
C               SMST    -   MEAN SQUARES TREATMENT
C               SMSB    -   MEAN SQUARES BLOCK
C               SMSE    -   MEAN SQUARES ERROR
C               SMSN    -   MEAN SQUARES NON-ADDITIVITY
C               SMSR    -   MEAN SQUARES REMAINDER
C                 FT    -   F VALUE OF TREATMENT
C                 FB    -   F VALUE OF BLOCK
C                 FN    -   F VALUE OF NON-ADDITIVITY
C
C
C           METHOD
C
C               SECTION 1 - THE CALCULATION OF THE NON-ADDITIVITY SUM
C                           OF SQUARES
C               SECTION 2 - THE CALCULATION OF THE ANALYSIS OF VARIANCE
C                           FOR RANDOMIZED BLOCKS DESIGN AND THE
C                           NON-ADDITIVITY TEST   (TUKEY'S TEST)
C
C
C ......................................................................
C
C
C               SECTION 1 - THE CALCULATION OF THE NON-ADDITIVITY SUM
C                           OF SQVARES
C
C
C           COMPUTE THE SUM OF I-TH ROW - X-I-DOT = X1(I)
C
      DO 10 I = 1,N
   10 X1(I)   = 0.0
      DO 20 I = 1,N
      DO 20 J = 1,M
   20 X1(I)   = X1(I)+X(I,J)
C
C           COMPUTE THE MEAN OF I-TH ROW -X-BAR-I-DOT = X1MEAN(I)
C
      SMM     = M
      DO 30 I = 1,N
   30 X1MEAN(I) = X1(I)/SMM
C
C           COMPUTE THE POPULATION MEAN - GMEAN
```

```
      TOTAL  = 0.0
      DO 40  I = 1,N
   40 TOTAL  = TOTAL+X1(I)
      NM     = N*M
      SNM    = NM
      GMEAN  = TOTAL/SNM
C
C                 COMPUTE DI(I), S1
C
      S1     = 0.0
      DO 50  I = 1,N
      DI(I)  = X1MEAN(I)-GMEAN
   50 S1     = S1+DI(I)**2
C
C                 COMPUTE THE SUM OF J-TH COLUMN - X-DOT-J = X2(J)
C
      DO 60  J = 1,9
   60 X2(J)     = 0.0
      DO 70  J = 1,9
      DO 70  I = 1,N
   70 X2(J)     = X2(J)+X(I,J)
C
C                 COMPUTE THE MEAN OF J-TH COLUMN - X-BAR-DOT-J = X2MEAN(J)
C
      SNN    = N
      DO 80  J = 1,9
   80 X2MEAN(J) = X2(J)/SNN
C
C                 COMPUTE DJ(J), S2
C
      S2     = 0.0
      DO 110 J = 1,9
      IF (X2(J)) 90, 100, 90
   90 DJ(J) = X2MEAN(J)-GMEAN
      S2     = S2+DJ(J)**2
      GO TO 110
  100 DJ(J) = 0.0
  110 CONTINUE
C
C                 COMPUTE W(I)
C
      DO 120 I = 1,N
  120 W(I) = 0.0
      DO 130 I = 1,N
      DO 130 J = 1,M
  130 W(I) = W(I)+X(I,J)*DJ(J)
C
C                 COMPUTE THE SUM OF W - SUMW
C
      SUMW = 0.0
      DO 140 I = 1,N
  140 SUMW = SUMW+W(I)
C
C                 COMPUTE SM.D
```

```
      SN = 0.0
      DO 150 I = 1,N
  150 SN = SN+W(I)*DI(I)
      D  = S1*S2
C
C               COMPUTE SNSS,THE NON-ADDITIVITY SUM OF SQUARES - SS
C
      SNSS = SN**2
      SS   = SNSS/D
C
C
C                    SECTION 2 - THE CALCULATION OF THE ANALYSIS OF VARIANCE
C                                FOR RANDOMIZED BLOCKS DESIGN
C                                AND THE NON-ADDITIVITY TEST (TUKEY'S TEST)
C
C               COMPUTE TOTAL SUM OF SQUARE - SSTT
C
      CT   = TOTAL**2/SNM
      TT   = 0.0
      DO 160 I = 1,N
      DO 160 J = 1,M
  160 TT   = TT+X(I,J)**2
      SSTT = TT-CT
C
C               COMPUTE TREATMENT SUM OF SQUARES - SST
C
      SX1  = 0.0
      DO 170 I = 1,N
  170 SX1  = SX1+X1(I)**2
      SSX1 = SX1/SMM
      SST  = SSX1-CT
C
C               COMPUTE BLOCK SUM OF SQUARES - SSB
C
      SX2  = 0.0
      DO 180 J = 1,M
  180 SX2  = SX2+X2(J)**2
      SSX2 = SX2/SNN
      SSB  = SSX2-CT
C
C               COMPUTE ERROR SUM OF SQUARE - SSE
C
      SSE  = SSTT-(SST+SSB)
C
C               COMPUTE REMAINDER SUM OF SQUARES - SSR
C
      SSR  = SSE-SS
C
C               COMPUTE DEGREES OF FREEDOM - DF
C
      NSDF = N*M-1
      MTDF = N-1
```

```
      NBDF = M-1
      NEDF = (N-1)*(M-1)
      NNDF = 1
      NRDF = NEDF-1
C
C           COMPUTE MEAN SQUARES - MS
C
      SNTDF = NTDF
      SNBDF = NBDF
      SNEDF = NEDF
      SNRDF = NRDF
C
      SMST = SST/SNTDF
      SMSB = SSB/SNBDF
      SMSE = SSE/SNEDF
      SMSN = SS
      SMSR = SSR/SNRDF
C
C           COMPUTE F VALUE
C
      FT = SMST/SMSE
      FB = SMSB/SMSE
      FN = SMSN/SMSR
C
      RETURN
      END
C .......................................................................
C
C
C
C        PROGRAM   FRIEDMAN'S TEST FOR TWO-WAY ANALYSIS OF VARIANCE
C
C
C           PURPOSE
C
C
C              TEST WHETHER A NUMBER OF SAMPLES ARE FROM THE SAME
C              POPULATION BY FRIEDMAN TWO-WAY ANALYSIS OF VARIANCE
C              TEST.
C
C
C           DESCRIPTION OF PARAMETERS
C
C              N  -  THE INPUT NUMBER OF ROWS (#TREATMENTS)
C              M  -  THE INPUT NUMBER OF COLUMNS (#BLOCKS)
C              U  -  THE INPUT POPULATION MEAN
C              T  -  THE INPUT TREATMENT EFFECT
C              B  -  THE INPUT BLOCK EFFECT
C              E  -  THE INPUT EXPERIMENTAL ERROR
C              X  -  THE OUTPUT MATRIX, N BY M OF THE MULTIPLICATIVE
C                    INPUT DATA FOR TESTING FRIEDMAN'S TEST
```

```
C                A   -   INPUT MATRIX, N BY M OF THE ORIGINAL DATA
C                R   -   OUTPUT MATRIX, N BY M OF THE RANKED DATA
C               RT   -   SUM OF RANKS OF EACH CASE (OUTPUT)
C               ST   -   FRIEDMAN STATISTIC OF TREATMENT (OUTPUT)
C               SB   -   FRIEDMAN STATISTIC OF BLOCK (OUTPUT)
C              TDF   -   TREATMENT NUMBER OF DEGREES OF FREEDOM (OUTPUT)
C              BDF   -   BLOCK NUMBER OF DEGREES OF FREEDOM (OUTPUT)
C
C
C           SUBROUTINE AND FUNCTION SUBPROGRAM REQUIRED
C
C               TWOAV    AND    RANK
C
C
C           METHOD
C
C              DESCRIBED IN S. SIEGEL,  NONPARAMETRIC STATISTICS FOR
C              THE BEHAVIORAL SCIENCES , MCGRAW-HILL, NEW YORK, 1956,
C              CHAPTER 7.
C
C
C ..................................................................
C
C
      DIMENSION T(9), B(9), E(9,9), X(9,9), XX(9,9), A(81), AA(81),
     1          R(81), RR(81), RT(9), RRT(9)
C
   10 U        = 0.0
      DO 20 I  = 1,9
      DO 20 J  = 1,9
      T(I)     = 0.0
      B(J)     = 0.0
      E(I,J)   = 0.0
      X(I,J)   = 0.0
   20 CONTINUE
C
C           READ INPUT DATA
C
      READ (1,30) N,M
   30 FORMAT (I1,2X,I1)
      IF (N.EQ.0) STOP
      READ (1,40) U,(T(I),I=1,N),(B(J),J=1,M),((E(I,J),J=1,M),I=1,N)
   40 FORMAT (16F5.2)
C
C      -    COMPUTE THE MULTIPLICATIVE INPUT DATA
C
      DO 50 I  = 1,N
      DO 50 J  = 1,M
      X(I,J)   = U*T(I)*B(J)*E(I,J)
   50 CONTINUE
C
C           SET K=1 TO TEST TREATMENT
            SET K=2 TO TEST BLOCK
```

```
      DO 520 K = 1,2
      IF (K.EQ.2) GO TO 70
      DO 60 I  = 1,9
      DO 60 J  = 1,9
      V        = X(I,J)
      XX(J,I)  = V
   60 CONTINUE
      NN       = M
      MM       = N
C
      CALL TWOAV (NN,MM,XX,AA,BB,BBT,SBT,BDF)
      GO TO 80
   70 CALL TWOAV (N,M,X,A,B,BT,SB,BDF)
      GO TO 380
C
C             PRINT THE MULTIPLICATIVE DATA
C
   80 WRITE (3,90)
   90 FORMAT ('1'////T54,'THE  MULTIPLICATIVE  DATA '//T53,'X(I,J)  =  U
     1*T(I)*B(J)*E(I,J)'///)
      WRITE (3,100)
  100 FORMAT (T23,109('*')/T23,'I',T131,'I'/T23,'I',T72,'BLOCK  (J)',T1
     131,'I'/T23,'I',T131,'I')
      WRITE (3,110)
  110 FORMAT (T14,9('*'),'I',107('*'),'I')
      WRITE (3,120)
  120 FORMAT (T14,'I   U    I',9(11X,'I'))
      WRITE (3,130) (U,(J,J = 1,9))
  130 FORMAT (T14,'I',T16,F6.2,T23,'I',9(5X,I1,5X,'I'))
      WRITE (3,140)
  140 FORMAT (T14,'I',T23,'I',9(11X,'I'))
      WRITE (3,150)
  150 FORMAT (T2,12('*'),'I',8('*'),'I',9(11('*'),'I'))
      WRITE (3,160)
  160 FORMAT (T2,'I',T14,'I .',T23,'I',9(11X,'I'))
      WRITE (3,170)
  170 FORMAT (T2,'I TREATMENT I  .B(J) I',9(4X,F6.2,1X,'I'))
      WRITE (3,180)
  180 FORMAT (T2,'I' T7,'(I)',T14,'I',T19,'.   I',9(11X,'I'))
      WRITE (3,190)
  190 FORMAT (T2,'I',T14,'I T(I).  I',9(11X,'I'))
      WRITE (3,200)
  200 FORMAT (T2,'I',11('-'),'I',8('-'),'I',9(11('-'),'I'))
      DO 220 I = 1,N
      WRITE (3,210) (I,T(I),(X(I,J),J = 1,9))
  210 FORMAT (T2,'I',T14,'I',T23,'I',9(11X,'I')/T2,'I',T8,I1,T14,'I',T16
     1,F6.2,T23,'I',9(1X,F9.2,1X,'I'))
  220 CONTINUE
      WRITE (3,230)
  230 FORMAT (T2,'I',T14,'I',T23,'I',9(11X,'I')/1X,130('*'))
```

```
C
C              PRINT TABLE (1) = FRIEDMAN'S TEST FOR TREATMENT
C
      WRITE (3,240)
  240 FORMAT ('1'///T58,'TABLE  (1)'//T47,'FRIEDMAN''S TEST FOR TREATMEN
     1T'////1X,131('*'))
      WRITE (3,250)
  250 FORMAT (T2,'*',T6,'*',T132,'*'/T2,'*',T6,'*',T62,'TREATMENT',T132,
     1'*'/T2,'*',T6,'*',T132,'*')
      WRITE (3,260)
  260 FORMAT (T2,'BLOCK',126('*'))
      WRITE (3,270)
  270 FORMAT (T2,'*',T6,'*',9(6X,I1,6X,'*'))
      WRITE (3,280)
  280 FORMAT (T2,'*',T6,'*',126('*'))
      WRITE (3,290)
  290 FORMAT (T2,'*',T6,'*',9(4X,'A',4X,'* R  *'))
      WRITE (3,300)
  300 FORMAT (T2,'*',T6,'*',9(9X,'*    *')/1X,131('*'))
      DO 320 I = 1,NN
      WRITE (3,310) I, AA(I),RR(I), AA(I+NN),RR(I+NN), AA(I+2*NN),RR(I+2
     1*NN), AA(I+3*NN),RR(I+3*NN), AA(I+4*NN),RR(I+4*NN), AA(I+5*NN),RR(
     2I+5*NN), AA(I+6*NN),RR(I+6*NN), AA(I+7*NN),RR(I+7*NN), AA(I+8*NN),
     3RR(I+8*NN)
  310 FORMAT (T2,'*',T6,'*',9(9X,'*',3X,'*')/T2,'*',1X,I1,1X,'*',9(F9.2,
     1'*',F3.1,'*'))
  320 CONTINUE
      WRITE (3,330)
  330 FORMAT (T2,'*',T6,'*',9(9X,'*    *'))
      WRITE (3,340)
  340 FORMAT (1X,131('*'))
      WRITE (3,350) (RRT(J),J = 1,9)
  350 FORMAT (6X,9(9X,F4.1)/6X,9(9X,'****'))
      WRITE (3,360) ST
  360 FORMAT (//T19,'......ST(CALCULATE)....   =',F20.3,T84,'.... ST(TABL
     1E)....   =',2X,15('.'))
      WRITE (3,370) TDF
  370 FORMAT (//T19,'..DEGREES OF FREEDOM...   =',T63,I2)
      GO TO 530
C
C              PRINT TABLE (2) = FRIEDMAN'S TEST FOR BLOCK
C
  380 WRITE (3,390)
  390 FORMAT ('1'///T60,'TABLE  (2)'//T49,'FRIEDMAN''S TEST FOR BLOCK'//
     1//1X,131('*'))
      WRITE (3,400)
  400 FORMAT (T2,'*',T6,'*',T132,'*'/T2,'*',T6,'*'T64,'BLOCK',T132,'*'/
     1T2,'*',T6,'*',T132,'*')
      WRITE (3,410)
  410 FORMAT (T2,'TREAT',126('*'))
```

```
      WRITE (3,420) (J,J = 1,9)
  420 FORMAT (T2,'*',T6,'*',9(6X,I1,6X,'*'))
      WRITE (3,430)
  430 FORMAT (T2,'MENT*',126('*'))
      WRITE (3,440)
  440 FORMAT (T2,'*',T6,'*',9(4X,'A',4X,'* R *'))
      WRITE (3,450)
  450 FORMAT (T2,'*',T6,'*',9(9X,'*   *')/1X,131('*'))
      DO 470 I = 1,N
      WRITE (3,460) I, A(I),R(I), A(I+N),R(I+N), A(I+2*N),R(I+2*N), A(I+
     13*N),R(I+3*N), A(I+4*N),R(I+4*N), A(I+5*N),R(I+5*N), A(I+6*N),R(I+
     26*N), A(I+7*N),R(I+7*N), A(I+8*N),R(I+8*N)
  460 FORMAT (T2,'*',T6,'*',9(9X,'*',3X,'*')/T2,'*',1X,I1,1X,'*',9(F9.2,
     1'*',F3.1,'*'))
  470 CONTINUE
      WRITE (3,480)
  480 FORMAT (T2,'*',T6,'*',9(9X,'*   *'))
      WRITE (3,490)
  490 FORMAT (1X,131('*'))
      WRITE (3,500) (RT(J),J = 1,9)
  500 FORMAT (6X,9(9X,F4.1)/6X,9(9X,'****'))
      WRITE (3,510) SB
  510 FORMAT (//T19,'......SB(CALCULATE)....  =',F20.3,T84,'.....SB(TABL
     1E)....  =',2X,15('.'))
      WRITE (3,520) BDF
  520 FORMAT (//T19,'..DEGREES OF FREEDOM...  =',T63,I2)
  530 CONTINUE
C
      GO TO 10
      END
```

```
      SUBROUTINE TWOAV (N,M,X,A,R,RT,S,NDF)
      DIMENSION X(9,9), A(81), W(18), R(81), RT(9)
C
C
C.......................................................................
C
C
C        PURPOSE
C
C
C           TEST WHETHER A NUMBER OF SAMPLES ARE FROM THE SAME
C           POPULATION BY FRIEDMAN TWO-WAY ANALYSIS OF VARIANCE.
C           TEST.
C
C
C        DESCRIPTION OF PARAMETERS
C
C           N   -  THE INPUT NUMBER OF ROWS (#TREATMENTS)
C           M   -  THE INPUT NUMBER OF COLUMNS (#BLOCKS)
C           X   -  THE INPUT MATRIX, N BY M
C           A   -  INPUT MATRIX, N BY M, OF THE ORIGINAL DATA
C           R   -  OUTPUT MATRIX, N BY M, OF THE RANKED DATA
C           W   -  WORK AREA OF LENGTH 2*M
C          RT   -  SUM OF RANKS OF EACH CASE (OUTPUT)
C           S   -  FRIEDMAN STATISTIC (OUTPUT)
C          NDF  -  NUMBER OF DEGREES OF FREEDOM (OUTPUT)
C
C
C        SUBROUTINE AND FUNCTION SUBPROGRAM REQUIRED
C
C            RANK
C
C
C        METHOD
C
C           DESCRIBED IN S. SIEGEL,  NONPARAMETRIC STATISTICS FOR
C           THE BEHAVIORAL SCIENCES , MCGRAW-HILL, NEW YORK, 1956,
C           CHAPTER 7.
C
C
C.......................................................................
C
C
C
C        INITIALIZATION
C
      DO 10 L = 1,81
      A(L) = 0.0
      R(L) = 0.0
   10 CONTINUE
      L    = 1
      DO 30 J = 1,M
      DO 20 I = 1,N
      A(L) = X(I,J)
```

```
      L      = L+1
   20 CONTINUE
   30 CONTINUE
C
C            RANK DATA IN EACH GROUP AND ASSIGN TIED OBSERVATIONS AVERAGE
C            OF TIED RANK
C
      DO 50 I = 1,N
      IJ   =  I-N
      IK   =  IJ
      DO 40 J = 1,M
      IJ   =  IJ+N
   40 W(J)=  A(IJ)
      CALL RANK (W,W(M+1),M)
      DO 50 J = 1,M
      IK   =  IK+N
      IW   =  M+J
   50 R(IK)    = W(IW)
C
C            COMPUTE SUM OF SQUARES OF SUM OF RANKS
C
      RTSQ   = 0.0
      IR     = 0
      DO 70 J = 1,M
      RT(J) = 0.0
      DO 60 I = 1,N
      IR     = IR+1
   60 RT(J)  = RT(J)+R(IR)
   70 RTSQ  = RTSQ+RT(J)**2
C
C            COMPUTE FRIEDMAN STATISTIC, S
C
      FNM = N*(M+1)
      FM  = M
      XR  = (12.0/(FM*FNM))*RTSQ-3.0*FNM
C
C            COMPUTE DEGREES OF FREEDOM
C
      NDF = M-1
C
      RETURN
      END
```

```
      SUBROUTINE RANK(W,R,M)
      DIMENSION W(9), R(9)
C
C
C .......................................................................
C
C
C              PURPOSE
C
C                 RANK A VECTOR OF VALUES.
C
C
C              DESCRIPTION OF PARAMETERS
C
C                   W  -  INPUT VECTOR OF M VALUES
C                   R  -  OUTPUT VECTOR OF LENGTH M, SMALLEST VALUE IS
C                         RANKED 1, LARGEST IS RANKED M.  TIES ARE
C                         ASSIGNED AVERAGE OF TIED RANKS.
C                   M  -  NUMBER OF VALUES (INPUT)
C
C
C              METHOD
C
C                   VECTOR IS SEARCHED FOR SUCCESSIVELY LARGER ELEMENTS, IF
C                   TIES OCCUR, THEY ARE LOCATED AND THEIR RANK VALUE
C                   CALCULATED. FOR EXAMPLE, IF 2 VALUES ARE TIED FOR SIXTH
C                   RANK, THEY ARE ASSIGNED A RANK OF 6.5 ( = (6+7)/2 ).
C
C
C .......................................................................
C
C
C              INITIALIZATION
C
      DO 10 I = 1,M
   10 R(I) = 0.0
C
C              FIND RANK OF DATA
C
      DO 100 I = 1,M
C
C              TEST WHETHER DATA POINT IS ALREADY RANKED
C
      IF (R(I)) 20, 20, 100
C
C              DATA POINT TO BE RANKED
C
   20 SMALL = 0.0
      EQUAL = 0.0
      X     = W(I)          X = W(I)
      DO 50 J = 1,M
      IF (W(J)-X) 30, 40, 50          <  ?  or  <     SMALL = SMALL+1
C
C              COUNT NUMBER OF DATA POINTS WHICH ARE SMALLER
```

```
   30 SMALL = SMALL+1.0
      GO TO 50
C
C              COUNT NUMBER OF DATA POINTS WHICH ARE EQUAL
C
   40 EQUAL = EQUAL+1.0
      R(J)  = -1.0
   50 CONTINUE
C
C              TEST FOR TIE
C
      IF (EQUAL-1.0) 60, 60, 70
C
C              STORE RANK OF DATA POINT WHERE NO TIE
C
   60 R(I) = SMALL+1.0
      GO TO 100
C
C              COMPUTE RANK OF TIED DATA POINTS
C
   70 P = SMALL+(EQUAL+1.0)*0.5
      DO 90 J = 1,M
      IF (R(J)+1.0) 90, 80, 90
   80 R(J) = P
   90 CONTINUE
  100 CONTINUE
C
      RETURN
      END
```

```
C ..........................................................................
C
C
C
C      PROGRAM  'THE HOMOGENEITY OF VARIANCE'
C
C
C
C         PURPOSE
C
C            COMPUTE BARTLETT'S TEST FOR HOMOGENEITY OF VARIANCE
C            BEFORE AND AFTER TRANSFORMATION.
C                  ONE OF THE BASIC ASSUMPTIONS UNDERLYING THE MODEL
C                  X(I,J)  =  U+T(I)+B(J)+E(I,J) IS THAT THE VARIANCE
C                  DUE TO EXPERIMENTAL ERROR WITH IN EACH OF THE
C                  TREATMENT POPULATION  BE HOMOGENEOUS.  MODERATE
C                  DEPARTURES FROM THIS ASSUMPTION DO NOT.  THAT IS,
C                  WHEN THE VARIANCE  IN THE POPULATION ARE NOT EQUAL
C                  USES BARTLETT'S TEST FOR HOMOGENEITY OF VARIANCE.
C
C
C         DESCRIPTION OF PARAMETERS
C
C             N   -  THE INPUT NUMBER OF ROWS (#TREATMENTS)
C             M   -  THE INPUT NUMBER OF COLUMNS (#BLOCKS)
C             U   -  THE INPUT POPULATION MEAN
C             T   -  THE INPUT TREATMENT EFFECT
C             B   -  THE INPUT BLOCK EFFECT
C             E   -  THE INPUT EXPERIMENTAL ERROR
C             X   -  THE OUTPUT MATRIX, N BY M OF THE ADDITIVE INPUT
C                    DATA  FOR TESTING THE HOMOGENEITY OF VARIANCE  .
C             Y   -  THE OUTPUT TRANSFORMATION MATRIX, N BY M
C            SS   -  THE VARIANCE (OUTPUT)
C            SD   -  THE STANDARD DEVIATION (OUTPUT)
C             Z   -  THE LOGARITHMIC OF VARIANCE (OUTPUT)
C          SUMZ   -  THE SUM OF Z (OUTPUT)
C         SUMSS   -  THE SUM OF VARIANCE (OUTPUT)
C            HT   -  THE BARTLETT STATISTIC OF TREATMENT (OUTPUT)
C           TDF   -  THE TREATMENT NUMBER OF DEGREES OF FREEDOM (OUTPUT)
C
C
C         SUBROUTINE AND FUNCTION SUBPROGRAM REQUIRED
C
C            'BARTL'
C
C
C         METHOD
C
C            DESCRIBED IN B.J. WINER, 'STATISTICAL PRINCIPLES IN
C            EXPERIMENTAL DESIGN', MCGRAW-HILL, TOKYO, 1971,
C            CHAPTER 3.
C
C
C ..........................................................................
```

```
C
      DIMENSION T(9), B(9), E(9,9), X(9,9), Y(9,9), PX(9,9), PPX(9,9),
     1           SS(9), SD(9), Z(9)
C
C
C                   SECTION 1   THE CALCULATION OF ADDITIVE INPUT DATA
C                                     X(I.J)   =   U+T(I)+B(J)+E(I,J)
C
C
C               INITIALIZATION
C
   10 U = 0.0
      DO 20 I  = 1,9
      DO 20 J  = 1,9
      T(I)     = 0.0
      B(J)     = 0.0
      E(I,J)   = 0.0
      X(I,J)   = 0.0
   20 CONTINUE
C
C               READ INPUT DATA
C
      READ (1,30) N,M
   30 FORMAT (I1,2X,I1)
      IF (M.EQ.0) STOP
      READ (1,40) U,(T(I),I=1,N),(B(4),J=1,M),((E(I,J),J=1,M),I=1,N)
   40 FORMAT (16F5.2)
C
C               COMPUTE THE ADDITIVE INPUT DATA
C
      DO 50 I  = 1,N
      DO 50 J  = 1,M
      X(I,J)   = U+T(I)+B(J)+E(I,J)
   50 CONTINUE
C
C
C                   SECTION 2   THE BARTLETT'S TEST BEFORE TRANSFORMATION
C
C
      DO 660 JJ = 0,6
      IF (JJ.GE.1) GO TO 60
      CALL BARTL (N,M,X,SS,SD,Z,SUMZ,SUMSS,HT,TDF)
      GO TO 200
C
C                   SECTION 3   BARTLETT'S TEST AFTER TRANSFORMATION.
C
C
   60 IF (JJ.EQ.3) GO TO 90
      IF (JJ.GE.4) GO TO 120
C
C               DATA ARE TRANSFORMED  BY ALOG10(X) AND ALOG10(X+1)
C
```

```
      DO 80 I = 1,N
      DO 80 J = 1,M
      C = Y(I,J)
      IF (JJ.EQ.2) GO TO 70
      Y(I,J) = ALOG10(C)
      GO TO 80
   70 Y(I,J) = ALOG10(C+1)
   80 CONTINUE
      GO TO 170
C
C              DATA ARE TRANSFORMED BY ARCSIN(SQRT(X))C
   90 TOTAL1  = 0.0
      DO 100 I = 1,N
      DO 100 J = 1,M
  100 TOTAL1   = TOTAL1+X(I,J)
      DO 110 J = 1,N
      DO 110 J = 1,M
      PX(I,J)  = X(I,J)/TOTAL1
      PPX(I,J) = PX(I,J)*100.0
      PP       = PX(I,J)
      CC       = SQRT(PP)
      Y(I,J)   = ARSIN(CC)*100.0
  110 CONTINUE
      GO TO 170
C
C              DATA ARE TRANSFORMED BY SQRT(X%),SQRT((X+1)%) AND
C                                     SQRT(X%)+SQRT((X+1)%)
C
  120 DO 160 I = 1,N
      DO 160 J = 1,M
      PPP = PPX(I,J)
      IF (PPP.LT.80.0) GO TO 130
      PPP = 100.0-PPP
  130 IF (JJ.EQ.5) GO TO 140
      IF (JJ.EQ.6) GO TO 150
      Y(I,J)   = SQRT(PPP)
      GO TO 160
  140 Y(I,J)   = SQRT(PPP+1.0)
      GO TO 160
  150 Y(I,J)   = SQRT(PPP)+SQRT(PPP+1.0)
  160 CONTINUE
C
  170 IF (M.EQ.9) GO TO 190
      MM = M+1
      DO 180 I =  1,N
      DO 180 J = MM,9
  180 Y(I,J)   = 0.0
C
  190 CALL BARTL (N,M,Y,SS,SD,Z,SUMZ,SUMSS,MT,TDF)
C
C              PRINT OUTPUT
C
  200 IF (JJ.EQ.1) GO TO 370
      IF (JJ.EQ.2) GO TO 390
```

```
      IF (JJ.EQ.3) GO TO 410
      IF (JJ.EQ.4) GO TO 430
      IF (JJ.EQ.5) GO TO 450
      IF (JJ.EQ.6) GO TO 470
      WRITE (3,210) JJ
  210 FORMAT ('1'/T63,'(',I1,')'/T56,'THE ADDITIVE DATA'/T53,'X(I,J)  =
     1U+T(I)+B(J)+E(I,J)'//T54,'BEFORE TRASFORMATION'///)
      WRITE (3,220)
  220 FORMAT (T23,109('*')/T23,'I',T131,'I'/T23,'I',T72,'BLOCK   (J)',T
     131,'I'/T23,'I',T131,'I')
      WRITE (3,230)
  230 FORMAT (T14,9('*'),'I',107('*'),'I')
      WRITE (3,240)
  240 FORMAT (T14,'I    U     I',9(11X,'I'))
      WRITE (3,250) (U,(J,J = 1,9))
  250 FORMAT (T24,'I',T26,F6.2,T23,'I',9(5X,I1,5X,'I'))
      WRITE (3,260)
  260 FORMAT (T14,'I',T23,'I',9(11X,'I'))
      WRITE (3,270)
  270 FORMAT (T2,12('*'),'I',8('*'),'I',9(11('*'),'I'))
      WRITE (3,280)
  280 FORMAT (T2,'I',T24,'I .',T23,'I',9(11X,'I'))
      WRITE (3,290)
  290 FORMAT (T2,'I TREATMENT I  .B(J) I',9(4X,F6.2,1X,'I'))
      WRITE (3,300)
  300 FORMAT (T2,'I',T7,'(I)',T14,'I',T29,'.   I',9(11X,'I'))
      WRITE (3,310)
  310 FORMAT (T2,'I',T14,'I T(I).   I',9(11X,'I'))
      WRITE (3,320)
  320 FORMAT (T2,'I',11('-'),'I',8('-'),'I',9(11('-'),'I'))
      DO 340 I = 1,N
      WRITE (3,330) (I,T(I),(X(I,J),J = 1,9))
  330 FORMAT (T2,'I',T24,'I',T23,'I',9(11X,'I')/T2,'I',T8,I1,T14,'I',T16
     1,F6.2,T23,'I',9(1X,F9.2,1X,'I'))
  340 CONTINUE
      WRITE (3,350)
  350 FORMAT (T2,'I',T14,'I',T23,'I',9(11X,'I')/1X,130('*'))
      WRITE (3,360)
  360 FORMAT ('1'/)
      GO TO 490
  370 WRITE (3,380) JJ
  380 FORMAT ('1'/T63,'(',I1,')'/T44,'THE LOGARITHMIC TRANSFORMATION   A
     1LOG10(X)'/)
      GO TO 490
  390 WRITE (3,400) JJ
  400 FORMAT ('1'/T63,'(',I1,')'/T43,'THE LOGARITHMIC TRANSFORMATION   A
     1LOG10(X+1)'/)
      GO TO 490
  410 WRITE (3,420)JJ
  420 FORMAT ('1'/T63,'(',I1,')'/T44,'THE ARCSINE TRANSFORMATION   ARCSI
     1N(SQRT(X))'/)
      GO TO 490
  430 WRITE (3,440) JJ
  440 FORMAT ('1'/T63,'(',I1,')'/T44,'THE SQUARE ROOT TRANSFORMATION   S
```

```
  1QRT(X%)'/)
    GO TO 490
450 WRITE (3,460) JJ
460 FORMAT ('1'/T63,'(',I1,')'/T43,'THE SQUARE ROOT TRANSFORMATION    S
   1QRT((X+1)%)'/)
    GO TO 490
470 WRITE (3,480) JJ
480 FORMAT ('1'/T63,'(',I1,')'/T39,'THE SQUARE ROOT TRANSFORMATION    S
   1QRT(X%)+SQRT((X+1)%)'/)
490 WRITE (3,500)
500 FORMAT (T49,'TEST FOR THE HOMOGENEITY OF VARIANCE')
    WRITE (3,510)
510 FORMAT (T47,'   ( BARTLETT''S TEST FOR TREATMENT )   '//T34,66('-'
   1))
    WRITE (3,520)
520 FORMAT (T34,'I',T48,'I',T61,'I',T89,'I',T99,'I')
    WRITE (3,530)
530 FORMAT (T34,'I',T37,'TREATMENT',T48,'I',T51,'STANDARD',T61,'I',T71
   1, 'VARIANCE',T89,'I',T91,'LOG(SS)',T99,'I')
    WRITE (3,540)
540 FORMAT (T34,'I',T40,'(I)',T48,'I',T51,'DEVIASION',T61,'I',T73,'(SS
   1)',T89,'I',T99,'I')
    WRITE (3,550)
550 FORMAT (T34,'I',T48,'I',T61,'I',T89,'I',T99,'I')
    WRITE (3,560)
560 FORMAT (T34,'I',13('-'),'I',12('-'),'I',27('-'),'I',9('-'),'I')
    WRITE (3,570)
570 FORMAT (T34,'I',T48,'I',T61,'I',T89,'I',T99,'I')
    DO 590 I = 1,N
    WRITE (3,580) I, SD(I), SS(I), Z(I)
580 FORMAT (T34,'I',T41,I1,T48,'I',F10.2,T61,'I',2X,F23.2,T89,'I',T91,
   1F6.3,T99,'I'/T34,'I'T48,'I',T61,'I',T89,'I',T99,'I')
590 CONTINUE
    WRITE (3,600)
600 FORMAT (T34,2(27('-'),'I'),9('-'),'I')
    WRITE (3,610)
610 FORMAT (T61,'I',T89,'I',T99,'I')
    WRITE (3,620) SUMSS, SUMZ
620 FORMAT (T46,'TOTAL',T61,'I',F25.2,T89,'I',F7.3,T99,'I')
    WRITE (3,630)
630 FORMAT (T61,'I',T89,'I',T99,'I')
    WRITE (3,640)
640 FORMAT (T61,39('-')/)
    WRITE (3,650) HT, TDF
650 FORMAT (T76,'HT =',F11.3,T94,'DF = ',I1)
660 CONTINUE
C
    GO TO 10
    END
```

```
      SUBROUTINE BARTL (N,M,X,SS,SD,Z,SUMZ,SUMSS,H,KDF)
      DIMENSION X(9,9), SUMX1(9), SUMX2(9), AMEAN(9),AAMEAN(9), SS(9),
     1          SD(9), Z(9)
      DOUBLE PRECISION SUMXZ, AAMEAN
C
C
C ....................................................................
C
C
C            PURPOSE
C
C               COMPUTE BARTLETT'S TEST FOR HOMOGENEITY OF VARIANCE.
C
C
C            DESCRIPTION OF PARAMETERS
C
C               N   -  THE INPUT NUMBER OF ROWS (#TREATMENTS)
C               M   -  THE INPUT NUMBER OF COLUMNS (#BLOCKS)
C               X   -  THE INPUT MATRIX, N BY M
C              SS   -  THE VARIANCE (OUTPUT)
C              SD   -  THE STANDARD DDEVIATION (OUTPUT)
C               Z   -  THE LOGARITHMIC OF VARIANCE (OUTPUT)
C            SUMZ   -  THE SUM OF Z (OUTPUT)
C           SUMSS   -  THE SUM OF VARIANCE (OUTPUT)
C               H   -  THE BARTLETT STATISTIC (OUTPUT)
C             KDF   -  THE NUMBER OF DEGREES OF FREEDOM (OUTPUT)
C
C
C            METHOD
C
C               DESCRIBED IN B.J. WINER, 'STATISTICAL PRINCIPLES IN
C               EXPERIMENTAL DESIGN', MEGRAW-HILL, TOKY+, 1971,
C               CHAPTER 3
C
C
C ....................................................................
C
C
      DO 10 I   = 1,N
      SUMX1(I) = 0.0
      SUMX2(I) = 0.0
   10 CONTINUE
      FM = M
      DO 40 I   = 1,N
      DO 20 J   = 1,M
      SUMX1(I) = SUMX1(I)+X(I,J)
      SUMX2(I) = SUMX2(I)+X(I,J)**2
   20 CONTINUE
      AMEAN(I)   = SUMX1(I)/FM
      AAMEAN(I)  = AMEAN(I)**2
      SS(I)      = (SUMX2(I)/FM)-AAMEAN(I)
      B       = SS(I)
      SD(I) = SQRT(B)
      IF (B.EQ.0.0) GO TO 30
```

```
      Z(I)   = ALOG10(B)
      GO TO 40
   30 Z(I)   =ALOG10(B+1.0)
   40 CONTINUE
      SUMZ   = 0.0
      SUMSS  = 0.0
      DO 50 I = 1,N
      SUMZ   = SUMZ+Z(I)
      SUMSS  = SUMSS+SS(I)
   50 CONTINUE
      FN  = N
      SSS = SUMSS/FN
      C   = ALOG10(SSS)
      D   = 2.3026*FN*(FN*C-SUMZ)
      G   =(FN+1.0)/(3.0*FN*FN)+1.0
      H   = D/G
      KDF = N-1
      RETURN
      END
```

Table A. *Upper tail probabilities for the null distribution of Friedman's S statistic:*
t = 3, r = 2(1)13; t = 4, r = 2(1)8; t = 5, r = 3, 4, 5

For given t and r, the tabled entry for the point x is $P_0\{S \geq x\}$. Under these conditions, if x is such that $P_0\{S \geq x\} = \alpha$, then $s(\alpha, t, r) = x$. For given t and r, the entries are terminated at $x_{t,r}$ where $x_{t,r}$ is the smallest value of x such that $P_0\{S \geq x\}$ is zero to three decimal places.

t = 3, r = 2

x	$P_0\{S \geq x\}$
0	1.000
1	.833
3	.500
4	.167

t = 3, r = 3

x	$P_0\{S \geq x\}$
.000	1.000
.667	.944
2.000	.528
2.667	.361
4.667	.194
6.000	.028

t = 3, r = 4

x	$P_0\{S \geq x\}$
.0	1.000
.5	.931
1.5	.653
2.0	.431
3.5	.273
4.5	.125
6.0	.069
6.5	.042
8.0	.005

t = 3, r = 5

x	$P_0\{S \geq x\}$
.0	1.000
.4	.954
1.2	.691
1.6	.522
2.8	.367
3.6	.182
4.8	.124
5.2	.093
6.4	.039
7.6	.024
8.4	.008
10.0	.001

t = 3, r = 6

x	$P_0\{S \geq x\}$
.000	1.000
.333	.956
1.000	.740
1.333	.570
2.333	.430
3.000	.252
4.000	.184
4.333	.142
5.333	.072
6.333	.052
7.000	.029
8.333	.012
9.000	.008
9.333	.006
10.333	.002
12.000	.000

t = 3, r = 7

x	$P_0\{S \geq x\}$
.000	1.000
.286	.964
.857	.768
1.143	.620
2.000	.486
2.571	.305
3.429	.237
3.714	.192
4.571	.112
5.429	.085
6.000	.051
7.143	.027
7.714	.021
8.000	.016
8.857	.008
10.286	.004
10.571	.003
11.143	.001
12.286	.000

t = 3, r = 8

x	$P_0\{S \geq x\}$
.00	1.000
.25	.967
.75	.794
1.00	.654
1.75	.531
2.25	.355
3.00	.285
3.25	.236
4.00	.149
4.75	.120
5.25	.079
6.25	.047
6.75	.038
7.00	.030
7.75	.018
9.00	.010
9.25	.008
9.75	.005
10.75	.002
12.00	.001
12.25	.001
13.00	.000

t = 3, r = 9

x	$P_0\{S \geq x\}$
.000	1.000
.222	.971
.667	.814
.889	.685
1.556	.569
2.000	.398
2.667	.328
2.889	.278
3.556	.187
4.222	.154
4.667	.107
5.556	.069
6.000	.057
6.222	.048
6.889	.031
8.000	.019

Table A. *(continued)*

t = 3, **r** = 9

x	$P_0\{S \geq x\}$
8.222	.016
8.667	.010
9.556	.006
10.667	.004
10.889	.003
11.556	.001
12.667	.001
13.556	.000

t = 3, **r** = 10

x	$P_0\{S \geq x\}$
.0	1.000
.2	.974
.6	.830
.8	.710
1.4	.601
1.8	.436
2.4	.368
2.6	.316
3.2	.222
3.8	.187
4.2	.135
5.0	.092
5.4	.078
5.6	.066
6.2	.046
7.2	.030
7.4	.026
7.8	.018
8.6	.012
9.6	.007
9.8	.006
10.4	.003
11.4	.002
12.2	.001
12.6	.001
12.8	.001
13.4	.000

t = 3, **r** = 11

x	$P_0\{S \geq x\}$
.000	1.000
.182	.976
.545	.844
.727	.732
1.273	.629
1.636	.470
2.182	.403
2.364	.351
2.909	.256
3.455	.219
3.818	.163
4.545	.116
4.909	.100
5.091	.087
5.636	.062
6.545	.043
6.727	.038
7.091	.027
7.818	.019
8.727	.013
8.909	.011
9.455	.006
10.364	.004
11.091	.003
11.455	.002
11.636	.001
12.182	.001
13.273	.001
13.636	.000

t = 3, **r** = 12

x	$P_0\{S \geq x\}$
.000	1.000
.167	.978
.500	.856
.667	.751
1.167	.654
1.500	.500
2.000	.434
2.167	.383
2.667	.287
3.167	.249
3.500	.191
4.167	.141
4.500	.123
4.667	.108
5.167	.080
6.000	.058
6.167	.051
6.500	.038
7.167	.027
8.000	.020
8.167	.017
8.667	.011
9.500	.007
10.167	.005
10.500	.004
10.667	.003
11.167	.002
12.167	.002
12.500	.001
12.667	.001
13.167	.001
13.500	.000

t = 3, **r** = 13

x	$P_0\{S \geq x\}$
.000	1.000
.154	.980
.462	.866
.615	.767
1.077	.675
1.385	.527
1.846	.463
2.000	.412
2.462	.316
2.923	.278
3.231	.217
3.846	.165
4.154	.145
4.308	.129
4.769	.098
5.538	.073
5.692	.065
6.000	.050
6.615	.037
7.385	.028
7.538	.025
8.000	.016
8.769	.012
9.385	.009
9.692	.007
9.846	.005
10.308	.004
11.231	.003
11.538	.002
11.692	.002
12.154	.001
12.462	.001
12.923	.001
14.000	.001
14.308	.000

t = 4, **r** = 2

x	$P_0\{S \geq x\}$
.0	1.000
.6	.958
1.2	.833

Table A. *(continued)*

t = 4, r = 2

x	$P_0\{S \geqslant x\}$
1.8	.792
2.4	.625
3.0	.542
3.6	.458
4.2	.375
4.8	.208
5.4	.167
6.0	.042

t = 4, r = 3

x	$P_0\{S \geqslant x\}$
.2	1.000
.6	.958
1.0	.910
1.8	.727
2.2	.608
2.6	.524
3.4	.446
3.8	.342
4.2	.300
5.0	.207
5.4	.175
5.8	.148
6.6	.075
7.0	.054
7.4	.033
8.2	.017
9.0	.002

t = 4, r = 4

x	$P_0\{S \geqslant x\}$
.0	1.000
.3	.992
.6	.928
.9	.900
1.2	.800
1.5	.754
1.8	.677

t = 4, r = 4

x	$P_0\{S \geqslant x\}$
2.1	.649
2.4	.524
2.7	.508
3.0	.432
3.3	.389
3.6	.355
3.9	.324
4.5	.242
4.8	.200
5.1	.190
5.4	.158
5.7	.141
6.0	.105
6.3	.094
6.6	.077
6.9	.068
7.2	.054
7.5	.052
7.8	.036
8.1	.033
8.4	.019
8.7	.014
9.3	.012
9.6	.007
9.9	.006
10.2	.003
10.8	.002
11.1	.001
12.0	.000

t = 4, r = 5

x	$P_0\{S \geqslant x\}$
.12	1.000
.36	.975
.60	.944
1.08	.857
1.32	.771
1.56	.709
2.04	.652
2.28	.561
2.52	.521

t = 4, r = 5

x	$P_0\{S \geqslant x\}$
3.00	.445
3.24	.408
3.48	.372
3.96	.298
4.20	.260
4.44	.226
4.92	.210
5.16	.162
5.40	.151
5.88	.123
6.12	.107
6.36	.093
6.84	.075
7.08	.067
7.32	.055
7.80	.044
8.04	.034
8.28	.031
8.76	.023
9.00	.020
9.24	.017
9.72	.012
9.96	.009
10.20	.007
10.68	.005
10.92	.003
11.16	.002
11.64	.002
11.88	.002
12.12	.001
12.60	.001
12.84	.000

t = 4, r = 6

x	$P_0\{S \geqslant x\}$
.0	1.000
.2	.996
.4	.957
.6	.940
.8	.874
1.0	.844
1.2	.789

t = 4, r = 6

x	$P_0\{S \geqslant x\}$
1.4	.772
1.6	.679
1.8	.668
2.0	.609
2.2	.574
2.4	.541
2.6	.512
3.0	.431
3.2	.386
3.4	.375
3.6	.338
3.8	.317
4.0	.270
4.2	.256
4.4	.230
4.6	.218
4.8	.197
5.0	.194
5.2	.163
5.4	.155
5.6	.127
5.8	.114
6.2	.108
6.4	.089
6.6	.088
6.8	.073
7.0	.066
7.2	.060
7.4	.056
7.6	.043
7.8	.041
8.0	.037
8.2	.035
8.4	.032
8.6	.029
8.8	.023
9.0	.022
9.4	.017
9.6	.014
9.8	.013
10.0	.010
10.2	.010
10.4	.009
10.6	.007

Table A. (continued)

t = 4, r = 6

x	$P_0\{S \geq x\}$
10.8	.006
11.0	.006
11.4	.004
11.6	.003
11.8	.003
12.0	.002
12.2	.002
12.6	.001
12.8	.001
13.0	.001
13.2	.001
13.4	.001
13.6	.000

t = 4, r = 7

x	$P_0\{S \geq x\}$
.086	1.000
.257	.984
.429	.963
.771	.906
.943	.845
1.114	.800
1.457	.757
1.629	.685
1.800	.652
2.143	.590
2.314	.557
2.486	.524
2.829	.456
3.000	.418
3.171	.382
3.514	.366
3.686	.310
3.857	.297
4.200	.262
4.371	.239
4.543	.220
4.886	.195
5.057	.180

t = 4, r = 7

x	$P_0\{S \geq x\}$
5.229	.161
5.571	.143
5.743	.122
5.914	.118
6.257	.100
6.429	.093
6.600	.085
6.943	.073
7.114	.063
7.286	.056
7.629	.052
7.800	.041
7.971	.038
8.314	.035
8.486	.033
8.657	.030
9.000	.023
9.171	.020
9.343	.017
9.686	.015
9.857	.013
10.029	.012
10.371	.010
10.543	.009
10.714	.008
11.057	.007
11.229	.005
11.400	.004
11.743	.004
11.914	.003
12.086	.003
12.429	.002
12.600	.002
12.771	.002
13.114	.001
13.286	.001
13.457	.001
13.800	.001
13.971	.001
14.143	.001
14.486	.000

t = 4, r = 8

x	$P_0\{S \geq x\}$
.00	1.000
.15	.998
.30	.971
.45	.959
.60	.912
.75	.890
.90	.849
1.05	.837
1.20	.765
1.35	.757
1.50	.710
1.65	.681
1.80	.654
1.95	.629
2.25	.558
2.40	.517
2.55	.507
2.70	.471
2.85	.450
3.00	.404
3.15	.389
3.30	.362
3.45	.350
3.60	.326
3.75	.323
3.90	.287
4.05	.278
4.20	.242
4.35	.226
4.65	.219
4.80	.193
4.95	.191
5.10	.168
5.25	.158
5.40	.148
5.55	.141
5.70	.121
5.85	.117
6.00	.110
6.15	.106
6.30	.100
6.45	.094

t = 4, r = 8

x	$P_0\{S \geq x\}$
6.60	.081
6.75	.079
7.05	.068
7.20	.060
7.35	.058
7.50	.051
7.65	.049
7.80	.046
7.95	.042
8.10	.038
8.25	.037
8.55	.031
8.70	.028
8.85	.025
9.00	.023
9.15	.022
9.45	.019
9.60	.016
9.75	.015
9.90	.014
10.05	.014
10.20	.011
10.35	.011
10.50	.009
10.65	.009
10.80	.008
10.95	.008
11.10	.006
11.25	.006
11.40	.005
11.55	.005
11.85	.004
12.00	.004
12.15	.004
12.30	.003
12.45	.003
12.60	.002
12.75	.002
12.90	.002
13.05	.002
13.20	.002
13.35	.001
13.50	.001

Table A. (continued)

t = 4, r = 8

x	$P_0\{S \geq x\}$
13.65	.001
13.80	.001
13.95	.001
14.25	.001
14.40	.001
14.55	.001
14.70	.001
14.85	.000

t = 5, r = 3

x	$P_0\{S \geq x\}$
.000	1.000
.267	1.000
.533	.988
.800	.972
1.067	.941
1.333	.914
1.600	.845
1.867	.831
2.133	.768
2.400	.720
2.667	.682
2.933	.649
.200	.595
3.467	.559
3.733	.493
4.000	.475
4.267	.432
4.533	.406
4.800	.347
5.067	.326
5.333	.291
5.600	.253
5.867	.236
6.133	.213
6.400	.172
6.667	.163
6.933	.127
7.200	.117
7.467	.096
7.733	.080

t = 5, r = 3

x	$P_0\{S \geq x\}$
8.000	.063
8.267	.056
8.533	.045
8.800	.038
9.067	.028
9.333	.026
9.600	.017
9.867	.015
10.133	.008
10.400	.005
10.667	.004
10.933	.003
11.467	.001
12.000	.000

t = 5, r = 4

x	$P_0\{S \geq x\}$
.0	1.000
.2	.999
.4	.991
.6	.980
.8	.959
1.0	.940
1.2	.906
1.4	.895
1.6	.850
1.8	.815
2.0	.785
2.2	.759
2.4	.715
2.6	.685
2.8	.630
3.0	.612
3.2	.579
3.4	.552
3.6	.500
3.8	.479
4.0	.442
4.2	.413
4.4	.395
4.6	.370

t = 5, r = 4

x	$P_0\{S \geq x\}$
4.8	.329
5.0	.317
5.2	.286
5.4	.275
5.6	.249
5.8	.227
6.0	.205
6.2	.197
6.4	.178
6.6	.161
6.8	.143
7.0	.136
7.2	.121
7.4	.113
7.6	.095
7.8	.086
8.0	.080
8.2	.072
8.4	.063
8.6	.060
8.8	.049
9.0	.043
9.2	.038
9.4	.035
9.6	.028
9.8	.025
10.0	.021
10.2	.019
10.4	.017
10.6	.014
10.8	.011
11.0	.010
11.2	.008
11.4	.007
11.6	.006
11.8	.005
12.0	.004
12.2	.004
12.4	.003
12.6	.002
12.8	.002
13.0	.001
13.2	.001
13.4	.001

t = 5, r = 4

x	$P_0\{S \geq x\}$
13.6	.001
13.8	.000

t = 5, r = 5

x	$P_0\{S \geq x\}$
.00	1.000
.16	1.000
.32	.994
.48	.986
.64	.972
.80	.958
.96	.932
1.12	.925
1.23	.891
1.44	.865
1.60	.842
1.76	.823
1.92	[illegible]
2.08	.765
2.24	.721
2.40	.707
2.56	.679
2.72	.657
2.88	.613
3.04	.594
3.20	.562
3.36	.535
3.52	.518
3.68	.494
3.84	.454
4.00	.443
4.16	.410
4.32	.398
4.48	.371
4.64	.349
4.80	.325
4.96	.316
5.12	.295
5.28	.275
5.44	.255
5.60	.246

Table A.* (continued)

t = 5, r = 5		t = 5, r = 5		t = 5, r = 5		t = 5, r = 5	
x	$P_0\{S > x\}$	x	$P_0\{S > x\}$	x	$P_0\{S > x\}$	x	$P_0\{S > x\}$
5.76	.227	8.16	.077	10.56	.019	12.96	.003
5.92	.218	8.32	.073	10.72	.018	13.12	.003
6.08	.195	8.48	.066	10.88	.015	13.28	.003
6.24	.183	8.64	.058	11.04	.013	13.44	.002
6.40	.174	8.80	.056	11.20	.012	13.60	.002
6.56	.164	8.96	.049	11.36	.012	13.76	.002
6.72	.151	9.12	.046	11.52	.010	13.92	.002
6.88	.146	9.28	.042	11.68	.009	14.08	.001
7.04	.130	9.44	.038	11.84	.008	14.24	.001
7.20	.121	9.60	.035	12.00	.007	14.40	.001
7.36	.112	9.76	.032	12.16	.006	14.56	.001
7.52	.107	9.92	.029	12.32	.006	14.72	.001
7.68	.094	10.08	.026	12.48	.005	14.88	.001
7.84	.089	10.24	.024	12.64	.004	15.04	.000
8.00	.082	10.40	.022	12.80	.004		

Computed by G. A. Mack on the Ohio State University IBM 370/165.

* Myles Hollander and Douglas A. Wolfe, Nonparametric Statistical Methods (New York : John Wiley & Sons, 1973), pp. 361:371

ประวัติผู้เขียน

เรือเอกหญิง ศิริลักษณ์ ตีไพบูลย์ เกิดเมื่อวันที่ 31 สิงหาคม พ.ศ. 2493 ที่จังหวัดพิษณุโลก สำเร็จปริญญาศิลปศาสตร์บัณฑิต (สถิติ) จากมหาวิทยาลัยธรรมศาสตร์ เมื่อปีการศึกษา 2516 และได้เข้าศึกษาต่อที่บัณฑิตวิทยาลัย ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2520 ปัจจุบันทำงานในตำแหน่งหัวหน้าแผนกสถิติ กองตรวจสอบและวิเคราะห์ สำนักงานปลัดบัญชีทหารเรือ