รายการอ้างอิง

ภาษาไทย

ธีระพร วีระถาวร, ดร. การอนุมานเชิงสถิติขั้นกลาง:โครงสร้างและความหมาย. กรุงเทพมหานคร : พิทักษ์การพิมพ์ , 2531.

ประชุม สุวัตถี. ทฤษฎีการอนุมานเชิงสถิติ. กรุงเทพมหานคร : สำนักพิมพ์จุฬาลงกรณ์ , 2527.

ภาษาอังกฤษ

Banks, J. and Carson, J.S. Discrete-Event System Simulation. New Jersey : Prentice-hall ,1984.

Boos, Dennis D. Minimum distance estimators for location and goodness of fit. Journal of the American Statistical Association 76 (1981) :663-670.

Hogg, Robert V. and Craig, A.T. Introduction to Mathematical Statistics. 4th ed. New York : Macmiiian, 1978.

_____. and Klugman, Stuart A. Loss Distributions. New York : John wiley & Son , 1984.

Ingram Olkin, Leon J. Gleser and Cyrus Derman. Probability Models and Applications. 2nd ed. New York : Macmiiian, 1994.

Law, Averill M. and Kelton, W. David. Simulation Modeling & Analysis. 2nd ed. New York : McGraw-Hill Book Co., 1991.

Parr,William C. and Schucany, William R. Minimum distance and robust estimation. Journal of the American Statistical Association 75 (1980) : 616-624

Rao, C.R. Theory of the method of estimation by minimum chi-square. Bull. Interat. Statist. Inst. 35 (1957) : 25-32.

ภาคผนวก

ภาคผนวก ก

การสร้างตัวเลขสุ่ม (Random Number)

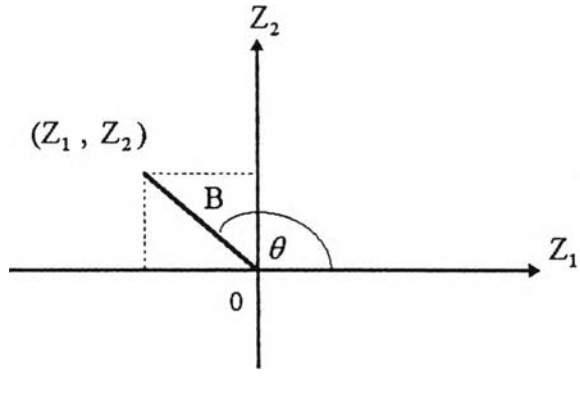
การสร้างลักษณะการแจกแจงแบบต่าง ๆ นั้น จะต้องอาศัยตัวเลขสุ่มเป็นพื้นฐานในการสร้าง สำหรับการวิจัยครั้งนี้ใช้วิธีการสร้างตัวเลขสุ่มตามวิธีของ White and Schmidt (1975) ซึ่งขั้นตอนในการสร้างจะแสดงรายละเอียดด้วยฟังก์ชันต่อไปนี้

```
FUNCTION RAND(IU)
IU = IU*16807
IF (IU.LT.0) IU = IU + 2147483647+1
RAND = IU
RAND = RAND*0.465661E-9
RETURN
END
```

ค่า IU จะเป็นค่า SEED หรือค่าเริ่มต้น ซึ่งจะต้องเป็นจำนวนเต็มบวกที่เป็นเลขคี่ RAND จะเป็นค่าของตัวเลขสุ่มที่มีค่าอยู่ระหว่าง 0 ถึง 1

การสร้างการแจกแจงแบบปกติ $N(\mu,\sigma^2)$

การผลิตเลขสุ่มที่มีการแจกแจงแบบปกติใช้วิธีของ Box และ Muller (1958) โดยผลิตเลขสุ่มที่มีการแจกแจงแบบปกติมาตรฐาน N(0,1) พร้อมกัน 2 ค่า และแต่ละค่าเป็นอิสระต่อกัน โดยใช้ตัวผลิต (Generator) Z_1 และ Z_2 พิจารณาดังรูปต่อไปนี้



พิจารณาจากรูปจะได้

$$Z_1 = B\cos(\theta) \quad (1)$$

$$Z_2 = B\sin(\theta) \quad (2)$$

เนื่องจาก $B = Z_1^2 + Z_2^2$ มีการแจกแจงแบบไคสแควร์ด้วยระดับความเป็นอิสระ 2 และเทียบเท่าการแจกแจงแบบเอกซ์โพเนนเชียล มีค่าเฉลี่ยเท่ากับ 2 โดยวิธีแปลงผกผัน (Inverse Transformations) สามารถสร้างเลขสุ่มที่มีการแจกแจงแบบเอกซ์โพเนนเชียลได้ดังนี้

$$B = (-2\ln R)^{1/2} \quad (3)$$

เมื่อ R เป็นเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอในช่วง (0,1)

จากการสมมาตรของการแจกแจงแบบปกติ (Normal Distribution) จะได้ว่ามุม θ มีการแจกแจงแบบสม่ำเสมอ ระหว่าง 0 ถึง 2π เรเดียน และมีรัศมี B กับมุม θ เป็นอิสระต่อกันจากสมการ (1) (2) และ(3) เราสามารถสร้างเลขสุ่มที่มีการแจกแจงแบบปกติมาตรฐานจากตัวเลขสุ่ม 2 ชุด R_1 และ R_2 กล่าวคือ

$$Z_1 = (-2\ln R_1)^{1/2}\cos(2\pi R_2)$$

$$Z_2 = (-2\ln R_1)^{1/2}\sin(2\pi R_2)$$

ซึ่ง R_1 และ R_2 เป็นตัวเลขสุ่มที่สร้างจากฟังก์ชัน FUNCTION RAND(IU) เมื่อได้ตัวเลขสุ่มที่มีการแจกแจงแบบปกติมาตรฐานแล้ว จะทำการแปลงตัวเลขสุ่มดังกล่าวโดยอาศัยฟังก์ชัน

$$EX_1 = \mu + \sigma Z_1$$

$$EX_2 = \mu + \sigma Z_2$$

ซึ่งจะได้ว่า EX_1 และ EX_2 มีการแจกแจงแบบปกติด้วยค่าเฉลี่ยเท่ากับ μ ค่าความแปรปรวนเท่ากับ σ^2 ($EX_i \sim N(\mu, \sigma^2)$; i = 1 ,2) โดยรายละเอียดโปรแกรมย่อยแสดงได้ดังนี้

```
      SUBROUTINE RNOR(EX1)
      COMMON/SEED/IU,KK
     *          /PAR/RMEW,SIGMA
      PI = 3.1415926
      IF (KK.EQ.1) GOTO 100
      RONE = RAND(IU)
      RTWO = RAND(IU)
      ZONE = SQRT(-2*ALOG(RONE))*COS(2*PI*RTWO)
      ZTWO = SQRT(-2*ALOG(RONE))*SIN(2*PI*RTWO)
      EX1 = ZONE*SIGMA+RMEW
      KK = 1
      GOTO 200
  100 EX1 = ZTWO*SIGMA+RMEW
      KK = 0
  200 RETURN
      END
```

การสร้างการแจกแจงแบบลอกนอร์มอล

การสร้างตัวแปรสุ่มที่มีการแจกแจงแบบลอกนอร์มอล จะอาศัยตัวแปรสุ่มที่มีการแจกแจงแบบปกติ จากความสัมพันธ์ดังนี้

$$Y \sim N(\mu, \sigma^2)$$
$$Y = \ln X$$
$$X \sim LN(\mu, \sigma^2)$$
$$X = \exp(Y)$$

ขั้นตอนในการสร้างตัวแปรสุ่มที่มีการแจกแจงแบบลอกนอร์มอลมีดังนี้

ขั้นที่ 1 สร้างตัวแปรสุ่มที่มีการแจกแจงแบบปกติ

ขั้นที่ 2 หาค่า $x = \exp(y)$

การสร้างการแจกแจงแบบไวบูลล์

ฟังก์ชันความหนาแน่นและฟังก์ชันการแจกแจง แสดงได้ดังนี้

$$f(x) = c\tau x^{\tau-1}\exp\left(-cx^{\tau}\right) \quad , \quad x>0$$

$$F(x) = 1-\exp\left(-cx^{\tau}\right) \quad , \quad x>0$$

การสร้างตัวแปรสุ่มให้มีการแจกแจงแบบไวบูลล์ อาศัยเทคนิคการแปลงผกผัน โดยมีขั้นตอนการสร้างดังนี้

ขั้นที่ 1 ให้ $F(x) = U$ โดยที่ U คือตัวเลขสุ่มแบบสม่ำเสมอช่วง (0,1)

ขั้นที่ 2 หาค่า x ในเทอมของ U ได้เป็น

$$x = \left[-\frac{\ln(1-U)}{c}\right]^{\frac{1}{\tau}}$$

ถ้าต้องการสร้างข้อมูลถูกตัดปลายทางซ้ายก็จะเพิ่มเงื่อนไขว่าต้องการให้ตัวแปรสุ่มมีค่ามากกว่าค่าตัดปลายทางซ้าย ดังนั้นคำสั่งในการสร้างตัวแปรให้มีการแจกแจงแบบไวบูลล์ คือ

```
      DO 10 I = 1,N
5     U = RAND(IU)
      UU1 = -1*ALOG(1-U)/XC1
      U1 = UU1**(1/XT1)
      IF (U1.LE.XD) GOTO 5
      X(I) = U1
10    CONTINUE
```

การสร้างการแจกแจงแบบพาร์โต

ฟังก์ชันความหนาแน่นและฟังก์ชันการแจกแจง แสดงได้ดังนี้

$$f(x) = \frac{\alpha\lambda^{\alpha}}{(\lambda+x)^{\alpha+1}} \quad , \quad x>0$$

$$F(x) = 1-\left(\frac{\lambda}{\lambda+x}\right)^{\alpha} \quad , \quad x>0$$

การสร้างตัวแปรสุ่มให้มีการแจกแจงแบบพาร์โต อาศัยเทคนิคการแปลงผกผัน โดยมีขั้นตอนการสร้างดังนี้

ขั้นที่ 1 ให้ $F(x) = U$ โดยที่ U คือตัวเลขสุ่มแบบสม่ำเสมอช่วง (0,1)

ขั้นที่ 2 หาค่า x ในเทอมของ U ได้เป็น

$$x = \frac{\lambda}{\sqrt[\alpha]{1-R}} - \lambda$$

ถ้าต้องการสร้างข้อมูลถูกตัดปลายทางซ้ายก็จะเพิ่มเงื่อนไขว่าต้องการให้ตัวแปรสุ่มมีค่ามากกว่าค่าตัดปลายทางซ้าย ดังนั้นคำสั่งในการสร้างตัวแปรให้มีการแจกแจงแบบพาร์โต คือ

```
      DO 10 I = 1,N
5     U = RAND(IU)
      UU1 = (1-U)**(1/XA1)
      U1 = (XL1/UU1)-XL1
      IF (U1.LE.XD) GOTO 5
      X(I) = U1
10    CONTINUE
```

การสร้างฟังก์ชันเพื่อคำนวณพื้นที่ใต้โค้งปกติมาตรฐาน

ในกรณีนี้จะสร้างจากคะแนนดิบที่ปรับเป็นคะแนนมาตรฐาน (Z) และนั่นคือค่า ZZ เทียบได้กับค่า Z คำนวณได้จากสูตร $Z = (X-\mu)/\sigma$ ซึ่งขั้นตอนในการสร้างจะแสดงรายละเอียดด้วยฟังก์ชันดังนี้

```
      FUNCTION PHIB(ZZ)
      IF ((ZZ.GT.7.).OR.(ZZ.LT.-7.)) THEN
         P = 1.
      ELSE
        R = EXP(-ZZ*ZZ/2.)/2.5066282746
        WW = 1./(1.+.33267*ABS(ZZ))
        P = 1.-R*(.4361836*WW-.1201676*(WW**2)+.937298*(WW**3))
      END IF
```

```
      IF (ZZ.GE.0.) THEN
        PHIB = P
      ELSE
        PHIB = 1.-P
      END IF
   33 RETURN
      END
```

การสร้างฟังก์ชันแกมมา ($\Gamma(\alpha)$)

```
      FUNCTION GAMF(ALP)
      YY1 = 1/(12*ALP)-1/(360*(ALP**3))-ALP
      YY2 = ALP**ALP
      YY3 = SQRT(2*22/(7*ALP))
      GAMF = YY3*YY2*EXP(YY1)
      RETURN
      END
```

ภาคผนวก ข

```
C*****************************
C         MAIN   PROGRAM
C*****************************
      DIMENTION  SUMP1(3),SUMP2(3)
      COMMON/SEED/IU,KK
     *      /PAR1/RMEW,SIGMA,XMW0,XSG0
     *      /PAR2/XC1,XT1,XC0,XT0
     *      /PAR3/XA1,XL1,XA0,XL0
     *      /SON/X(1000),XBAR(200)
     *      /SAM/C(0:200),FQ(0:200),F(0:200),EFQ(0:200)
     *      /SIR/SEP1(3),SEP2(3)
     *      /SAY/P1(3),P2(3),RELP1(3),RELP2(3),RELME(3),
     *            SMSEP1(3),SMSEP2(3),SMSEME(3),
     *            VARP1(3),VARP2(3),VARME(3)
     *      /PAR4/PP1(3,1000),PP2(3,1000)
     *      /PAR5/XD,N,K
     *      /SIT/DIVG,NEG
     *      /SET/N1,N2
      CHARACTER DIVG*1,NEG*1
      N = 100
      XD = 0.5
      SIGMA = 1.
      RMEW = 1.
      XC1 = 0.1
      XT1 = 1.
      XA1 = 7.
      XL1 = 10.
      XN = N
      K = SQRT(XN)
      IU = 53
      IR = 1000
      ICOUNT = 0
      KK = 0
C---------------------------------------
```

```
      RP1 = RMEW
      RP2 = SIGMA
C     RP1 = XC1
C     RP2 = XT1
C     RP1 = XA1
C     RP2 = XL1
      DO 44 JJ = 1,3
       SUMP1(JJ) = 0.
       SUMP2(JJ) = 0.
       SMSEP1(JJ) = 0.
       SMSEP2(JJ) = 0..
 44 CONTINUE
C---------------------------
      DO 500 IZ = 1,IR
 111    K = SQRT(XN)
C---------- GEN DATA OF LOGNORMAL DIST. ----------
      DO 10 I = 1,N
   5    W1 = RAND(IU)
        CALL RNOR(EX1)
        U1 = EXP(EX1)
        IF (U1.LE.XD) GOTO 5
        X(I) = U1
  10  CONTINUE
C--------- GEN DATA OF WEIBULL DIST. ---------
C       DO 10 I = 1,N
C   5   U = RAND(IU)
C       UU1 = -1*ALOG(1-U)/XC1
C       U1 = UU1**(1/XT1)
C       IF (U1.LE.XD) GOTO 5
C         X(I) = U1
C  10   CONTINUE
C---------- GEN DATA OF PARETO DIST. ----------
C       DO 10 I = 1,N
C   5     U = RAND(IU)
C         UU1 = (1-U)**(1/XA1)
C         U1 = (XL1/UU1)-XL1
```

```
C          IF (U1.LE.XD) GOTO 5
C            X(I) = U1
C   10   CONTINUE
      CALL SSORT
      CALL GROUP
      CALL RMOM
      IF (XSG0.LE.0) GOTO 111
C     IF ((XA0.LE.0).OR.(XL0.LE.0)) GOTO 111
C     N1 = 1
C     N2 = 1
C   11 CALL PM
C     IF (N1.GT.K) GOTO 111
C     IF ((XT0.LE.0.).OR.(XC0.LE.0.)) GOTO 11
      MAX = 30
      DIVG = 'N'
      NEG = 'N'
      CALL RMLE(MAX,ZL1,ZL2)
      IF ((NEG.EQ.'Y').OR.(DIVG.EQ.'Y')) GOTO 111
      CALL RMCE(MAX,ZC1,ZC2)
      IF ((NEG.EQ.'Y').OR.(DIVG.EQ.'Y')) GOTO 111
      CALL RMDE(MAX,ZD1,ZD2)
      IF ((NEG.EQ.'Y').OR.(DIVG.EQ.'Y')) GOTO 111
C-------------------------------------------------
      PP1(1,IZ) = ZL1
      PP1(2,IZ) = ZC1
      PP1(3,IZ) = ZD1
      PP2(1,IZ) = ZL2
      PP2(2,IZ) = ZC2
      PP2(3,IZ) = ZD2
      SUMP1(1) = ZL1 + SUMP1(1)
      SUMP1(2) = ZC1 + SUMP1(2)
      SUMP1(3) = ZD1 + SUMP1(3)
      SUMP2(1) = ZL2 + SUMP2(1)
      SUMP2(2) = ZC2 + SUMP2(2)
      SUMP2(3) = ZD2 + SUMP2(3)
      SMSEP1(JJ) = SEP1(JJ) + SMSEP1(JJ)
```

```
      SMSEP2(JJ) = SEP2(JJ) + SMSEP2(JJ)
C-------------------------------------------------
C   AVERAGE MEW AND SIGMA AND
C   RELATIVE ERROR AND VARIANCE AND
C   MEAN SQUARE ERROR
C-------------------------------------------------
    DO 77 JJ = 1,3
     P1(JJ) = SUMP1(JJ)/IZ
     P2(JJ) = SUMP2(JJ)/IZ
77  CONTINUE
       DO 22 I = 1,3
        SVP1 = 0.
        SVP2 = 0.
        DO 23 J = 1,IZ
         SVP1 = SVP1+((PP1(I,J)-P1(I))**2)
         SVP2 = SVP2+((PP2(I,J)-P2(I))**2)
23       CONTINUE
        VARP1(I) = SVP1/IZ
        VARP2(I) = SVP2/IZ
22     CONTINUE
       DO 222 IQ = 1,3
        RELP1(IQ) = ABS(P1(IQ)-RP1)*100/RP1
        RELP2(IQ) = ABS(P2(IQ)-RP2)*100/RP2
        RELME(IQ) = (RELP1(IQ)+RELP2(IQ))/2
        SMSEME(IQ) = (SMSEP1(IQ)+SMSEP2(IQ))/2
        VARME(IQ) = (VARP1(IQ)+VARP2(IQ))/2
222    CONTINUE
C------------ WRITE OUTPUT ----------------
       WRITE (6,238)RP1,RP2,XD,K,N
238    FORMAT (4X,'MEW =',F10.7,4X,'SIGMA =',F10.7,2X,'D',F6.2,' K',I4,' N',I4)
       WRITE (6,208) P1(1),P1(2),P1(3)
208    FORMAT (3X,'MEW',8X,3(8X,F10.7))
       WRITE (6,181)SMSEP1(1),SMSEP1(2),SMSEP1(3)
181    FORMAT (3X,'MSE-MEW = ',1X,3(3X,F15.7)/75('-'))
       WRITE (6,212) P2(1),P2(2),P2(3)
212    FORMAT (3X,'SIGMA',6X,3(8X,F10.7))
```

```
      WRITE (6,118) SMSEP2(1),SMSEP2(2),SMSEP2(3)
118   FORMAT (3X,'MSE-SIGMA = ',3(2X,F15.7,1X)/75('-'))
      WRITE (6,128) SMSEME(1),SMSEME(2),SMSEME(3)
128   FORMAT (3X,'AVG-MSE = ',1X,3(3X,F15.7)/75('=')/)
     END IF
500  CONTINUE
100 STOP
    END
C*************************************
C    RANDOM DATA
C*************************************
    FUNCTION RAND(IU)
    IU = IU*16807
    IF (IU.LT.0) IU = IU + 2147483647+1
    RAND = IU
    RAND = RAND*0.465661E-9
    RETURN
    END
C*********************************
C    NORMAL DISTRIBUTION
C*********************************
    SUBROUTINE RNOR(EX1)
    COMMON/SEED/IU,KK
   *     /PAR1/RMEW,SIGMA,XMW0,XSG0
    PI = 3.1415926
    IF (KK.EQ.1) GOTO 100
    RONE = RAND(IU)
    RTWO = RAND(IU)
    ZONE = SQRT(-2*ALOG(RONE))*COS(2*PI*RTWO)
    ZTWO = SQRT(-2*ALOG(RONE))*SIN(2*PI*RTWO)
    EX1 = ZONE*SIGMA+RMEW
    KK = 1
    GOTO 200
100  EX1 = ZTWO*SIGMA+RMEW
    KK = 0
200  RETURN
```

```
      END
C*************************************************
C     LITTLE PHI ( (LNC(I)-MEW)/SIGMA = Y )
C     PDF. OF Y  WHEN Y IS STANDARD NORMAL
C*************************************************
      FUNCTION PHIL(Y)
      PI = 3.1415926
      IF((Y.GT.7.).OR.(Y.LT.-7.)) THEN
        PHIL = 0.
      ELSE
        XX = -1.*(Y**2)/4.
        PHIL1 = EXP(XX)/SQRT(2*PI)
        PHIL = EXP(XX)*PHIL1
      END IF
      RETURN
      END
C*************************************************
C     BIG PHI  ( (LNC(I)-MEW)/SIGMA = Y )
C     CDF. OF Y  WHEN Y IS STANDARD NORMAL
C               AND Y IS UPPER LIMIT
C*************************************************
      FUNCTION PHIB(ZZ)
      IF ((ZZ.GT.7.).OR.(ZZ.LT.-7.)) THEN
        P = 1.
      ELSE
        R = EXP(-ZZ*ZZ/2.)/2.5066282746
        WW = 1./(1.+.33267*ABS(ZZ))
        P = 1.-R*(.4361836*WW-.1201676*(WW**2)+.937298*(WW**3))
      END IF
      IF (ZZ.GE.0.) THEN
        PHIB = P
      ELSE
        PHIB = 1.-P
      END IF
33 RETURN
```

```
C************************************************
C     GAMMA FUNCTION AT ALPHA (1+1/TOWN)
C************************************************
      FUNCTION GAMF(ALP)
      YY1 = 1/(12*ALP)-1/(360*(ALP**3))-ALP
      YY2 = ALP**ALP
      YY3 = SQRT(2*22/(7*ALP))
      GAMF = YY3*YY2*EXP(YY1)
      RETURN
      END
C************************************************
C     CUMULATIVE DISTRIBUTION FUNCTION
C     OF LOGNORMAL DIST.
C************************************************
      FUNCTION CDF(XD,XMW2,XSG2,CCI)
      XX1 = (ALOG(CCI)-XMW2)/XSG2
      XX2 = (ALOG(XD)-XMW2)/XSG2
      XX3 = 1-PHIB(XX2)
      IF ((XX2.GT.7).OR.(XX3.EQ.0)) THEN
        CDF = 0.
      ELSE
        CDF = (PHIB(XX1)-PHIB(XX2))/(1-PHIB(XX2))
      END IF
      RETURN
      END
C************************************************
C     CUMULATIVE DISTRIBUTION FUNCTION
C     OF PARETO DIST.
C************************************************
      FUNCTION CDF(XD,XA2,XL2,CCI)
      XX = (XL2+XD)/(XL2+CCI)
      CDF= 1-(XX**XA2)
      RETURN
      END
```

```
C*******************************
C       SORT DATA
C*******************************
      SUBROUTINE SSORT
      COMMON/SON/X(1000),XBAR(200)
     *           /PAR4/XD,N,K
      IA = N-1
      DO 20 I = 1,IA
        IB = I+1
        DO 20 J = IB,N
          IF (X(I) .GT. X(J)) THEN
            TEMP = X(I)
            X(I) = X(J)
            X(J) = TEMP
          END IF
20    CONTINUE
      RETURN
      END
C*******************************
C     GROUPED DATA
C*******************************
      SUBROUTINE GROUP
      COMMON/SON/X(1000),XBAR(200)
     *      /SAM/C(0:200),FQ(0:200),F(0:200),EFQ(0:200)
     *      /PAR1/RMEW,SIGMA,XMW0,XSG0
     *      /PAR2/XC1,XT1,XC0,XT0
     *      /PAR3/XA1,XL1,XA0,XL0
     *      /PAR4/XD,N,K
      XLEN = (X(N)+0.001-XD)/K
      C(0) = XD
      DO 50 J = 1,K
50    C(J) = XD+J*XLEN
      DO 77 I = 1,K
      J = I-1
      CCI = C(I)
      CCJ = C(J)
```

```
      FCI = CDF(XD,RMEW,SIGMA,CCI)
      FCJ = CDF(XD,RMEW,SIGMA,CCJ)
C     FCI = CDF(XD,XC1,XT1,CCI)
C     FCJ = CDF(XD,XC1,XT1,CCJ)
C     FCI = CDF(XD,XA1,XL1,CCI)
C     FCJ = CDF(XD,XA1,XL1,CCJ)
      EFQ(I) = N*(FCI-FCJ)
77 CONTINUE
      NUM = 1
      FQ(0) = 0.
      DO 30 J = 1,K
        FQ(J) = 0.
        DO 35 I = NUM,N
          IF (X(I).LE.C(J)) THEN
            FQ(J) = FQ(J)+1
          ELSE
            GOTO 40
          END IF
35      CONTINUE
40      NUM = I
30    CONTINUE
C-----------------------
      J = 0
      I = 0
79    J = J+1
      I = I+1
      IF (I.GT.K) GOTO 82
      EFQ(J) = EFQ(I)
      C(J) = C(I)
78    EFQJ = EFQ(J)
      IF (EFQJ.GE.5) THEN
        GOTO 79
      ELSE
        I = I+1
        IF (I.GT.K) GOTO 81
        EFQ(J) = EFQ(J)+EFQ(I)
```

```
        C(J) = C(I)
        GOTO 78
      END IF
   81 K = J
      EFQJ = EFQ(J)
      IF (EFQJ.LT.5) THEN
        K = J-1
        EFQ(K) = EFQ(K) + EFQ(J)
        C(K) = C(J)
        J = J-1
        GOTO 81
      ELSE
        GOTO 83
      END IF
   82 K = J-1
   83 NUM = 1
      FQ(0) = 0.
      DO 60 J = 1,K
        FQ(J) = 0.
        DO 65 I = NUM,N
          IF (X(I).LE.C(J)) THEN
            FQ(J) = FQ(J)+1
          ELSE
            GOTO 70
          END IF
   65   CONTINUE
   70   NUM = I
   60 CONTINUE
      IK = 1
      DO 32 J = 1,K
        SUMX = 0
   31   IJ = IK
        IF (IJ.GT.N) GOTO 54
        IF (X(IJ).LE.C(J)) THEN
          SUMX = SUMX+X(IJ)
          IK = IJ+1
```

```
        GOTO 31
       END IF
54     FQJ = FQ(J)
       IF (FQJ.EQ.0) THEN
        XBAR(J) = 0.
       ELSE
        XBAR(J) = SUMX/FQ(J)
       END IF
32   CONTINUE
     F(0) = 0.
     DO 55 I = 1,K
      J = I-1
      F(I) = FQ(I)/N+F(J)
55   CONTINUE
     RETURN
     END
C*************************************************
C     INITIAL VALUE BY METHOD OF MOMENT
C     OF LOGNORMAL DIST.
C*************************************************
     SUBROUTINE RMOM
     COMMON/SAM/C(0:200),FQ(0:200),F(0:200),EFQ(0:200)
    *      /SON/X(1000),XBAR(200)
    *      /PAR1/RMEW,SIGMA,XMW0,XSG0
    *      /PAR4/XD,N,K
     SUMX = 0.
     SUMX2 = 0.
     DO 601 I = 1,K
       SUMX = SUMX+(FQ(I)*XBAR(I))
       SUMX2 = SUMX2+(FQ(I)*((XBAR(I))**2))
601  CONTINUE
     EX = SUMX/N
     EX2 = SUMX2/N
     XSG0 = ALOG(EX2)-2*ALOG(EX)
     XMW0 = ALOG(EX)-(XSG0**2)/2
     RETURN
```

```
      END
C*********************************
C     PERCENTILE MATCHING
C     OF WEIBULL DIST.
C*********************************
      SUBROUTINE PM
      DOUBLE PRECISION XC1,XT1,XC0,XT0
      COMMON/SEED/IU,KK
     *     /SON/X(1000),XBAR(200)
     *     /SAM/C(0:200),FQ(0:200),F(0:200),EFQ(0:200)
     *     /PAR2/XC1,XT1,XC0,XT0
     *     /PAR4/XD,N,K
     *     /SET/N1,N2
      IF (N2.LE.K) GOTO 55
54    N1 = N1+1
55    N2 = N2+1
      IF (N1.GT.K) GOTO 305
      IF (N2.GT.K) THEN
        N2 = N1+1
        GOTO 54
      END IF
      TT1 = 1-F(N1)
      TT2 = 1-F(N2)
      IF ((TT1.EQ.0.).OR.(TT2.EQ.0.)) GOTO 55
      T1 = ALOG(TT1)
      T2 = ALOG(TT2)
      IF (T1.EQ.T2) GOTO 55
      TT3 = C(N1)/C(N2)
      IF ((T1.EQ.0.).OR.(T2.EQ.0.).OR.(TT3.EQ.0.)) GOTO 55
      T3 = ALOG(TT3)
      T4 = ALOG(T1/T2)
      IF (T3.EQ.0.) GOTO 55
      XT0 = T4/T3
      C1 = C(N1)**(XT0)
      XC0 = -1*T1/C1
      IF ((XT0.LT.0).OR.(XC0.LT.0)) GOTO 55
```

```
305  RETURN
     END
C*****************************************
C     INITIAL VALUE BY METHOD OF MOMENT
C      OF PARETO DIST.
C*****************************************
     SUBROUTINE RMOM
     COMMON/SON/X(1000),XBAR(200)
    *      /SAM/C(0:200),FQ(0:200),F(0:200),EFQ(0:200)
    *      /PAR3/XA1,XL1,XA0,XL0
    *      /PAR4/XD,N,K
     SUMX = 0.
     SUMX2 = 0.
     DO 601 I = 1,K
       SUMX = SUMX+(FQ(I)*XBAR(I))
       SUMX2 = SUMX2+(FQ(I)*((XBAR(I))**2))
601  CONTINUE
     EX = SUMX/N
     EX2 = SUMX2/N
C     XA0 = 2*(EX**2-EX2)/(2*(EX**2)-EX2)
C     XL0 = (XA0-1)*EX
     AA1 = 2*((EX**2)-EX2)
     AA2 = 2*(EX**2)-EX2-2*EX*XD+(XD**2)
     XA0 = AA1/AA2
     XL0 =(EX-XD)*(XA0-1)-XD
     RETURN
     END
C*****************************************
C       DIFF PPP  OF   LOGNORMAL DIST.
C*****************************************
     SUBROUTINE DIFP(I,XMW2,XSG2,P,D1PM,D1PS,D2PM,D2PSM,D2PS)
     DOUBLE PRECISION P,XP
     COMMON/SON/X(1000),XBAR(200)
    *      /SAM/C(0:200),FQ(0:200),F(0:200),EFQ(0:200)
    *      /PAR4/XD,N,K
    *      /SIT/DIVG,NEG
```

```
CHARACTER DIVG*1,NEG*1
J = I-1
CCI = C(I)
CCJ = C(J)
XX2 = ALOG(CCI)-XMW2
X2 = XX2/XSG2
XX1 = ALOG(CCJ)-XMW2
X1 = XX1/XSG2
XX  = (ALOG(XD)-XMW2)/XSG2
IF (XX.GT.7.) THEN
  XP = 0.
  DIVG = 'Y'
  GOTO 22
ELSE
  IF  (XX.LT.-7.) THEN
    XP = 1.
  ELSE
    XP = 1-PHIB(XX)
  END IF
END IF
IF (XP.EQ.0.) THEN
  DIVG = 'Y'
  GOTO 22
ELSE
  P = (PHIB(X2)-PHIB(X1))/XP
END IF
IF (P.EQ.0.) THEN
  DIVG = 'Y'
  GOTO 22
END IF
XDP = PHIL(X1)-PHIL(X2)-P*PHIL(XX)
    D1PM = XDP/(XP*XSG2)
XRP = X1*PHIL(X1)-X2*PHIL(X2)-P*XX*PHIL(XX)
    D1PS = XRP/(XP*XSG2)
XAP = (XRP/(XSG2**2))-(D1PM*PHIL(XX)/XSG2)
XAPP = XAP/XP
```

```
      XBP = PHIL(XX)/((XSG2*XP)**2)
          D2PM = XAPP-XBP*XDP
      XCP1 = PHIL(X1)*((X1**2)-1)/(XSG2**2)
      XCP2 = PHIL(X2)*((X2**2)-1)/(XSG2**2)
      XCP3 = P*PHIL(XX)*((XX**2)-1)/(XSG2**2)
      XCP4 = PHIL(XX)*D1PS/XSG2
      XCP = (XCP1-XCP2-XCP3-XCP4)
          D2PSM = (XCP-XX*XBP*XDP*XP)/XP
      XEP1 = X1*PHIL(X1)*((X1**2)-2)/(XSG2**2)
      XEP2 = X2*PHIL(X2)*((X2**2)-2)/(XSG2**2)
      XEP3 = P*XX*PHIL(XX)*((XX**2)-2)/(XSG2**2)
      XEP4 = XX*PHIL(XX)*D1PS/XSG2
      XEP = (XEP1-XEP2-XEP3-XEP4)
          D2PS = (XEP-XX*XBP*XRP*XP)/XP
   22 RETURN
      END
C*****************************************
C       DIFF FFF   OF  LOGNORMAL DIST.
C*****************************************
      SUBROUTINE DIFF(I,XMW2,XSG2,FF,D1FM,D1FS,D2FM,D2FSM,D2FS)
      DOUBLE PRECISION XP
      COMMON/SON/X(1000),XBAR(200)
     *      /SAM/C(0:200),FQ(0:200),F(0:200),EFQ(0:200)
     *      /PAR4/XD,N,K
     *      /SIT/DIVG,NEG
      CHARACTER DIVG*1,NEG*1
      CCI = C(I)
          FF = F(I)-CDF(XD,XMW2,XSG2,CCI)
      XX2 = ALOG(CCI)-XMW2
      X2 = XX2/XSG2
      XX  = (ALOG(XD)-XMW2)/XSG2
      D1XM = -1*PHIL(X2)/XSG2
      D1XS = -1*X2*PHIL(X2)/XSG2
      D2XM = -1*X2*PHIL(X2)/(XSG2**2)
      D2XSM = PHIL(X2)*(1-(X2**2))/(XSG2**2)
      D2XS = X2*PHIL(X2)*(2-(X2**2))/(XSG2**2)
```

```
      D1DM = -1*PHIL(XX)/XSG2
      D1DS = -1*X1*PHIL(XX)/XSG2
      D2DM = -1*X1*PHIL(XX)/(XSG2**2)
      D2DSM = PHIL(XX)*(1-(XX**2))/(XSG2**2)
      D2DS = XX*PHIL(XX)*(2-(XX**2))/(XSG2**2)
C-----------------------------------------------
      IF (XX.GT.7.) THEN
       XP = 0.
       DIVG = 'Y'
       GOTO 56
      ELSE
       IF  (XX.LT.-7.) THEN
         XP = 1.
       ELSE
         XP = 1-PHIB(XX)
       END IF
      END IF
      IF (XP.EQ.0.) THEN
       DIVG = 'Y'
       GOTO 56
      END IF
      XP2 = 1-PHIB(X2)
      XAP = XP*D1XM-XP2*D1DM
      XBP = XP*D1XS-XP2*D1DS
          D1FM = XAP/(XP**2)
          D1FS = XBP/(XP**2)
      XCP = XP*D2XM-XP2*D2DM
      XCPP = XCP/(XP**2)
          D2FM = XCPP+2*D1DM*XAP/(XP**3)
      XDP = XP*D2XSM-D1XM*D1DS-XP2*D2DSM+D1DM*D1XS
      XDPP = XDP/(XP**2)
          D2FSM = XDPP+2*D1DS*XAP/(XP**3)
      XEP = XP*D2XS-XP2*D2DS
      XEPP = XEP/(XP**2)
          D2FS = XEPP+2*D1DS*XBP/(XP**3)
56    RETURN
```

```
      END
C************************************
C       DIFF PPP  OF  WEIBULL DIST.
C************************************
      SUBROUTINE DIFP(I,XC2,XT2,P,D1PC,D1PT,D2PC,D2PTC,D2PT)
      DOUBLE PRECISION XP,XP2,XP1,XDP2,XDP1,XXP2,XXP,
     *      D1PC,D1PT,D2PC,D2PTC,D2PT,
     *      XC1,XT1,XC0,XT0,XC2,XT2
      COMMON/SON/X(1000),XBAR(200)
     *     /SAM/C(0:200),FQ(0:200),F(0:200),EFQ(0,200)
     *     /PAA4/XD,N,K
     *     /SIT/DIVG,NEG
      CHARACTER DIVG*1,NEG*1
        J = I-1
        CCI = C(I)
        CCJ = C(J)
        XDP2 = (CCI**XT2)-(XD**XT2)
        XDP1 = (CCJ**XT2)-(XD**XT2)
         XP2 = EXP(-1*XC2*XDP2)
         XP1 = EXP(-1*XC2*XDP1)
          P = (XP1)-(XP2)
          IF (P.EQ.0.) THEN
            DIVG = 'Y'
            GOTO 67
          END IF
          D1PC = (XDP2*XP2)-(XDP1*XP1)
        CC1 = CCI**XT2
        R1 = CC1*ALOG(CCI)
        CC2 = XD**XT2
        R2 = CC2*ALOG(XD)
        RDP2 = R1-R2
        CC3 = CCJ**XT2
        R3 = CC3*ALOG(CCJ)
        RDP1 = R3-R2
          D1PT = ZC1*((RDP2*XP2)-(RDP1*XP1))
        TDP2 = XC2*((XD**XT2)-(CCI**XT2))+1
```

```
      TDP1 = XC2*((XD**XT2)-(CCJ**XT2))+1
        D2PC = (XDP1**2)*XP1-(XDP2**2)*XP2
        D2PTC = ((RDP2*XP2)*TDP2)-((RDP1*XP1)*TDP1)
      S1 = (ALOG(CCI))**2
      SS1 = S1*(CCI**XT2)
      S2 = (ALOG(XD))**2
      SS2 = S2*(XD**XT2)
      SDP2 = SS1-SS2
      S3 = (ALOG(CCJ))**2
      SS3 = S3*(CCJ**XT2)
      SDP1 = SS3-SS2
      D2PT2 = (XC2)*(SDP2-XC2*(RDP2**2))*XP2
      D2PT1 = (XC2)*(SDP1-XC2*(RDP1**2))*XP1
        D2PT = D2PT2-D2PT1
67    RETURN
      END
C************************************
C       DIFF FFF  OF  WEIBULL DIST.
C************************************
      SUBROUTINE DIFF(I,XC2,XT2,FF,D1FC,D1FT,D2FC,D2FTC,D2FT)
      DOUBLE PRECISION XDP,XP,XDP2,SDP2,RDP2,XDPP1,XDPP2,CC11,CDF,XDPP3,
     *      R1,R2,RDP,S1,S2,SS1,SS2,SDP,XDPP11,
     *      FF,D1FC,D1FT,D2FC,D2FTC,D2FT,
     *      XC1,XT1,XC0,XT0,XT2,XC2
      COMMON/SON/X(1000),XBAR(200)
     *     /SAM/C(0:200),FQ(0:200),F(0:200),EFQ(0:200)
     *     /PAR4/XD,N,K
     *     /SIT/DIVG,NEG
      CHARACTER DIVG*1,NEG*1
        CCI = C(I)
        XDPP11 = (XD/CCI)**XT2
        IF (XDPP11.EQ.0.) THEN
          DIVG = 'Y'
          GOTO 67
        END IF
        XDPP1 = 1/XDPP11
```

```
      XDPP2 = XDPP1-1
      XDP = (XD**XT2)*XDPP2
      XDPP3 = XC2*XDP
      XP = EXP(-1*XDPP3)
      IF (XP.EQ.0.) THEN
        DIVG = 'Y'
        GOTO 67
      END IF
      CDF = 1.-XP
        FF = F(I)-CDF
        D1FC = XDP*XP
      R1 = XDPP1*ALOG(CCI)
      R2 = R1-ALOG(XD)
      RDP = (XD**XT2)*R2
        D1FT = XC2*XP*RDP
      TDP = 1.-XDPP3
      XDP2 = XDP*XP
        D2FC = -1*XDP2*XDP
        D2FTC = XP*RDP*TDP
      S1 = (ALOG(CCI))**2
      S2 = (ALOG(XD))**2
      SS1 = XDPP1*S1
      SS2 = SS1-S2
      SDP = (XD**XT2)*SS2
      SDP2 = XP*SDP
      RDP2 = XP*RDP
        D2FT = XC2*(SDP2-XC2*(RDP2*RDP))
67    RETURN
      END
C*********************************
C     DIFF  PPP  OF  PARETO DIT.
C*********************************
      SUBROUTINE DIFP(I,XA2,XL2,P,D1PA,D1PL,D2PA,D2PAL,D2PL)
      DOUBLE PRECISION P,XDP2,XDP1,D1PL,D1PA,D2PL,D2PAL,D2PA
      COMMON/SON/X(1000),XBAR(200)
     *      /SAM/C(0:200),FQ(0:200),F(0:200),EFQ(0:200)
```

```
     /PAR4/XD,N,K
    J = I-1
    CCI = C(I)
    CCJ = C(J)
    XP2 = (XL2+XD)/(XL2+CCI)
    XP1 = (XL2+XD)/(XL2+CCJ)
      P = (XP1**XA2)-(XP2**XA2)
    XDP2 = (((XP2**(XA2-1))*(CCI-XD))/(XL2+CCI))/(XL2+CCI)
    XDP1 = (((XP1**(XA2-1))*(CCJ-XD))/(XL2+CCJ))/(XL2+CCJ)
      D1PL = XA2*(XDP1-XDP2)
    RDP2 = (XP2**XA2)*ALOG(XP2)
    RDP1 = (XP1**XA2)*ALOG(XP1)
      D1PA = RDP1-RDP2
    TDP2 = ((XA2+1)/(XL2+CCI))*(CCI-XD)-2
    TDP1 = ((XA2+1)/(XL2+CCJ))*(CCJ-XD)-2
      D2PL = XA2*(XDP1*TDP1-XDP2*TDP2)/(XL2+XD)
    EDP2 = XA2*ALOG(XP2)+1
    EDP1 = XA2*ALOG(XP1)+1
      D2PAL = XDP1*EDP1-XDP2*EDP2
    SDP2 = (XP2**XA2)*((ALOG(XP2))**2)
    SDP1 = (XP1**XA2)*((ALOG(XP1))**2)
      D2PA = SDP1-SDP2
    RETURN
    END
C***********************************
C       DIFF FFF  OF  PARETO DIST.
C***********************************
     SUBROUTINE DIFF(I,XA2,XL2,FF,D1FA,D1FL,D2FA,D2FAL,D2FL)
     DOUBLE PRECISION XDPP,D1FL,D1FA,D2FL,D2FAL,D2FA
     COMMON/SON/X(1000),XBAR(200)
    *      /SAM/C(0:200),FQ(0:200),F(0:200),EFQ(0:200)
    *      /PAR4/XD,N,K
     CCI = C(I)
       FF = F(I)-CDF(XD,XA2,XL2,CCI)
     XP = (XL2+XD)/(XL2+CCI)
     XDPP = (XP**(XA2-1))
```

```
      XDP = (XDPP*(CCI-XD)/(XL2+CCI))/(XL2+CCI)
         D1FL = -1*XDP*XA2
      RDP = (XP**XA2)*ALOG(XP)
         D1FA = -1*RDP
      TDP = ((CCI-XD)/(XL2+CCI))*(XA2+1)-2
         D2FL = -1*XA2*XDP*TDP/(XL2+XD)
      EDP = XA2*ALOG(XP)+1
         D2FAL = -1*XDP*EDP
      SDP = (XP**XA2)*((ALOG(XP))**2)
         D2FA = -1*SDP
      RETURN
      END
C*****************************************
C     MAXIMUM LIKELIHOOD ESTIMATOR
C*****************************************
      SUBROUTINE RMLE(MAX,ZL1,ZL2)
      DOUBLE PRECISION DET,P
      COMMON/PAR1/RMEW,SIGMA,XMW0,XSG0
     *     /PAR2/XC1,XT1,XC0,XT0
     *     /PAR3/XA1,XL1,XA0,XL0
     *     /SON/X(1000),XBAR(200)
     *     /SAM/C(0:200),FQ(0:200),F(0:200),EFQ(0:200)
     *     /SIR1/SEP1(3),SEP2(3)
     *     /SIT/DIVG,NEG
      CHARACTER DIVG*1,NEG*1
      RP10 = XMW0
      RP20 = XSG0
C     RP10 = XC0
C     RP20 = XT0
C     RP10 = XA0
C     RP20 = XL0
      RP1 = RMEW
      RP2 = SIGMA
C     RP1 = XC1
C     RP2 = XT1
C     RP1 = XA1
```

```
C     RP2 = XL1
      ZL1 = 0.
      ZL2 = 0.
      IC = 0
      DO 65 M = 1,MAX
       G1 = 0.
       G2 = 0.
       G11 = 0.
       G12 = 0.
       G22 = 0.
       DO 70 I = 1,K
        CALL DIFP(I,RP10,RP20,P,D1P1,D1P2,D2P1,D2P12,D2P2)
        IF (DIVG.EQ.'Y') THEN
         GOTO 12
        END IF
        XG1 = (D1P1/P)*FQ(I)
        XG2 = (D1P2/P)*FQ(I)
        XG11 = (D2P1/P-(D1P1/P)**2)*FQ(I)
        XG12 = ((D2P12/P-(D1P1/P)*(D1P1/P))*FQ(I)
        XG22 = ((D2P2/P-(D1P2/P)**2)*FQ(I)
         G1 = G1+XG1
         G2 = G2+XG2
        G11 = G11+XG11
        G12 = G12+XG12
        G21 = G12
        G22 = G22+XG22
70     CONTINUE
       DET = (G11*G22)-(G12*G21)
      IF (DET.EQ.0) THEN
       DIVG = 'Y'
       GOTO 12
      END IF
         ZL1 = RP10-((G22/DET)*G1-(G12/DET)*G2)
         ZL2 = RP20-((G11/DET)*G2-(G21/DET)*G1)
        IC = IC+1
       IF (RP10.LE.0.).OR.(RP20.LE.0)) THEN
```

```
        NEG = 'Y'
        GOTO 12
       END IF
       IF((ABS(ZL1-RP10).LT.0.0001).AND.(ABS(ZL2-RP20).LT.0.0001))THEN
C        RP10 = ZL1
C        RP20 = ZL2
        GOTO 600
       ELSE
        RP10 = ZL1
        RP20 = ZL2
       END IF
       IF (IC.GE.MAX) THEN
        DIVG = 'Y'
        GOTO 12
       END IF
 65  CONTINUE
600  SEP1(1) = (ZL1-RP1)**2
     SEP2(1) = (ZL2-RP2)**2
 12  RETURN
     END
C*****************************************
C     MINIMUM CHI-SQUARE ESTIMATOR
C*****************************************
     SUBROUTINE RMCE(MAX,ZC1,ZC2)
     DOUBLE PRECISION DET,P
     COMMON/PAR1/RMEW,SIGMA,XMW0,XSG0
    *      /PAR2/XC1,XT1,XC0,XT0
    *      /PAR3/XA1,XL1,XA0,XL0
    *      /SON/X(1000),XBAR(200)
    *      /SAM/C(0:200),FQ(0:200),F(0:200),EFQ(0:200)
    *      /SIR/SEP1(3),SEP2(3)
    *      /SIT/DIVG,NEG
     CHARACTER DIVG*1,NEG*1
     RP10 = XMW0
     RP20 = XSG0
C     RP10 = XC0
```

```
C     RP20 = XT0
C     RP10 = XA0
C     RP20 = XL0
      RP1 = RMEW
      RP2 = SIGMA
C     RP1 = XC1
C     RP2 = XT1
C     RP1 = XA1
C     RP2 = XL1
      ZC1 = 0.
      ZC2 = 0.
      IC = 0
      DO 165 M = 1,MAX
      G1 = 0.
      G2 = 0.
      G11 = 0.
      G12 = 0.
      G22 = 0.
       DO 170 I = 1,K
        CALL DIFP(I,RP10,RP20,P,D1P1,D1P2,D2P1,D2P12,D2P2)
        IF (DIVG.EQ.'Y') THEN
          GOTO 13
        END IF
        FQI = FQ(I)
        RG = FQI/((N*P)
          XG1 = N*D1P1-1*(D1P1/P)*RG*FQI
           G1 = G1+XG1
          XG2 = N*D1P2-1*(D1P2/P)*RG*FQI
           G2 = G2+XG2
          XG11 = N*D2P1-(D2P1/P-2*(D1P1/P)*(D1P1/P))*RG*FQI
           G11 = G11+XG11
          XG12 = N*D2P12-(D2P12/P-2*(D1P1/P)*(D1P2/P))*RG*FQI
           G12 = G12+XG12
           G21 = G12
          XG22 = N*D2P2-(D2P2/P-2*(D1P2/P)*(D1P2/P))*RG*FQI
           G22 = G22+XG22
```

```
170 CONTINUE
    DET = (G11*G22)-(G12*G21)
    IF (DET.EQ.0) THEN
     DIVG = 'Y'
     GOTO 13
    END IF
     ZC1 = RP10-((G22/DET)*G1-(G12/DET)*G2)
     ZC2 = RP20-((G11/DET)*G2-(G21/DET)*G1)
    IC = IC+1
     IF (RP10.LE.0.).OR.(RP20.LE.0)) THEN
       NEG = 'Y'
       GOTO 12
     END IF
    IF((ABS(ZC1-RP10).LT.0.0001).AND.(ABS(ZC2-RP20).LT.0.0001))THEN
     NEG = 'Y'
C      RP10 = ZC1
C      RP20 = ZC2
     GOTO 700
    ELSE
     RP10 = ZC1
     RP20 = ZC2
    END IF
    IF (IC.GE.MAX) THEN
       DIVG = 'Y'
       GOTO 13
     END IF
165  CONTINUE
700  SEP1(2) = (ZC1-RP1)**2
    SEP2(2) = (ZC2-RP2)**2
 13  RETURN
    END
C*************************************************************
C    CRAMER-VON MISES MINIMUM DISTANCE ESTIMATOR
C*************************************************************
    SUBROUTINE RMDE(MAX,ZD1,ZD2)
    DOUBLE PRECISION DET
```

```
      COMMON/PAR1/RMEW,SIGMA
     *     /PAR2/XC1,XT1,XC0,XT0
     *     /PAR3/XA1,XL1,XA0,XL0
     *     /SON/X(1000),XBAR(200)
     *     /SAM/C(0:200),FQ(0:200),F(0:200),EFQ(0:200)
     *     /SIR/SEP1(3),SEP2(3)
     *     /SIT/DIVG,NEG
      CHARACTER DIVG*1,NEG*1
      RP10 = XMW0
      RP20 = XSG0
C     RP10 = XC0
C     RP20 = XT0
C     RP10 = XA0
C     RP20 = XL0
      RP1 = RMEW
      RP2 = SIGMA
C     RP1 = XC1
C     RP2 = XT1
C     RP1 = XA1
C     RP2 = XL1
      ZD1 = 0.
      ZD2 = 0.
      IC = 0
      DO 265 M = 1,MAX
      G1 = 0.
      G2 = 0.
      G11 = 0.
      G12 = 0.
      G22 = 0.
       DO 270 I = 1,K
        CALL DIFF(I,RP10,RP20,FF,D1F1,D1F2,D2F1,D2F12,D2F2)
        IF (DIVG.EQ.'Y') THEN
          GOTO 14
        END IF
          XG1 = -2*FF*D1F1
           G1 = G1+XG1
```

```
          XG2 = -2*FF*D1F2
           G2 = G2+XG2
          XG11 = 2*((D1F1**2)-(FF*D2F1))
           G11 = G11+XG11
          XG12 = 2*((D1F1*D1F2)-(FF*D2F12))
           G12 = G12+XG12
           G21 = G12
          XG22 = 2*((D1F2**2)-(FF*D2F2))
           G22 = G22+XG22
270 CONTINUE
      IT = 100000000.
      DET = (G11*G22-G12*G21)*IT
      IF (DET.EQ.0) THEN
        DIVG = 'Y'
        GOTO 14
      END IF
        ZD1 = RP10-(((G1/DET)*G22-(G12/DET)*G2)*IT)
        ZC2 = RP10-(((G11/DET)*G2-(G1/DET)*G21)*IT)
      IC = IC+1
        IF (RP10.LE.0.).OR.(RP20.LE.0)) THEN
          NEG = 'Y'
          GOTO 12
        END IF
      IF((ABS(ZD1-RP10).LT.0.0001).AND.(ABS(ZD2-RP20).LT.0.0001))THEN
C         RP10 = ZD1
C         RP20 = ZD2
        GOTO 800
      ELSE
        RP10 = ZD1
        RP20 = ZD2
      END IF
      IF (IC.GE.MAX) THEN
         DIVG = 'Y'
         GOTO 14
        END IF
265   CONTINUE
```

```
800   SEP1(3) = (ZD1-RP1)**2
      SEP2(3) = (ZD2-RP2)**2
 14   RETURN
      END
```

ประวัติผู้เขียน

นางสาวปาริฉัตร อัครจันทโชติ เกิดเมื่อวันที่ 13 มีนาคม 2513 สำเร็จการศึกษาปริญญาวิทยาศาสตรบัณฑิต (วท.บ.) สาขาสถิติ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ในปีการศึกษา 2534 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2535