รายการอ้างอิง

ภาษาไทย

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

สากัณย์ สุวรรณการ. วิธีนอนพาราเมตริกสำหรับการประมาณค่าฟังก์ชันการอยู่รอดในปัญหา 2 ตัวอย่างที่มีค่าสังเกตไม่สมบูรณ์. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย, 2530.

นันทพร อารยะสกุลวงศ์. การประมาณค่าฟังก์ชันการอยู่รอดสำหรับข้อมูลที่มีค่าถูกตัดทิ้ง. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย, 2539.

ภาษาอังกฤษ

Ferguson Thomas S. and Eswar G. Phadia, "Bayesian Nonparametric Estimation Based on Censored Data," The Annals of Statistics 7(1979) : 163-186.

Kaplan EL and Meier P., "Nonparametric Estimation From Incomplete Observations," Journal of the American Statistical Association 53(1958) : 457-481.

Lawless J.F, Statistical Models and Methods for lifetime Data. New York: John Wiley and Sons, 1982.

London, D., Survival Models and Their Estimation. Winsted: ACTEX Publications, Inc., 1988.

Rupert G. Miller, Jr., Survival Analysis. New York: John Wiley and Sons, 1981.

Susarla V. and Ryzin J. Van, "Nonparametric Bayesian Estimation of Survival Curve from Incomplete observations," Journal of the American Statistical Association 71(1976): 897-902.

Tobias, Paul A., Applied reliability. Van Nostrand Reinhod Company Inc., 1986.

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

การสร้างตัวเลขสุ่ม (Random Number)

การสร้างลักษณะการแจกแจงแบบต่างๆ นั้น จะต้องอาศัยตัวเลขสุ่มเป็นพื้นฐานในการสร้าง สำหรับการวิจัยครั้งนี้จะใช้วิธีการสร้างตัวเลขสุ่มตามวิธีของ White และ Schmidt (1975) ซึ่งขั้นตอนในการสร้าง จะแสดงรายละเอียดด้วยโปรแกรมย่อยต่อไปนี้

```
      SUBROUTINE RAND(IX,IY,YFL)
      IY = IX*65539
      IF (IY) 10,20,20
10    IY = IY+2147483647+1
20    YFL = IY
      YFL = YFL/2147483647
      IX = IY
      RETURN
      END
```

ค่า IX จะเป็นค่า SEED เป็นค่าเริ่มต้น ซึ่งจะต้องเป็นจำนวนเต็มบวกที่เป็นเลขคี่ RAND จะเป็นค่าของตัวเลขสุ่มที่มีค่าอยู่ระหว่าง 0 กับ 1

การสร้างการแจกแจงแบบไวบูลล์ (Weibull Distribution)

ฟังก์ชันความหนาแน่น และฟังก์ชันการแจกแจงสะสม แสดงได้ดังนี้

$$f(t)=\begin{cases} c\tau t^{\tau-1}\exp(-ct^{\tau}) & ,t\geq 0, c>0, \tau>0 \\ 0 & ,t<0 \end{cases}$$

$$F(t)=1-\exp(-ct^{\tau})$$

การสร้างตัวแปรสุ่มให้มีการแจกแจงแบบไวบูลล์ อาศัยหลักการแปรผกผันโดยมีขั้นตอนการสร้างดังนี้

ขั้นที่ 1 F(t) = YFL โดยที่ YFL คือตัวเลขสุ่มแบบสม่ำเสมอช่วง (0,1)

ขั้นที่ 2 หาค่า

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

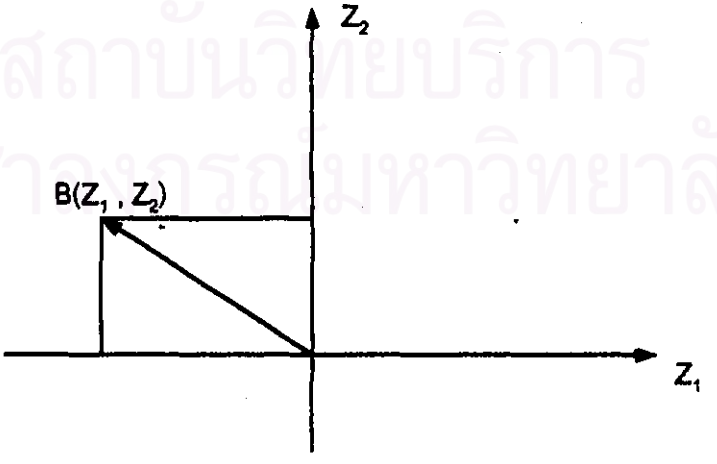
ฟังก์ชันที่ใช้สร้างตัวแปรสุ่มที่มีการแจกแจงแบบไวบูลล์ คือฟังก์ชัน WEIBUL(CT,TT)

```
      FUNCTION WEIBUL(CT,TT)
      COMMON /SEED/IX
10    CALL RAND(IX,IY,YFL)
      IF((YFL.EQ.1.0).OR.(YFL.EQ.0.0))GOTO 10
      WEIBUL = ((1.0/CT)*(-1.0*ALOG(1.0-YFL)))**(1.0/TT)
      RETURN
      END
```

การสร้างการแจกแจงแบบลอกนอร์มอล

การสร้างตัวแปรสุ่มให้มีการแจกแจงแบบลอกนอร์มอล ทำได้โดยการหาค่า Exponential ของตัวแปรสุ่มที่มีการแจกแจงปกติ โดยที่การสร้างตัวแปรสุ่มที่มีการแจกแจงปกติมีวิธีการดังนี้

การสร้างเลขสุ่มที่มีการแจกแจงแบบปกติใช้วิธีการของ Box Muller (1958) โดยการสร้างตัวเลขสุ่มที่มีการแจกแจงปกติมาตราฐาน : N(0,1) พร้อมกัน 2 ค่าและแต่ละค่าเป็นอิสระกันโดยใช้ตัวผลิต (Generater) Z_1 และ Z_2 พิจารณาดังรูปต่อไปนี้



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

พิจารณาจากรูปจะได้

$$Z_1 = B\cos(\theta) \quad \text{———————} \quad (1)$$
$$Z_2 = B\sin(\theta) \quad \text{———————} \quad (2)$$

เนื่องจาก $B^2 = Z_1^2 + Z_2^2$ มีการแจกแจงไคสแควร์ด้วยระดับความเป็นอิสระ 2 และเทียบเท่ากับการแจกแจงเอกซโพเนนเชียล ค่าเฉลี่ยเท่ากับ 2 โดยวิธีการแปรผกผัน (Inverse Transformation) สามารถสร้างเลขสุ่มที่มีการแจกแจงแบบเอกซโพเนนเชียลได้ดังนี้

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

เมื่อ R เป็นเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอในช่วง (0,1)

จากการสมมาตรของการแจกแจงปกติ (Normal Distribution) จะได้ว่ามุม θ มีการแจกแจงสม่ำเสมอระหว่าง 0 ถึง 2π เรเดียน และรัศมี B กับมุม θ เป็นอิสระกัน จาก (1),(2) และ (3) เราสามารถสร้างเลขสุ่มที่มีการแจกแจงปกติมาตรฐาน จากเลขสุ่ม 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 เป็นตัวเลขสุ่มที่สร้างขึ้นจากโปรแกรมย่อย RAND (IX,IY,YFL) เมื่อได้ตัวเลขสุ่มที่มีการแจกแจงปกติมาตรฐานแล้ว จะทำการแปลงค่าเลขสุ่มดังกล่าวโดยอาศัยฟังก์ชัน

$$GN_1 = \mu + \delta Z_1$$
$$GN_2 = \mu + \delta Z_2$$

ซึ่งจะได้ว่า GN_1 และ GN_2 มีการแจกแจงแบบปกติด้วยค่าเฉลี่ยเท่ากับ μ และมีความแปรปรวนเท่ากับ δ^2 $(GN_i \sim N(\mu,\delta^2)\ ;i=1,2)$

ฟังก์ชันที่ใช้ในการสร้างเลขสุ่มที่มีการแจกแจงแบบปกติ มีค่าเฉลี่ยเท่ากับ μ และมีค่าเบี่ยงเบนมาตราฐานเท่ากับ δ คือฟังก์ชัน NORMAL(DMEAN,SIGMA)

```
FUNCTION NORMAL(DMEAN,SIGMA)
    COMMON /SEED/IX/SELECT/KK
    REAL*8  NORMAL
    PI = 3.141526
    IF(KK.EQ.1) GOTO 10
    CALL RAND(IX,IY,YFL)
```

```
      RONE = YFL
      CALL RAND(IX,IY,YFL)
      RTWO = YFL
      ZONE = SQRT(-2*ALOG(RONE))*COS(2*PI*RTWO)
      ZTWO = SQRT(-2*ALOG(RONE))*SIN(2*PI*RTWO)
      NORMAL = ZONE*SIGMA+DMEAN
      KK = 1
      RETURN
   10 NORMAL = ZTWO*SIGMA+DMEAN
      KK = 0
      RETURN
      END
```

การสร้างการแจกแจงแบบพาเรโต (Pareto Distribution)

ฟังก์ชันความหนาแน่น และฟังก์ชันการแจกแจงสะสม แสดงได้ดังนี้

$$f(t) = \begin{cases} \alpha\lambda^{\alpha}(\lambda+t)^{-\alpha-1} & ,t>0,\alpha>0,\alpha>1 \\ 0 & ,t<0 \end{cases}$$

$$F(t) = 1-\left(\frac{\lambda}{\lambda+1}\right)^{\alpha}$$

การสร้างตัวแปรสุ่มให้มีการแจกแจงแบบพาเรโต อาศัยหลักการแปรผกผันโดยมีขั้นตอนการสร้างดังนี้

ขั้นที่ 1 F(t) = YFL โดยที่ YFL คือตัวเลขสุ่มแบบสม่ำเสมอช่วง (0,1)

ขั้นที่ 2 หาค่า

$$t = \left[-\frac{\lambda}{\left[1-F(t)\right]^{\frac{1}{\alpha}}}\right]-\lambda$$

ฟังก์ชันที่ใช้สร้างตัวแปรสุ่มที่มีการแจกแจงแบบพาเรโต คือฟังก์ชัน PARE(AL,AGT)

```
      FUNCTION PARE(AL,AGT)
      COMMON /SEED/IX
   10 CALL RAND(IX,IY,YFL)
      IF((YFL.EQ.1.0).OR.(YFL.EQ.0.0))GOTO 10
     PARE = (AGT/((1-YFL)**(1/AL)))-AGT
     RETURN
     END
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข.

ตัวอย่าง โปรแกรมภาษาฟอร์แทน 77 ที่ใช้ในการศึกษาวิจัย ในการจำลองสถานการณ์ของข้อมูลที่มีการแจกแจงแบบลอกนอร์มอล

```
C*************************************************************************
□
C*     ESTIMATING SURVIVAL FUNCTION FOR RIGHT CENSOR DATA          *
C*************************************************************************
      DIMENSION X(100),IDX(100),Y(100),SLOG(100),TC(10),NPC(4),XN(100),
     * REPL(100),SREPL(100),SPL(100),SUMSP(100),AVGSP(100),
     * RELT(100),SRELT(100),SLT(100),SUMSL(100),AVGSL(100),
     * REMO(100),SREMO(100),SMO(100),SUMSMO(100),AVGSMO(100),
     * REML(100),SREML(100),SML(100),SUMSML(100),AVGSML(100),
     * REBL(100),SREBL(100),SBL(100),SUMSBL(100),AVGSBL(300),
     * AVGPL(10,10),AVGLT(10,10),AVGMO(10,10),AVGML(10,10),AVGBL(10,10)
      COMMON /SEED/IX/SELECT/KK
      INTEGER PC
      REAL*8 NORMAL
      DATA (NPC(J),J=1,4)/20,30,40,50/
      DATA (TC(K),K=1,9)/2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0/
C*************************************************************************
C*                    CONSTANT VALUE                                     *
C*************************************************************************
      SIGMA  = 0.5
      ET     = 6.0
      N      = 100
      MPC    = 4
      NTG    = 9
      LO     = 1000
      ALPHAR = 1.0
      TINT   = 1.0
```

```
      TSTEP  = 1.0
      IX     = 673
C*************************************************************
□
C*                    MAIN PROGRAM                           *
C*************************************************************
C*  SET PARAMETER
      OPEN (1,FILE ='C:\PROG\L100.TXT')
      OPEN (2,FILE ='C:\PROG\SL100.TXT')
      DMEAN   = ALOG(ET)-(SIGMA**2)/2
      T50     = EXP(DMEAN)
      XMEN    = T50*EXP(SIGMA**2)/2
      VAR     = T50*EXP(SIGMA**2)*(EXP(SIGMA**2)-1)
C *  OPEN(6,FILE='LPT1')
      WRITE(6,*)
      WRITE(6,5)
 5    FORMAT(21X,'LOGNORMAL DISTRIBUTION')
      WRITE(6,6)DMEAN,SIGMA,XMEN,VAR
 6    FORMAT('DMEAN=',F7.4,2X,'SIGMA=',F7.4,2X,'MEAN=',F6.4,2X,
     *'VAR=',F10.4)
      WRITE(6,7)N,IX,ALPHAR,LO
 7    FORMAT(6X,'N=',I2,2X,'IX=',I4,2X,'ALPHAR=',F4.2,2X,
     *'LOOP=',I4)
C*************************************************************
C              LOOP  FOR VARY TC POINT                      *
C*************************************************************
      DO 991 K=1,NTC
      NP = 0
      IX = 673
C######### VARY POINT ############
      DO 992 TK1=TINT,TC(K),TSTEP
        NP    = NP+1
        Y(NP) = TK1
```

```
992   CONTINUE
C######### VARY PERCENT CENSORED ###########
      DO 200 J=1,MPC
      PC      = NPC(J)
      NC      = (N*PC)/100
      NU      = N-NC
      IA      = 65539
      KK      = 0
      DO 110 LI=1,NP
      SREPL(LI)      = 0.0
      SRELT(LI)      = 0.0
      SREMO(LI)      = 0.0
      SREML(LI)      = 0.0
      SREBL(LI)      = 0.0
      SUMSP(LI)      = 0.0
      SUMSL(LI)      = 0.0
      SUMSMO(LI)     = 0.0
      SUMSML(LI)     = 0.0
      SUMSBL(LI)     = 0.0
110   CONTINUE
      SUM     = 0.0
      STMED   = 0.0
C######### VALUE BEGIN ###########
      DO 101 L=1,LO
      K1      = 0
      K2      = 0
      I       = 0
      SUMT    = 0.0
      PDL     = 0.0
      PLT     = 0.0
      PMO     = 0.0
      PML     = 0.0
      PBL     = 0.0
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
C######### GEN DATA LOGNORMAL DISTRIBUTION ###########
      DO 30 I1=1,N
60     XN(I1)  = NORMAL(DMEAN,SIGMA)
       X(I1)   = EXP(XN(I1))
       IDX(I1) = JINDEX(X(I1),TC(K))
       IF (IDX(I1).EQ.1) THEN
        K1     = K1 + 1
        IF (K1.LE.NU) THEN
         GOTO 30
        ELSE
        IF (K2.LT.NC) THEN
         GOTO 60
        ELSE
         GOTO 31
        ENDIF
        ENDIF
       ELSE
        K2     = K2 + 1
        IF (K2.LE.NC) THEN
         GOTO 30
        ELSE
        IF (K1.LT.NU) THEN
         GOTO 60
        ELSE
         GOTO 31
        ENDIF
        ENDIF
       ENDIF
 30   CONTINUE
 31   DO 32 I1=1,N
      IF(X(I1).EQ.TC(K)) THEN
       I = I + 1
      ENDIF
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
      SUMT    = SUMT + X(I1)
      SUM     = SUM + X(I1)
   32 CONTINUE
      POS1    = N/2
      POS2    = POS1 + 1
      TMED    = (X(POS1)+X(POS2))/2
      STMED   = STMED + TMED
      CALL RANK(N,X,IDX)

C######## SEND DATA TO CAL S(X) ##########
      DO 140 LI=1,NP
      SLOG(LI) = 0.0
      XS      = Y(LI)
      Z0      = (ALOG(XS/T50)/SIGMA)
      CD      = CDFN(Z0)
      SLOG(LI) = 1 - CD
      IF (LI.EQ.1) THEN
        XS0 = 0.0
        CALL PRODUC(N,NC,X,XS0,XS,TC(K),PDL)
        SPL(LI)  = PDL
        CALL LIFE(N,NC,X,XS0,XS,TC(K),PLT)
        SLT(LI)  = PLT
        CALL CLS(N,X,XS0,XS,PMO)
        SMO(LI)  = PMO
      ELSE
        XS0 = Y(LI-1)
        CALL PRODUC(N,NC,X,XS0,XS,TC(K),PDL)
        SPL(LI)  = SPL(LI-1)*PDL
        CALL LIFE(N,NC,X,XS0,XS,TC(K),PLT)
        SLT(LI)  = SLT(LI-1)*PLT
        CALL CLS(N,X,XS0,XS,PMO)
        SMO(LI)  = SMO(LI-1)*PMO
      ENDIF
```

```
      CALL BAYD(N,NU,X,XS,TC(K),SUMT,ALPHAR,BAY)
      SBL(LI) = BAY
      CALL BAYH(N,NU,X,XS,TC(K),SUMT,PML)
      SML(LI) = PML
      SUMSP(LI)      = SUMSP(LI) + SPL(LI)
      SUMSL(LI)      = SUMSL(LI) + SLT(LI)
      SUMSMO(LI)     = SUMSMO(LI) + SMO(LI)
      SUMSML(LI)     = SUMSML(LI) + SML(LI)
      SUMSBL(LI)     = SUMSBL(LI) + SBL(LI)
140   CONTINUE

C###### STOP DO LOOP 1 TO NP #########
C
101   CONTINUE
C
C###### STOP DO LOOP 1 TO LO #########
C
      SUMPL    = 0.0
C
      SUMLT    = 0.0
C
      SUMMO    = 0.0
      SUMML    = 0.0
      SUMBL    = 0.0
      AMED     = STMED/LO
      TMEAN    = (SUM/(N*LO))
      XMEAN    = T50*EXP((SIGMA**2)/2)
      WRITE(6,*)
      WRITE(6,291)TMEAN,XMEAN,TC(K)
291   FORMAT(2X,'DTMEAN=',F12.4,'XMEAN=',F12.4,'TC=',F6.2)
      WRITE(6,*)
      DO 160 LI=1,NP
       TT = Y(LI)
```

```
      AVGSP(LI)     = SUMSP(LI)/LO
      AVGSL(LI)     = SUMSL(LI)/LO
      AVGSMO(LI)    = SUMSMO(LI)/LO
      AVGSML(LI)    = SUMSML(LI)/LO
      AVGSBL(LI)    = SUMSBL(LI)/LO
      CALL RELATE(SLOG(LI),AVGSP(LI),RE)
      REPL(LI)      = RE
      CALL RELATE(SLOG(LI),AVGSL(LI),RE)
      RELT(LI)      = RE
      CALL RELATE(SLOG(LI),AVGSMO(LI),RE)
      REMO(LI)      = RE
      CALL RELATE(SLOG(LI),AVGSML(LI),RE)
      REML(LI)      = RE
      CALL RELATE(SLOG(LI),AVGSBL(LI),RE)
      REBL(LI)      = RE
      SUMPL = SUMPL + REPL(LI)
      SUMLT = SUMLT + RELT(LI)
      SUMMO = SUMMO + REMO(LI)
      SUMML = SUMML + REML(LI)
      SUMBL = SUMBL + REBL(LI)
  160 CONTINUE
C###### STOP DO LOOP CAL RELATIVE ERROR #########
     AVGPL(K,J)     = SUMPL/NP
     AVGLT(K,J)     = SUMLT/NP
     AVGMO(K,J)     = SUMMO/NP
     AVGML(K,J)     = SUMML/NP
     AVGBL(K,J)     = SUMBL/NP
     WRITE(6,*)
     WRITE(6,392)PC
  392 FORMAT (2X,'Percent of Censoring = ',I2,'%')
     WRITE(6,193)
  193 FORMAT(1X,72('-'))
     WRITE(6,1161)
```

```
1161 FORMAT(2X,'T',5X,'SPL',5X,'APEPL',
   * 5X,'SLT',5X,'APELT',5X,'SMO',5X,'APEMO',
   * 5X,'SML',5X,'APEML',5X,'SBL',5X,'APEBL')
    WRITE(6,193)
    DO 163 LI = 1,NP
    TT = Y(LI)
    WRITE(6,1163)TT,AVGSP(LI),REPL(LI),
   * AVGSL(LI),RELT(LI), AVGSMO(LI),REMO(LI),
   * AVGSML(LI),REML(LI), AVGSBL(LI),REBL(LI)
    WRITE(1,1163)TT,AVGSP(LI),REPL(LI),
   * AVGSL(LI),RELT(LI), AVGSMO(LI),REMO(LI),
   * AVGSML(LI),REML(LI), AVGSBL(LI),REBL(LI)
1163  FORMAT(1XF5.2,5(F9.3,F9.3))
163   CONTINUE
C###### STOP DO LOOP PRINT VALUE OF S(X) AND ERROR #########
    WRITE(6,193)
    WRITE(6,1201)AVGPL(K,J),AVGLT(K,J),AVGMO(K,J),
   *     AVGML(K,J),AVGBL(K,J)
1201  FORMAT(1X,'MAPE',5(5X,F10.4))
    WRITE(*,2221)'   CONTINUE (Y/N) : '
2221  FORMAT(A\)
    READ(*,2221)ANS
    IF ((ANS.EQ.'N').OR.(ANS.EQ.'n')) THEN
     GOTO 2223
    ENDIF
 200  CONTINUE
C###### STOP DO LOOP MAIN #########
 991  CONTINUE
C###### STOP DO LOOP MAIN #########
    WRITE(6,*)
    WRITE(6,*)'    **********SUMMARY*********'
    DO 199 K = 1,NTC
     Z01 = (ALOG(TC(K)/T50))/SIGMA
```

```
      CDD = CDFN(Z01)
      SL  = 1 - CDD
     WRITE(6,*)
     WRITE(6,8)SL,TC(K)
  8  FORMAT(1X,'S(TC)=',F7.5,2X,'TC=',F5.2)
     WRITE(6,193)
     WRITE(6,1)
  1  FORMAT(3X,'TC',6X,'PC',8X,'LT',8X,'MOMENT',8X,'BAYH',8X,'BAYG')
  2  FORMAT(62('-'))
     DO 210 J = 1,MPC
       PC = NPC(J)
     WRITE(6,1202)PC,AVGPL(K,J),AVGLT(K,J),AVGMO(K,J),
    * AVGML(K,J),AVGBL(K,J)
1202  FORMAT(3X,I2,5(1X,F10.5))
     WRITE(2,1202)PC,AVGPL(K,J),AVGLT(K,J),AVGMO(K,J),
    * AVGML(K,J),AVGBL(K,J)
 210  CONTINUE
       WRITE(6,193)
 199  CONTINUE
2223  ENDFILE (UNIT = 1)
       ENDFILE (UNIT = 2)
       STOP
     END
C#########################################
C     SUBROUTINE RANDOM BARIABLE
C#########################################
     SUBROUTINE RAND(IX,IY,YFL)
     IY = IX*65539
     IF(IY)10,20,20
 10  IY = IY+2147483647+1
 20  YFL = IY
     YFL = YFL/2147483647
     IX = IY
```

```
      RETURN
      END
C#########################################
C     GEN DATA : NORMAL DISTRIBUTION
C#########################################
      FUNCTION NORMAL(DMEAN,SIGMA)
      COMMON /SEED/IX/SELECT/KK
      REAL*8  NORMAL
      PI = 3.141526
      IF(KK.EQ.1) GOTO 10
      CALL RAND(IX,IY,YFL)
      RONE = YFL
      CALL RAND(IX,IY,YFL)
      RTWO = YFL
      ZONE = SQRT(-2*ALOG(RONE))*COS(2*PI*RTWO)
      ZTWO = SQRT(-2*ALOG(RONE))*SIN(2*PI*RTWO)
      NORMAL = ZONE*SIGMA+DMEAN
      KK = 1
      RETURN
   10 NORMAL = ZTWO*SIGMA+DMEAN
      KK = 0
      RETURN
      END

C#########################################
C       FUNCTION FOR FIND INDEX
C#########################################
      FUNCTION JINDEX(X,TC)
       IF(X.LE.TC) THEN
        JINDEX = 1
       ELSE
        X = TC
        JINDEX = 0
```

```
      ENDIF
      RETURN
      END

C#####################################################
C       FUNCTION FOR FIND VALUE OF STANDARD NORMAL
C#####################################################
      FUNCTION CDFN(Z0)
      TLZ0 = (1.0/2.5066282746)*EXP((-1.0*(Z0**2)/2.0))
      TLWW = 1.0/(1.0+0.33267*ABS(Z0))
      TLP  = 1.0-TLZ0*(0.4361836*TLWW-0.1201676*(TLWW**2)
     *      +0.937298*(TLWW**3))
      IF(Z0.GE.0) THEN
        CDFN = TLP
      ELSE
        CDFN = 1.0-TLP
      ENDIF
      RETURN
      END

C#####################################################
C       SUBROUTINE FOR SORTING DATA
C#####################################################
      SUBROUTINE RANK(N,X,IDX)
      INTEGER I,J,SP,MID,LEFT(100),RIGHT(100),IDX(100)
      REAL X(100),PIVOT
      LEFT(1) = 1
      RIGHT(1) = N
      SP = 1
   10 IF (LEFT(SP).LT.RIGHT(SP)) GOTO 20
      SP = SP-1
      GOTO 130
   20 I = LEFT(SP)
```

```
      J = RIGHT(SP)
      PIVOT = X(J)
      MID = (I+J)/2
      IF (J-I.LT.6) GOTO 50
      IF ((PIVOT.GT.X(I)).AND.(PIVOT.LT.X(MID))) GOTO 50
      IF ((PIVOT.LT.X(I)).AND.(PIVOT.GT.X(MID))) GOTO 50
      IF ((X(I).LT.X(MID)).AND.(X(I).GT.PIVOT)) GOTO 30
      IF ((X(I).GT.X(MID)).AND.(X(I).LT.PIVOT)) GOTO 30
      CALL SWAP(X(I),X(J),IDX(I),IDX(J))
      GOTO 40
   30 CALL SWAP(X(I),X(J),IDX(I),IDX(J))
   40 PIVOT = X(J)
   50 IF (I.GE.J) GOTO 110
   60 IF (X(I).GE.PIVOT) GOTO 70
      I = I+1
      GOTO 60
   70 J = J-1
   80 IF(.NOT.((I.LT.J).AND.(PIVOT.LT.X(J)))) GOTO 90
      J = J-1
      GOTO 80
   90 IF(I.LT.J) CALL SWAP(X(I),X(J),IDX(I),IDX(J))
  100 GOTO 50
  110 J = RIGHT(SP)
      CALL SWAP(X(I),X(J),IDX(I),IDX(J))
      IF ((I-LEFT(SP)).GE.(RIGHT(SP)-I)) GOTO 120
      LEFT(SP+1) = LEFT(SP)
      RIGHT(SP+1) = I-1
      LEFT(SP) = I+1
      GOTO 125
  120 LEFT(SP+1) = I+1
      RIGHT(SP+1) = RIGHT(SP)
      RIGHT(SP) = I-1
  125 SP=SP+1
```

```
130 IF (SP.GT.0) GOTO 10
    RETURN
    END

C##################################################
C       SUBRUTINE SWAP X(I) WITH X(I+1)
C##################################################
    SUBROUTINE SWAP(X1,X2,IDX1,IDX2)
    T = X1
    X1 = X2
    X2 = T
    II = IDX1
    IDX1 = IDX2
    IDX2 = II
    RETURN
    END

C##################################################
C       SUBRUTINE FOR FINE S(X) BY PL ESTIMATOR
C##################################################
    SUBROUTINE PRODUC(N,NC,X,T0,T,TC,PDL)
    DIMENSION X(300)
    REAL  NK,NK1
    NK  = 0.0
    NK1 = 0.0
    DO 20 K = 1,N
     IF (X(K).GT.T) THEN
       NK1 = NK1+1
     ENDIF
     IF (X(K).GT.T0) THEN
       NK = NK+1
     ENDIF
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
20  CONTINUE
    IF (T.EQ.TC) THEN
     NK1 = NC
    ENDIF

    SX  = NK1/NK
    PDL = SX
    RETURN
    END


C##################################################
C       SUBRUTINE FOR FINDS(X) BY LT ESTIMATOR
C##################################################
    SUBROUTINE LIFE(N,NC,X,T0,T,TC,PLT)
    DIMENSION X(300)
    REAL   NK,NK1
    NK  = 0.0
    NK1 = 0.0
    DO 20 K = 1,N
     IF ((X(K).GT.T0).AND.(X(K).LE.T)) THEN
       NK1 = NK1+1
     ENDIF
     IF (X(K).GT.T0) THEN
       NK = NK+1
     ENDIF

20  CONTINUE
    IF (T.EQ.TC) THEN
     NK1 = NK1-NC
    ENDIF

    SX  = NK1/(NK-(NK1/2))
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
      SX1 = EXP(-1*SX)
      PLT = SX1
      RETURN
      END
C#####################################################
C       SUBRUTINE FOR FINDS(X) BY CLASSIC ESTIMATOR
C#####################################################
      SUBROUTINE CLS(N,X,T0,T,PMO)
      DIMENSION X(100)
      REAL  NK1
      NK1 = 0.0
      SK1 = 0.0
      SUN = 0.0
      DO 20 K = 1,N
       IF (X(K).LT.T) THEN
        IF ((X(K).GE.T0).AND.(X(K).LT.T)) THEN
         SUN = SUN+X(K)-T0
         NK1 = NK1+1
        ENDIF
         NK = K
       ENDIF
 20   CONTINUE
      SX  = ((N-NK)/(N-NK+SUN))
      PMO = SX
      RETURN
      END

C#####################################################
C  SUBROUTINE FOR FIND S(X) BY BAYESIAN DIRICHLET ESTIMATOR
C#####################################################
      SUBROUTINE BAYD(N,NU,X,T,TC,SUMT,ALPHAR,BAY)
      DIMENSION X(300)
      J = 0
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
      FAC = 1.0
      RX = 1.0
      CETA = NU/SUMT
      ALPHAT = EXP((-1)*CETA*T)
      ALPHAF = EXP((-1)*CETA*TC)
      DO 10 I = 1,N
       IF (X(I).LE.T) THEN
         IF ((X(I).EQ.T).AND.(X(I).EQ.TC)) THEN
           J = NU
           RX = (ALPHAF+(N-J)+1)/(ALPHAF+(N-J))
           FAC = RX
         ELSE
           J = I
           RX = 1.0
           FAC = RX
         ENDIF
       ENDIF
   10  CONTINUE
       BAY = ((ALPHAT+N-J)/(ALPHAR+N))*FAC
       RETURN
       END

C##########################################################
C  SUBROUTINE FOR FIND S(X) BY BAYESIAN HOMOGENEOUS ESTIMATOR
C##########################################################
       SUBROUTINE BAYH(N,NU,X,T,TC,SUMT,PML)
       DIMENSION X(100)
       J = 0
       FAC = 1.0
       RX = 1.0
       CETA = NU/SUMT
       DO 10 I = 1,N
        IF (X(I).LE.T) THEN
```

```
      IF ((X(I).EQ.T).AND.(X(I).EQ.TC)) THEN
       RX = 1.0
       J = NU
      ELSE
       RX = EXP(CETA*X(I)/((N-I+2)*(N-I+1)))*(N+2-I-1)/(N+2-I)
       J = I
      ENDIF
      FAC = FAC*RX
     ENDIF
  10 CONTINUE
     PML = EXP(-1*CETA*T/(N+1-J))*FAC
     RETURN
     END

C#############################################################
C       SUBROUTINE FOR FIND RELATIVE ERROR
C#############################################################
     SUBROUTINE RELATE(S,ST,RE)
      DIFF = S-ST
      RE =(ABS(DIFF)/S)*100
     RETURN
     END
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียน

นางสาววิริยา นิ่มนวล เกิดวันที่ 1 กันยายน พ.ศ. 2515 สำเร็จการศึกษา ปริญญาวิทยาศาสตรบัณฑิต จากภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยบูรพาในปีการศึกษา 2536 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาประกันภัย ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2537

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย