รายการอ้างอิง

ภาษาไทย

เดือน สินธุพันธ์ประทุม. เทคนิคฟอร์แทรน 77. สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, กรุงเทพมหานคร 2537.

ธีระพร วีระถาวร. การอนุมานเชิงสถิติขั้นกลาง : โครงสร้างและความหมาย. สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, กรุงเทพมหานคร 2537.

ปาริฉัตร อัครจันทโชติ. " การประมาณพารามิเตอร์ของการแจกแจงความสูญเสียสำหรับข้อมูลแบบกลุ่ม ซึ่งถูกตัดปลายทางซ้าย," (วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ จุฬาลงกรณ์มหาวิทยาลัย,2539), 129 หน้า.

ภาษาอังกฤษ

Bruce W. Turnbull. The Empirical Distribution Function with Arbitrarily Grouped, Censored and Truncated Data. Journal of the Royal Statistical Society, May , 1976.

Dick London , FSA. Survival Models and Their estimation. Actex Publications, United States of America, 1943.

I.B. Hossack, J.H. Pollard and B. Zehnwirth. Introductory Statistics with applications in general insurance. Cambridge University Press.

Jerry Banks and John S. Carson, II. Discrete-Event System Simulation. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Richard L. Burden and J. Donglas Faires. Numerical Analysis (Fourth Edition). Boston: PWS KENT Publishing Company.

Robert V. Hogg and Stuart A. Klugman . Loss Distribution. United States of America.

ภาคผนวก ก

ฟังก์ชันการแจกแจง	*Distribution function*	:	$F(x)$
ฟังก์ชันความหนาแน่น	*Probability density function*	:	$f(x)$

โมเมนต์ที่ n *Moments* : $E[X^n]$

การแจกแจงไวบูลล์ (Weibull Distribution)

Support : $x > 0$

Parameters : $c > 0\ ,\ \ \tau > 0$

D.f. : $F(x) = 1 - e^{-cx^{\tau}}$

P.d.f. : $f(x) = c\tau x^{\tau-1} e^{-cx^{\tau}}$

Moments : $E[X^n] = \dfrac{\Gamma(1+\frac{n}{\tau})}{c^{(n/\tau)}}$

Mode : $0\ ,\ \tau \le 1\ \ ;\ \ \left(\dfrac{\tau-1}{c^{\tau}}\right)^{1/\tau},\ \ \tau > 1$

Derivatives of d.f. :

$$\frac{\partial F(x)}{\partial c} = x^{\tau} e^{-cx^{\tau}}$$

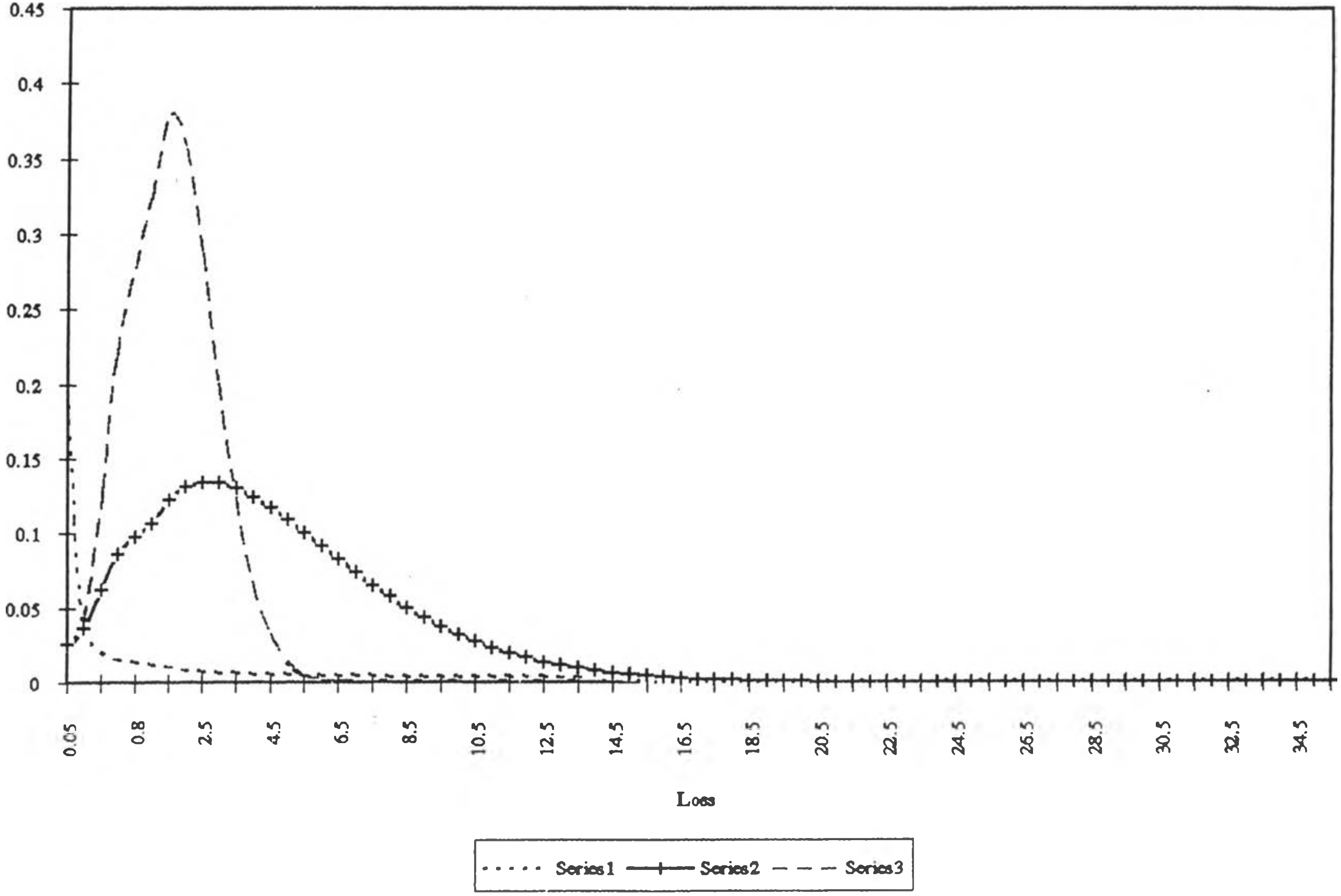
$$\frac{\partial F(x)}{\partial \tau} = cx^{\tau} e^{-cx^{\tau}} \ln x$$

ตัวอย่างฟังก์ชันการแจกแจงไวบูลล์

Parameterr			
c	τ	Mean	Graph
0.200000	1.0	5	Series 1
0.076718	1.5	5	Series 2
0.196351	2.0	5	Series 3

รูปภาพแสดงในหน้าถัดไป

Weibull Distribution



การแจกแจงลอกนอร์มอล (Lognormal Distribution)

Support : $x > 0$

Parameters : $-\infty < \mu < \infty$, $\sigma > 0$

D.f. : $$F(x) = \Phi\left(\frac{\ln x - \mu}{\sigma}\right)$$

P.d.f. : $$f(x) = \frac{\exp\left[-\frac{1}{2}\left(\frac{\ln x - \mu}{\sigma}\right)^2\right]}{x\sigma\sqrt{2\pi}}$$

Moments : $$E\left[X^n\right] = \exp\left(n\mu + \frac{1}{2}n^2\sigma^2\right)$$

Mode : $e^{\mu-\sigma}$

Derivatives of d.f. :

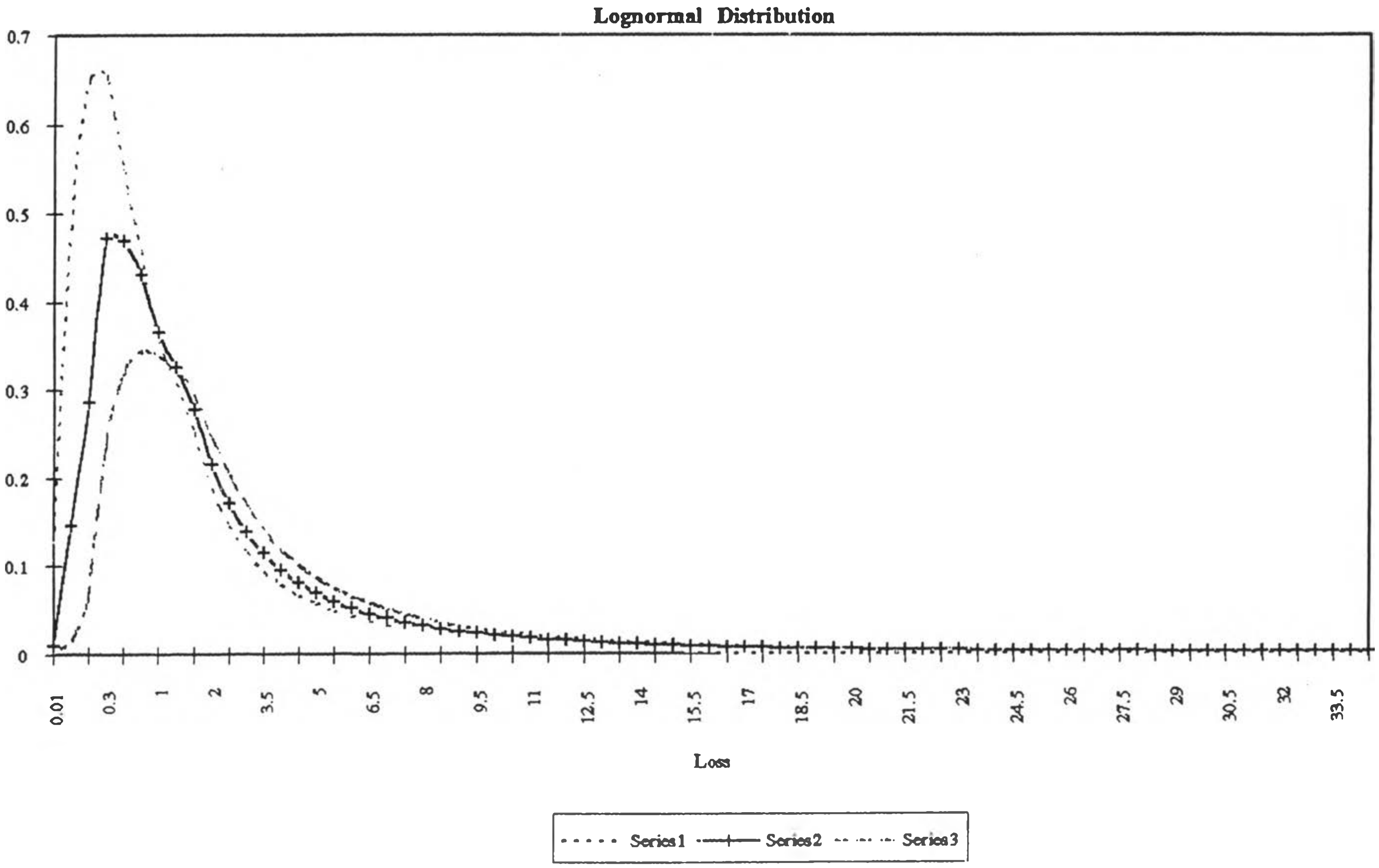
$$\frac{\partial F(x)}{\partial \mu} = -\frac{1}{\sigma}\phi\left(\frac{\ln x - \mu}{\sigma}\right)$$

$$\frac{\partial F(x)}{\partial \sigma} = -\frac{\ln x - \mu}{\sigma^2}\phi\left(\frac{\ln x - \mu}{\sigma}\right)$$

ตัวอย่างฟังก์ชั่นการแจกแจงลอกนอร์มอล

Parameter		Mean	Graph
μ	σ		
0.50	1.48959	5	Series 1
0.75	1.31106	5	Series 2
1.00	1.10403	5	Series 3

รูปภาพแสดงในหน้าถัดไป


Lognormal Distribution
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.01
0.3
1
2
3.5
5
6.5
8
9.5
11
12.5
14
15.5
17
18.5
20
21.5
23
24.5
26
27.5
29
30.5
32
33.5
Loss
Series1
Series2
Series3

ภาคผนวก ข

ความเอนเอียง (BIAS)

$\hat{\theta}$ เป็นตัวประมาณของ θ ที่เป็นตัวแปรสุ่ม (random variables) ซึ่งจะทำการเปรียบเทียบค่า $E[\hat{\theta}]$ กับ θ

ถ้า

$E[\hat{\theta}] = \theta$ ....(ข.1) จะได้ว่า $\hat{\theta}$ เป็นความไม่เอนเอียงของ θ (unbiased of θ)

ถ้า

$E[\hat{\theta}] > \theta$ ....(ข.2) จะได้ว่า $\hat{\theta}$ เป็นความเอนเอียงบวกของ θ (positively biased)

ถ้า

$E[\hat{\theta}] < \theta$ ....(ข.3) จะได้ว่า $\hat{\theta}$ เป็นความเอนเอียงลบของ θ (negatively biased)

โดยทั่ว ๆ ไปแล้วความเอนเอียงของ $\hat{\theta}$ จะได้ว่า

$$Bias(\hat{\theta}) = E[\hat{\theta}] - \theta \quad \text{.....................(ข.4)}$$

กรณีที่ $E[\hat{\theta}]$ เป็นฟังก์ชันของขนาดตัวอย่าง n ที่ทำการประมาณค่า θ ในลักษณะที่ $E[\hat{\theta}]$ ไม่เท่ากับ θ จะเรียกว่า $\hat{\theta}$ เอนเอียงจริง (not strictly unbiased)

อย่างไรก็ตาม ถ้า $\lim_{n \to \infty} E[\hat{\theta}] = \theta$ ....(ข.5)

จะเรียกว่า $\hat{\theta}$ เป็นตัวประมาณที่ไม่เอนเอียงเมื่อใกล้อนันต์ (asymptotically unbiased)

ความคลาดเคลื่อนกำลังสองเฉลี่ย (MEAN SQUARE ERROR AND VARIANCE)

$\hat{\theta}$ เป็นตัวแปรสุ่มขณะที่ θ เป็นค่าคงที่ ซึ่งค่า $\hat{\theta}$ เป็นค่าที่ไม่ทราบค่า เรากำหนดค่าคาดหวังความแตกต่างกำลังสองระหว่าง $\hat{\theta}$ และ θ เป็นความคลาดเคลื่อนกำลังสองเฉลี่ยของตัวประมาณ $\hat{\theta}$ นั่นคือ

$$MSE(\hat{\theta}) = E[(\hat{\theta}-\theta)^2] \quad \text{.....................(ข.6)}$$

โดยกำหนด

$$Var(\hat{\theta}) = E[(\hat{\theta}-E[\hat{\theta}])^2] \quad \text{.............(ข.7)}$$

ในการวัดค่าระหว่างความคลาดเคลื่อนกำลังสองเฉลี่ย และความแปรปรวนของ $\hat{\theta}$ เริ่มจาก (ข.6) โดยการบวกและลบค่าด้วย $E[\hat{\theta}]$

$$\begin{aligned} MSE(\hat{\theta}) &= E[(\hat{\theta}-E[\hat{\theta}]+E[\hat{\theta}]-\theta)^2] \\ &= E[(\hat{\theta}-E[\hat{\theta}])^2+2(E[\hat{\theta}]-\theta)(\hat{\theta}-E[\hat{\theta}])+(E[\hat{\theta}]-\theta)^2] \\ &= E[(\hat{\theta}-E[\hat{\theta}])^2]+2(E[\hat{\theta}]-\theta)E[\hat{\theta}-E[\hat{\theta}])+E[(E[\hat{\theta}]-\theta)^2] \end{aligned}$$

เมื่อ $2(E[\hat{\theta}]-\theta)$ เป็นค่าคงที่

$$E[\hat{\theta}-E[\hat{\theta}]] = E[\hat{\theta}]-E[E[\hat{\theta}]] = 0$$

เมื่อ $E[E[\hat{\theta}]] = E[\hat{\theta}]$ ดังนั้น จะได้ว่า

$$\begin{aligned} MSE(\hat{\theta}) &= E[(\hat{\theta}-E[\hat{\theta}]+E[(E[\hat{\theta}]-\theta)^2] \\ &= E[(\hat{\theta}-E[\hat{\theta}])^2]+(E[\hat{\theta}]-\theta)^2 \quad \text{.......................(ข.8)} \end{aligned}$$

เมื่อ $(E[\hat{\theta}]-\theta)$ เป็นค่าคงที่ ใช้ (ข.7) และ (ข.4) จาก (ข.8) จะได้

$$\begin{aligned} MSE(\hat{\theta}) &= Var(\hat{\theta})+[Bias(\hat{\theta})] \quad \text{..........................(ข.9)} \\ &= Var(\hat{\theta})+[E[\hat{\theta}]-\theta]^2 \end{aligned}$$

ตัวประมาณ $\hat{\theta}$ เป็นตัวประมาณที่มีความคลาดเคลื่อนกำลังสองเฉลี่ยต่ำสุด (MSE) ถ้า $\hat{\theta}$ มีค่า $E(\hat{\theta}-\theta)^2$ ต่ำสุด หรือ

$$MSE \approx \frac{\Sigma(\hat{\theta}-\theta)^2}{n} \qquad \text{.................................(ข.10)}$$

โดยที่ $\hat{\theta}$ เป็นตัวประมาณของพารามิเตอร์ θ

ภาคผนวก ค

วิธีมอนติคาร์โล (Monte Carlo method)

เทคนิคในการจำลองตัวแบบทางคณิตศาสตร์มีอยู่หลายวิธี วิธีมอนติคาร์โลเป็นวิธีหนึ่งที่นิยมใช้กันอย่างแพร่หลายในปัจจุบัน ซึ่งหลักการของวิธีมอนติคาร์โลนั้นเป็นการจำลองตัวเลขสุ่ม (Random Number) มาช่วยในการหาคำตอบของปัญหาที่ต้องการศึกษา

ในการวิจัยครั้งนี้จะใช้วิธีการมอนติคาร์โลดังกล่าวในการสร้างข้อมูลที่มีสภาพการแจกแจงตามที่ต้องการ ซึ่งขั้นตอนของวิธีมอนติคาร์โลที่ใช้กันอยู่ในปัจจุบันเป็น 3 ขั้นตอนดังนี้

1. การสร้างตัวเลขสุ่ม การใช้ตัวเลขสุ่มสำคัญมากในวิธีมอนติคาร์โล เพราะว่าหลักการของวิธีมอนติคาร์โลนั้นจะใช้ตัวเลขมาช่วยในการหาคำตอบของปัญหา ซึ่งลักษณะตัวเลขสุ่มจะมีการแจกแจงแบบสม่ำเสมอ (Uniform distribution) ในช่วง (0,1) สำหรับวิธีการสร้างตัวเลขสุ่มนั้นมีผู้เสนอไว้หลายวิธี แต่วิธีหนึ่งที่ดีคือวิธีที่ไวท์และชมิดท์ (White and Schemidt 1975: 421) เสนอไว้กล่าวคือลักษณะของตัวเลขสุ่มที่เกิดขึ้นมีการแจกแจงแบบสม่ำเสมอ (0,1) และเป็นอิสระซึ่งกันและกัน

วิธีการผลิตเลขสุ่มแบบ Linear Congruential Method จะผลิตชุดตัวเลขสุ่มจำนวนเต็ม $x_1, x_2, ...$ มีค่าอยู่ระหว่าง 0 ถึง M-1 จากสมการตัวผลิต

$$x_1 = (ax_{i-1} + c) \bmod M \qquad ; i = 1,2,...$$

ตัวเลขจำนวนเต็ม $x_1, x_2, ...$ จะมีการแจกแจงสม่ำเสมอ U(0,M-1) เพราะฉะนั้น ตัวเลขสุ่ม $R_1, R_2, ...$ มีการแจกแจงสม่ำเสมอ U(0,1) ผลิตได้จากสมการ

$$R_i = \frac{x_i}{M} \qquad ; \ i = 1,\ 2,...$$

กำหนดให้ a เป็นค่าคงที่

c เป็นค่าส่วนที่เพิ่ม (increment)

x_0 เป็นตัวเลขนำ

M เป็น Modulus

mod หมายความว่า $(ax_{i-1} + c)$ หารด้วย M จนกระทั่งเหลือเศษน้อยกว่าค่า M เลขที่เหลือจึงเป็นเลขสุ่มของเลขสุ่มตัวถัดไปคือ x_i

ถ้ากำหนดค่า c<>0 เรียกตัวผลิตว่า Mixed Congruential Method แต่ถ้ากำหนด c=0 เรียกตัวผลิตนี้ว่า Multiplicative Congruential Methhod การกำหนดค่า c, a, M และ x0 มีความสำคัญมากเนื่องมาจากค่าเหล่านี้มีผลโดยตรงต่อคุณสมบัติทางสถิติ และความยาวของชุดตัวเลขสุ่มจาก สูตร $R_i = \frac{x_i}{M}$ จะได้ว่า R_i มีค่าอยู่ในเซตของ $\left\{0, \frac{1}{M}, \frac{2}{M}, \ldots, \frac{(M-1)}{M}\right\}$ ทั้งนี้เพราะว่าค่าของ x_i เป็นเลขจำนวนอยู่ในเซต $\{0,1,2,\ldots,(M-1)\}$ เพราะฉะนั้นค่า R_i มีค่าไม่ต่อเนื่อง แทนที่จะเป็นค่าต่อเนื่องที่มีการแจกแจงสม่ำเสมอ U(0,1) อย่างไรก็ตามจะประมาณค่าต่อเนื่องได้ โดยการกำหนดค่า M ให้มีขนาดใหญ่มาก ๆ จะมีผลทำให้ช่องว่าง R_i ที่มีความต่อเนื่องโดยประมาณ ลักษณะการกระทำดังกล่าวเป็นการสร้างความหนาแน่น (Density) ในกลุ่มตัวเลขสุ่มให้มีความหนาแน่นสูงในช่วง[0,1] และเพื่อหลีกเลี่ยงชุดตัวเลขสุ่มซ้ำในการใช้งานครั้งหนึ่ง ๆ ตัวผลิตควรมีความยาวของชุดตัวเลขสุ่มมากที่สุดเท่าที่จะเป็นไปได้

การกำหนดค่า c, a, M และx_0 มีความสำคัญมาก เนื่องมาจากมีผลโดยตรงต่อคุณสมบัติทางสถิติ และความยาวของชุดตัวเลขสุ่ม ตัวผลิตเลขสุ่มที่ได้ผ่านการทดสอบแล้วเป็นอย่างมากคือ Multiplication Congruential Method ที่กำนด c=0 และกำหนด $a = 7^5 = 16807$ การกำหนดค่า M ให้มีขนาดใหญ่มาก ๆ และเป็นเลขคี่ที่สามารถคำนวณได้จากเครื่องคอมพิวเตอร์ โดยที่ $M = 2^b$ เมื่อ b เป็นค่าความยาว 1 word หรือจำนวน bit ใน 1 word ของเครื่องคอมพิวเตอร์ 32 bit ซึ่ง bit สุดท้าย 1 bit ใช้สำหรับแสดงเครื่องหมาย ดังนั้นเลขจำนวนเต็มที่ใหญ่ที่สุดใน 1 word และเป็นเลขคี่ที่คอมพิวเตอร์ได้รับคือ $2^{b-1} - 1$ เท่ากับ $2^{31-1} - 1 = 2147483647$ นั่นคือค่า M ควรมีค่าเท่ากับ 2147483647

จากที่กล่าวมาทั้งหมดข้างต้น สามารถนำมาเขียนโปรแกรมภาษาฟอร์แทรน ฟังก์ชันที่ใช้คือ RAND (IU) ซึ่ง IU คือเลขสุ่มที่เป็นค่าเริ่มต้นที่เข้าไปในโปรแกรมย่อย ลักษณะโปรแกรมที่ใช้มีดังนี้

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

2. การประยุกต์ปัญหาที่ต้องการศึกษาเพื่อใช้กับตัวเลขสุ่มซึ่งขั้นตอนนี้ขึ้นอยู่กับลักษณะของปัญหาที่ต้องการศึกษา บางปัญหาอาจจะไม่ได้ใช้ตัวเลขสุ่มโดยตรง แต่อาจมีขั้นตอนอื่นอีกหลายขั้นตอน ซึ่งบางขั้นตอนต้องใช้ตัวเลขสุ่ม

3. การทดลองกระทำซ้ำ (Replication) เมื่อประยุกต์ปัญหาให้ใช้กับตัวเลขสุ่มได้แล้ว ขั้นตอนต่อไปก็คือ การทดลองโดยใช้กระบวนการของการสุ่ม (Random Process) มากระทำในลักษณะที่ซ้ำ ๆ กัน เพื่อหาคำตอบของปัญหาที่ต้องการศึกษา

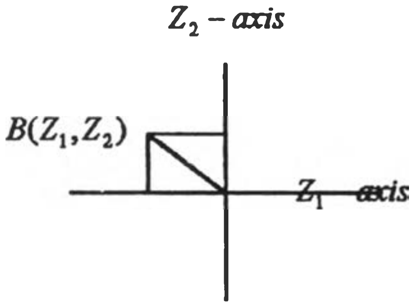
การผลิตเลขสุ่มที่มีการแจกแจงลอกนอร์มอล

การแจกแจงปกติ

การแจกแจงแบบปกติโดยใช้เทคนิคการแปลงโดยตรงจาก

$$f(x) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2p}} e^{-\frac{x^2}{2}} dx \quad ; \quad -\infty < x < \infty$$

Box และ Muller สร้างเลขสุ่มที่มีการแจกแจงแบบปกติมาตรฐาน ที่มีค่าเฉลี่ยเป็น 0 และค่าแปรปรวนเป็น 1 พร้อม ๆ กัน 2 ค่า ดังนี้



$Z_1 = B\cos\theta$

$Z_2 = B\sin\theta$

$B^2 = Z_1^2 + Z_2^2$ มีการแจกแจงไคสแคว์ (Chi-Square distribution) ด้วยระดับความเป็นอิสระ = 2 ซึ่งเทียบเท่า(equivalent) กับการแจกแจงเอ็กซ์โปเนนเชียล (exponential distribution) ด้วยค่าเฉลี่ย =2 ดังนั้นรัศมี B มีค่าดังนี้ $B = (-2\ln R)^{\frac{1}{2}}$ โดยการสมมาตร (Symmetry) ของการแจกแจงแบบปกติ (normal distribution) θ มีการแจกแจงสม่ำเสมอ (uniform distribution) ระหว่าง 0 กับ 2π เรเดียน ซึ่งค่า B และ θ เป็น Multually Independent

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

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

จากคุณสมบัติของการแจกแจงลอกนอร์มอล (logmormal Distribution)

กำหนด $EX1 \sim N(\mu,\sigma^2)$
$W1 = e^{EX1} \sim LN(\mu,\sigma^2)$

ขั้นตอนในการสร้าง

1. ผลิตเลขสุ่ม Y ~ N นั่นคือ ตัวแปรสุ่ม Y มีการแจกแจงแบบปกติ มีค่าเฉลี่ยเท่ากับ μ และค่าความแปรปรวนเท่ากับ σ^2

1.1 ผลิตเลขสุ่ม 2 ค่า คือ Z_1 และ Z_2 จากการแจกแจงสม่ำเสมอช่วง 0 ถึง 1 ให้

$$ZONE = (-2\ln Z_2)^{\frac{1}{2}}\cos(2\pi Z_1)$$

$$ZTWO = (-2\ln Z_2)^{\frac{1}{2}}\sin(2\pi Z_1)$$

ซึ่ง ZONE และ ZTWO คือ เลขสุ่มที่มีการแจกแจงแบบปกติ มีค่าเฉลี่ย เท่ากับ 0 และค่าความแปรปรวนเท่ากับ 1

โปรแกรมย่อยสำหรับจำลองแบบประชากรที่มีการแจกแจงเป็นปกติ มีค่าเฉลี่ย และค่าเบี่ยงเบนมาตรฐาน จะเรียกใช้ SUBROUTINE RNOR(EX1) ซึ่งจะได้

$$EX1 = ZONE * SIGMA + RMEW$$
$$EX1 = ZTWO * SIGMA + RMEW$$

2. ผลิตเลขสุ่มที่มีการแจกแจงลอกนอร์มอล

$$W1 = e^{EX1}$$

3. กระทำซ้ำข้อ 1. ถึงข้อ 2. จนกระทั่งได้จำนวนข้อมูลครบตามที่กำหนดในการตัดปลายทางขวาและซ้าย

4. ทำการเรียงข้อมูลจากน้อยไปหามาก

คำสั่งในโปรแกรมย่อยเพื่อทำการผลิตเลขสุ่มที่มีการแจกแจงลอกนอร์มอล โดยกำหนดจำนวนเลขสุ่มที่ต้องการเท่ากับ N

```
      SUBROUTINE  LOGNOR
      COMMON/SEED/IU,N,KK
     *      /PAR/RMEW,SIGMA,XMUE,XVAR
     *      /SON/X(1000),XBAR(200)
```

```
*       /PAR1/XD,XW,K
*       /SS/IR,CC
          I2=0
          I3=0
      DO 10  I=1,N
5        W1=RAND(IU)
        IF((W1.LE.0.).OR.(W1.GE.1.)) GOTO 5
          CALL  RNOR(EX1)
         U1=EXP(EX1)
        IF(U1.LE.XD) GOTO 5
          G=N-CC
        IF((U1.GT.XD).AND.(U1.LE.XW)) THEN
              I2=I2+1
          IF (I2.LE.G)  THEN
              X(I)=U1
           ELSE
             GOTO 5
          END IF
         END IF
        IF((U1.GT.XW).AND.(I3.LE.CC)) THEN
             I3=I3+1
          IF(I3.LE.CC) THEN
            X(I)=XW
           ELSE
            GOTO 5
          END IF
        END IF
10     CONTINUE
         CALL SSORT(N)
         RETURN
      END
```

```
      SUBROUTINE  RNOR(EX1)
      COMMON/SEED/IU,N,KK
     *      /PAR/RMEW,SIGMA,XMUE,XVAR
     *      /SON/X(1000),XBAR(200)
         PI=3.1415926
      IF(KK.EQ.1) GOTO 100
         Z1=RAND(IU)
         Z2=RAND(IU)
      ZONE=SQRT(-2*ALOG(Z1))*COS(2*PI*Z2)
      ZTWO=SQRT(-2*ALOG(Z1))*SIN(2*PI*Z2)
          EX1=ZONE*SIGMA+RMEW
          KK=1
          GOTO 200
100   EX1=ZTWO*SIGMA+RMEW
          KK=0
200     RETURN
      END
```

การผลิตเลขสุ่มที่มีการแจกแจงไวบูลล์

จากฟังก์ชันสะสมสามารถแปลงผกผัน (Inverted) อยู่ในรูปของ

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

ขั้นตอนในการสร้าง

1. ผลิตเลขสุ่ม R U(0,1)
2. ให้ $X = \left[\frac{-\ln(1-R)}{c}\right]^{1/\tau}$

3. กระทำซ้ำข้อ 1. ถึงข้อ 2. จนกระทั่งได้จำนวนข้อมูลครบตามที่กำหนดในการตัดปลายทางขวาและซ้าย

4. ทำการเรียงข้อมูลจากน้อยไปหามาก

คำสั่งในโปรแกรมย่อยเพื่อทำการผลิตเลขสุ่มที่มีการแจกแจงไวบูลล์ โดยกำหนดจำนวนเลขสุ่มที่ต้องการเท่ากับ N

```
      SUBROUTINE  WEI
      COMMON/SEED/IU,N
     *      /PAR/T,CT,PT,PCT
     *      /SON/X(1000),XBAR(200)
     *      /PAR1/XD,XW,K
     *      /SS/IR,CC
          I2=0
          I3=0
      DO 10  I=1,N
5       W1=RAND(IU)
      IF((W1.LE.0.).OR.(W1.GE.1.)) GOTO 5
        XX=(-ALOG(1.-W1)/CT)**(1./T)
       IF (XX.LE.XD) GOTO 5
            G=N-CC
       IF((XX.GT.XD).AND.(XX.LE.XW)) THEN
                I2=I2+1
            IF (I2.LE.G)  THEN
                 X(I)=XX
             ELSE
                  GOTO 5
            END IF
        END IF
        IF((XX.GT.XW).AND.(I3.LE.CC)) THEN
             I3=I3+1
           IF (I3.LE.CC)  THEN
               X(I)=XW
```

```
      ELSE
        GOTO 5
      END IF
    END IF
10  CONTINUE
      CALL SSORT(N)
      RETURN
    END
```

ภาคผนวก ง

```
C*************************************************************************C
C******************************** MAIN PROGRAM ***********************C
C*************************************************************************C
COMMON/SEED/IU,N,KK
     *     /PAR/RMEW,SIGMA,XMUE,XVAR
     *     /SON/X(1000),XBAR(200) /SAM/C(100),FQ(100),F(100)
     *     /SAD/EFQ(100) /SED/SUMX(200)
     *     /SSD/XEW(1000),XIG(1000),X1C(1000),X2C(1000)
     *     /PAR1/XD,XW,K
     *     /SSU/SUMM,SUMS,SUMD1,SUMD2
     *     /SIT/DIVG /SS/IR,CC /FIN/HOP1,HOP2,SUC1,SUC2
   CHARACTER DIVG*1
   N=100
   K=10
   KK=0
   MAX=200
   SIGMA=1.31106
   RMEW=0.75
   XD=0.2
   XW=30.0
   IPCUT=10
   CC=(N*IPCUT)/100
          IU=13
          DIVG='N'
          IR=1000
          LP=500
          SUMM=0.
          SUMS=0.
          SUMD1=0.
          SUMD2=0.
    WRITE(6,21)
21  FORMAT(2X,'####### LOGNORMAL OF MAXIMUM LIKELYHOOD #######')
   WRITE(6,19)XD,XW
```

```
19    FORMAT(2X,'DEDUCT=',F10.5,2X,'REIN=',F10.5)
      WRITE(6,25) RMEW,SIGMA
25    FORMAT(2X,' FRAR1=',F8.5,4X,'FPAR2=',F8.5)
      WRITE(6,20) N,K,IPCUT,IR
20    FORMAT(2X,'    N=',I4,7X,'    K=',I3,2X,'PERCUT=',I3,2X,'IR=',I4 /60('-'))
      DO 500 IZ=1,IR
           LOP=1
C            WRITE(6,30)IZ
C30           FORMAT(2X,'IR=',I3)
101     CALL  LOGNOR
         DIVG='N'
        CALL  GROUP
        CALL  RMOM
       IF(XVAR.LE.0) GOTO 101
        CALL  CRAM(IZ,MAX)
         IF(DIVG.EQ.'Y') THEN
          LOP=LOP+1
          IF(LOP.GE.LP) THEN
             WRITE(6,35)
35            FORMAT(2X,'****** ERROR PROGRAM *****')
             GOTO 70
            ELSE
                 GOTO 101
          END IF
         END IF
      IF(IZ.EQ.IR) THEN
       CALL VALUE (IZ)
       GOTO 70
      END IF
500     CONTINUE
70     STOP
      END
C****************************************************C
C**********   GEN DATA OF LOGNORMAL DIST. ********C
C****************************************************C
      SUBROUTINE  LOGNOR
      COMMON/SEED/IU,N,KK
```

```
     *     /PAR/RMEW,SIGMA,XMUE,XVAR
     *     /SON/X(1000),XBAR(200)
     *     /PAR1/XD,XW,K
     *     /SS/IR,CC
             I2=0
             I3=0
        DO 10  I=1,N
5         W1=RAND(IU)
          IF((W1.LE.0.).OR.(W1.GE.1.)) GOTO 5
             CALL  RNOR(EX1)
           U1=EXP(EX1)
          IF(U1.LE.XD) GOTO 5
             G=N-CC
          IF((U1.GT.XD).AND.(U1.LE.XW)) THEN
                 I2=I2+1
             IF (I2.LE.G)  THEN
                  X(I)=U1
              ELSE
                 GOTO 5
             END IF
           END IF
          IF((U1.GT.XW).AND.(I3.LE.CC)) THEN
                I3=I3+1
             IF(I3.LE.CC) THEN
                X(I)=XW
               ELSE
                GOTO 5
             END IF
          END IF
10      CONTINUE
            CALL SSORT(N)
            RETURN
        END
C*****************************************************C
C************** FUNCTION  RANDOM *****************C
C*****************************************************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
C************* NORMAL DISTRIBUTION ****************C
C*****************************************************C
    SUBROUTINE RNOR(EX1)
    COMMON/SEED/IU,N,KK
   *     /PAR/RMEW,SIGMA,XMUE,XVAR
   *     /SON/X(1000),XBAR(200)
      PI=3.1415926
    IF(KK.EQ.1) GOTO 100
      Z1=RAND(IU)
      Z2=RAND(IU)
    ZONE=SQRT(-2*ALOG(Z1))*COS(2*PI*Z2)
    ZTWO=SQRT(-2*ALOG(Z1))*SIN(2*PI*Z2)
       EX1=ZONE*SIGMA+RMEW
       KK=1
       GOTO 200
100  EX1=ZTWO*SIGMA+RMEW
       KK=0
200    RETURN
    END
C*****************************************************C
C************** SHELLSORT DATA ********************C
C*****************************************************C
    SUBROUTINE SSORT(N)
    COMMON/SON/X(1000),XBAR(200)
       JUMP=N
25      JUMP=JUMP/2
      IF (JUMP.EQ.0) GOTO 99
         J2=N-JUMP
        DO 40 J=1,J2
         I=J
```

```
30        J3=I+JUMP
       IF (X(I).LE.X(J3))  GOTO 40
         CALL SWAP(X(I),X(J3))
       I=I-JUMP
       IF (I.GT.0) GOTO 30
40    CONTINUE
      GOTO 25
99    RETURN
      END
C*************** SWAP ***************************C
      SUBROUTINE  SWAP(AM,AN)
        HOLD=AM
         AM =AN
         AN=HOLD
        RETURN
      END
C**************************************************C
C********** CUMMULATIVE LOGNORMAL DIST.  ******C
C**************************************************C
      FUNCTION  CDF(XD,XW,XM,XS,CCI)
        X1=(ALOG(CCI)-XM)/XS
        X2=(ALOG(XD)-XM)/XS
        X3=(ALOG(XW)-XM)/XS
        X4=PHIB(X3)-PHIB(X2)
       IF ((X2.GT.7).OR.(X4.EQ.0)) THEN
                CDF=0.
         ELSE
              A=PHIB(X3)-PHIB(X2)
             CDF=(PHIB(X1)-PHIB(X2))/A
        END IF
        RETURN
      END
C***************************************************C
C****** PDF. OF Y WHEN Y IS STANDARD NORMAL ****C
C***************************************************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
C***** CDF. OF ZZ WHEN ZZ IS STANDARD NORMAL  ****C
C*****************************************************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))
            A=0.4361836
            B=0.1201676
            C=0.937298
            PROB=1.-R*(A*WW-B*(WW**2)+C*(WW**3))
         END IF
       IF (ZZ.GE.0.) THEN
           PHIB=PROB
       ELSE
           PHIB=1.-PROB
       END IF
      RETURN
     END
C*****************************************************C
C**************** GROUP  DATA  *********************C
C*****************************************************C
     SUBROUTINE GROUP
     COMMON/SEED/IU,N,KK
    *     /PAR/RMEW,SIGMA,XMUE,XVAR
    *     /SON/X(1000),XBAR(200)
```

```
     *     /SAM/C(100),FQ(100),F(100)
     *     /SAD/BFQ(100)
     *     /SED/SUMX(200)
     *     /PAR1/XD,XW,K
      XIN=(XW-XD)/K
      DO 25 I=1,K
        C(I)=XD+(I*XIN)
25    CONTINUE
C##################
        DO 77 I=1,K
         J=I-1
        CCI=C(I)
          IF(J.EQ.0) THEN
              CCJ=XD
            ELSE
              CCJ=C(J)
          END IF
77      CONTINUE
C##################
         NUM=1
      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(J)=0.
31         IJ=IK
        IF(IJ.GT.N) GOTO 54
        IF(X(IJ).LE.C(J)) THEN
```

```
      SUMX(J)=SUMX(J)+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(J)/FQ(J)
     END IF
32   CONTINUE
        F(1)=FQ(1)/N
     DO 55  I=2,K
        J=I-1
        F(I)=FQ(I)/N+F(J)
55      CONTINUE
     RETURN
    END
C***************************************************C
C********* INITIAL VALUE BY MOMENT METHOD  ******C
C***************************************************C
    SUBROUTINE  RMOM
    COMMON/SEED/IU,N,KK
    *    /SON/X(1000),XBAR(200)
    *    /SAM/C(100),FQ(100),F(100)
    *    /SAD/EFQ(100)
    *    /PAR/RMEW,SIGMA,XMUE,XVAR
    *    /PAR1/XD,XW,K
      SUMX1=0.
      SUMX2=0.
     DO 601 I=1,K
     SUMX1=SUMX1+(FQ(I)*XBAR(I))
     SUMX2=SUMX2+(FQ(I)*((XBAR(I))**2))
601  CONTINUE
      EX1=SUMX1/N
      EX2=SUMX2/N
      XVAR=ALOG(EX2)-2*ALOG(EX1)
```

```
      XMUE=ALOG(EX1)-(XVAR/2)
     RETURN
    END
C********************************************************C
C********* MAXIMUM LIKELIHOOD OF LOGNORMAL ******C
C********************************************************C
    SUBROUTINE MLI(I,XM,XS,XG1,XG11,XG12,XG2,XG22)
    COMMON/SON/X(1000),XBAR(200)
   *     /SAM/C(100),FQ(100),F(100)
   *     /PAR/RMEW,SIGMA,XMUE,XVAR
   *     /PAR1/XD,XW,K
   *     /SIT/DIVG
    CHARACTER DIVG*1
          J=I-1
        CCI=C(I)
      IF(J.EQ.0) THEN
         CCJ=XD
       ELSE
         CCJ=C(J)
      END IF
        X2=(ALOG(CCI)-XM)/XS
        X1=(ALOG(CCJ)-XM)/XS
       XXD=(ALOG(XD)-XM)/XS
       XXW=(ALOG(XW)-XM)/XS
      IF(XXD.GT.7.) THEN
         XP=0.
          DIVG='Y'
           GOTO 22
      ELSE
         IF(XXD.LT.-7.) THEN
             XP=PHIB(XXW)
          ELSE
             PH1=PHIB(XXW)
             PH2=PHIB(XXD)
            XP=PH1-PH2
         END IF
      END IF
```

```
      PH3=PHIB(X2)
      PH4=PHIB(X1)
      P=PH3-PH4
           AQ=PHIL(X1)
           AW=PHIL(X2)
           AE=PHIL(XXD)
           AY=PHIL(XXW)
       P1=(AQ-AW)/XS
      XP1=(AE-AY)/XS
       P2=(X1*AQ-X2*AW)/XS
      XP2=(XXD*AE-XXW*AY)/XS
      P11=P2/(XS**3)
     XP11=XP2/(XS**3)
      P12=AW*(1.-(X2**2))/(XS**2)
     *     - AQ*(1.-(X1**2))/(XS**2)
     XP12=AY*(1.-(XXW**2))/(XS**2)
     *     - AE*(1.-(XXD**2))/(XS**2)
      P22=X2*(2.-(X2**2))*AW/(XS**2)
     *    -X1*(2.-(X1**2))*AQ/(XS**2)
     XP22=XXW*(2.-(XXW**2))*AY/(XS**2)
     *    -XXD*(2.-(XXD**2))*AE/(XS**2)
     A1=P1/P
     B1=XP1/XP
     C1=A1-B1
      A2=P2/P
      B2=XP2/XP
      C2=A2-B2
     A11=((P*P11)-(P1**2))/(P**2)
     B11=((XP*XP11)-(XP1**2))/(XP**2)
     C11=A11-B11
      A12=((P*P12)-(P1*P2))/(P**2)
      B12=((XP*XP12)-(XP1*XP2))/(XP**2)
      C12=A12-B12
     A22=((P*P22)-(P2**2))/(P**2)
     B22=((XP*XP22)-(XP2**2))/(XP**2)
     C22=A22-B22
           XG1=C1*FQ(I)
```

```
          XG2=C2*FQ(I)
          XG11=C11*FQ(I)
          XG12=C12*FQ(I)
          XG22=C22*FQ(I)
22    RETURN
      END
C*******************************************************C
C******* LEAST SQUARE METHOD OF LOGNORMAL *********C
C*******************************************************C
      SUBROUTINE  LSL(I,XM,XS,XG1,XG11,XG12,XG2,XG22)
      COMMON/SON/X(1000),XBAR(200)
     *      /SAM/C(100),FQ(100),F(100)
     *      /PAR/RMEW,SIGMA,XMUE,XVAR
     *      /PAR1/XD,XW,K
     *      /SIT/DIVG
     *      /SEED/IU,N,KK
      CHARACTER  DIVG*1
          J=I-1
         CCI=C(I)
        IF(J.EQ.0) THEN
          CCJ=XD
        ELSE
          CCJ=C(J)
        END IF
         X2=(ALOG(CCI)-XM)/XS
         X1=(ALOG(CCJ)-XM)/XS
        XXD=(ALOG(XD)-XM)/XS
        XXW=(ALOG(XW)-XM)/XS
       IF(XXD.GT.7.) THEN
          XP=0.
          DIVG='Y'
          GOTO  22
        ELSE
          IF(XXD.LT.-7.) THEN
            XP=PHIB(XXW)
          ELSE
           PH1=PHIB(XXW)
```

```
          PH2=PHIB(XXD)
          XP=PH1-PH2
        END IF
      END IF
       PH3=PHIB(X2)
       PH4=PHIB(X1)
      P=PH3-PH4
          AQ=PHIL(X1)
          AW=PHIL(X2)
          AB=PHIL(XXD)
          AR=PHIL(XXW)
       P1=(AQ-AW)/XS
       XP1=(AB-AR)/XS
       P2=(X1*AQ-X2*AW)/XS
       XP2=(XXD*AB-XXW*AR)/XS
       P11=P2/(XS**3)
      XP11=XP2/(XS**3)
       P12=AW*(1.-(X2**2))/(XS**2)
*         -AQ*(1.-(X1**2))/(XS**2)
     XP12=AR*(1.-(XXW**2))/(XS**2)
*         -AB*(1.-(XXD**2))/(XS**2)
    P22=X2*(2.-(X2**2))*AW/(XS**2)
*    -X1*(2.-(X1**2))*AQ/(XS**2)
   XP22=XXW*(2.-(XXW**2))*AR/(XS**2)
*    -XXD*(2.-(XXD**2))*AB/(XS**2)
        Z=N*(P/XP)
     Z1=((XP*P1)-(P*XP1))/(XP**2)
     Z2=((XP*P2)-(P*XP2))/(XP**2)
    Z11=((XP*P11)-(P*XP11))/(XP**2)
*     -(2*XP1*((XP*P1)-(P*XP1)))/(XP**3)
    Z12=((XP2*P1)+(XP*P12)-(P2*XP1)-(P*XP12))/(XP**2)
*     -(2*XP2*((XP*P1)-(P*XP1)))/(XP**3)
    Z22=((XP*P22)-(P*XP22))/(XP**2)
*     -(2*XP2*((XP*P2)-(P*XP2)))/(XP**3)
          AB=Z-FQ(I)
      XG1=2*N*AB*Z1
      XG11=2*N*((AB*Z11)+(N*(Z1**2)))
```

```
      XG12=2*N*((AB*Z12)+(N*Z1*Z2))
      XG22=2*N*((AB*Z22)+(N*(Z2**2)))
      XG2=2*N*AB*Z2
22    RETURN
      END
C*********************************************************C
C****  MINIMUM CHI-SQUARE  METHOD  OF LOGNORMAL  ***C
C*********************************************************C
      SUBROUTINE  MC(I,XM,XS,XG1,XG11,XG12,XG2,XG22)
      COMMON/SON/X(1000),XBAR(200)
     *      /SAM/C(100),FQ(100),F(100)
     *      /PAR/RMEW,SIGMA,XMUE,XVAR
     *      /PAR1/XD,XW,K
     *      /SIT/DIVG
     *      /FIN/HOP1,HOP2,SUC1,SUC2
     *      /SEED/IU,N,KK
      CHARACTER  DIVG*1
          J=I-1
         CCI=C(I)
        IF(J.EQ.0) THEN
          CCJ=XD
        ELSE
          CCJ=C(J)
        END IF
         X2=(ALOG(CCI)-XM)/XS
         X1=(ALOG(CCJ)-XM)/XS
        XXD=(ALOG(XD)-XM)/XS
        XXW=(ALOG(XW)-XM)/XS
         X3=PHIB(XXW)-PHIB(XXD)
        IF(XXD.GT.7.) THEN
          XP=0.
          DIVG='Y'
          GOTO  22
        ELSE
          IF(XXD.LT.-7.) THEN
            XP=PHIB(XXW)
          ELSE
```

```
          XP=X3
        END IF
      END IF
       P=PHIB(X2)-PHIB(X1)
         AQ=PHIL(X1)
         AW=PHIL(X2)
         AE=PHIL(XXD)
         AR=PHIL(XXW)
        P1=(AQ-AW)/XS
       XP1=(AE-AR)/XS
        P2=(X1*AQ-X2*AW)/XS
       XP2=(XXD*AE-XXW*AR)/XS
       P11=P2/(XS**3)
      XP11=XP2/(XS**3)
       P12=AW*(1.-(X2**2))/(XS**2)
     *      -AQ*(1.-(X1**2))/(XS**2)
     XP12=AR*(1.-(XXW**2))/(XS**2)
     *      -AE*(1.-(XXD**2))/(XS**2)
    P22=X2*(2.-(X2**2))*AW/(XS**2)
     *  -X1*(2.-(X1**2))*AQ/(XS**2)
    XP22=XXW*(2.-(XXW**2))*AR/(XS**2)
     *  -XXD*(2.-(XXD**2))*AE/(XS**2)
          Z=P/XP
     Z1=((XP*P1)-(P*XP1))/(XP**2)
     Z2=((XP*P2)-(P*XP2))/(XP**2)
    Z11=((XP*P11)-(P*XP11))/(XP**2)
     *    -(2*XP1*((XP*P1)-(P*XP1)))/(XP**3)
    Z12=((XP2*P1)+(XP*P12)-(P2*XP1)-(P*XP12))/(XP**2)
     *    -(2*XP2*((XP*P1)-(P*XP1)))/(XP**3)
    Z22=((XP*P22)-(P*XP22))/(XP**2)
     *    -(2*XP2*((XP*P2)-(P*XP2)))/(XP**3)
         ST=FQ(I)-(N*Z)
         RST=ST/Z
      XO1=-RST*Z1*(2*N+RST)
      XO2=-RST*Z2*(2*N+RST)
         FOR1=-((2*N/Z)*(Z11*ST-Z1*N*Z1)-2*Z1*Z1*RST/Z)
         FOR2=(RST/Z)*(Z11*ST-2*N*Z1*Z1)-2*Z1*Z1*RST/(Z**2)
```

```
        XG11=FOR1-FOR2
          FOR3=-((2*N/Z)*(Z22*ST-Z2*N*Z2)-2*Z2*Z2*RST/Z)
          FOR4=(RST/Z)*(Z22*ST-2*N*Z2*Z2)-2*Z2*Z2*RST/(Z**2)
        XG22=FOR3-FOR4
          FOR5=-((2*N/Z)*(Z12*ST-Z1*N*Z2)-2*Z1*Z2*RST/Z)
          FOR6=(RST/Z)*(Z12*ST-2*N*Z1*Z2)-2*Z1*Z2*RST/(Z**2)
        XG12=FOR5-FOR6
22      RETURN
      END
C****************************************************C
C********* GEN DATA OF WEIBULL DIST. *************C
C****************************************************C
      SUBROUTINE WEI
      COMMON/SEED/IU,N
     *      /PAR/T,CT,PT,PCT
     *      /SON/X(1000),XBAR(200)
     *      /PAR1/XD,XW,K
     *      /SS/IR,CC
        I2=0
        I3=0
      DO 10 I=1,N
5       W1=RAND(IU)
      IF((W1.LE.0.).OR.(W1.GE.1.)) GOTO 5
       XX=(-ALOG(1.-W1)/CT)**(1./T)
       IF (XX.LE.XD) GOTO 5
          G=N-CC
       IF((XX.GT.XD).AND.(XX.LE.XW)) THEN
            I2=I2+1
          IF (I2.LE.G)  THEN
             X(I)=XX
           ELSE
              GOTO 5
          END IF
        END IF
        IF((XX.GT.XW).AND.(I3.LE.CC)) THEN
           I3=I3+1
         IF (I3.LE.CC)  THEN
```

```
          X(I)=XW
        ELSE
          GOTO 5
        END IF
      END IF
10    CONTINUE
        CALL  SSORT(N)
        RETURN
      END
C**************************************************C
C************ PERCENTINE MATCHING ************C
C**************************************************C
      SUBROUTINE PER
      COMMON/SEED/IU,N
     *      /PAR/T,CT,PT,PCT
     *      /SON/X(1000),XBAR(200)
     *      /SAM/C(100),FQ(100),F(100)
     *      /SET/N1,N2
     *      /PAR1/XD,XW,K
      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
        PT=T4/T3
        C1=C(N1)**PT
       PCT=-1*T1/C1
      IF((PT.LT.0).OR.(PCT.LT.0)) GOTO 55
C       WRITE(6,304)PT,PCT
C304    FORMAT(2X,' INITIAL VALUE     PT=',E12.7,2X,'CPT=',E12.7/60('-'))
305     RETURN
      END
C****************************************************C
C*********** CUMMULATIVE DIST. OF WEIBULL  ********C
C****************************************************C
      FUNCTION  CDF(XD,XW,XM,XS,CCI)
       A= EXP(-XS*(XD**XM))-EXP(-XS*(CCI**XM))
       B= EXP(-XS*(XD**XM))-EXP(-XS*(XW**XM))
       CDF=A/B
       RETURN
      END
C*****************************************************C
C********** MAXIMUM LIKELIHOOD OF WEIBULL********C
C*****************************************************C
      SUBROUTINE  MLI(I,XM,XS,XO1,XO11,XO12,XO2,XO22)
C       DOUBLE PRECISION DF,CF,BF,AF,OK1,OK2,OK3,OK4
      COMMON/PAR/T,CT,PT,PCT
     *      /SAM/C(100),FQ(100),F(100)
     *      /SET/N1,N2
     *      /PAR1/XD,XW,K
     *      /SIT/DIVG
      CHARACTER DIVG*1
          J=I-1
          CCI=C(I)
        IF(J.EQ.0) THEN
          CCJ=XD
        ELSE
          CCJ=C(J)
        END IF
      IF((XM.GE.20.).OR.(XM.LE.0)) THEN
```

```
      DIVG='Y'
      GOTO 67
    ELSE
      GOTO 76
    END IF
76    DF=(XD**XM)
      CF=(XW**XM)
      BF=(CCJ**XM)
      AF=(CCI**XM)
      IF((DF.GE.174.673).OR.(DF.LE.-174.673)) THEN
        DIVG='Y'
        GOTO 67
      END IF
      IF((CF.GE.174.673).OR.(CF.LE.-174.673)) THEN
        DIVG='Y'
        GOTO 67
      END IF
      IF((BF.GE.174.673).OR.(BF.LE.-174.673)) THEN
        DIVG='Y'
        GOTO 67
      END IF
      IF((AF.GE.174.673).OR.(AF.LE.-174.673)) THEN
        DIVG='Y'
        GOTO 67
      END IF
        OK1=DF*XS
        OK2=CF*XS
        OK3=BF*XS
        OK4=AF*XS
      IF((OK1.GE.174.673).OR.(OK1.LE.-174.673)) THEN
          DIVG='Y'
          GOTO 67
      END IF
      IF((OK2.GE.174.673).OR.(OK2.LE.-174.673)) THEN
          DIVG='Y'
          GOTO 67
      END IF
```

```
      IF((OK3.GE.174.673).OR.(OK3.LE.-174.673)) THEN
         DIVG='Y'
         GOTO 67
      END IF
      IF((OK4.GE.174.673).OR.(OK4.LE.-174.673)) THEN
         DIVG='Y'
         GOTO 67
      END IF
90       P=EXP(-OK3)-EXP(-OK4)
         XP=EXP(-OK1)-EXP(-OK2)
66       P1=XS*(AF*EXP(-OK4)*ALOG(CCI)
     *      - BF*EXP(-OK3)*ALOG(CCJ))
      XP1=XS*(CF*EXP(-OK2)*ALOG(XW)
     *      - DF*EXP(-OK1)*ALOG(XD))
      P11=XS*(AF*(ALOG(CCI)**2)*EXP(-OK4)*(1.-OK4))
     *    - XS*(BF*(ALOG(CCJ)**2)*EXP(-OK3)*(1.-OK3))
      XP11=XS*(CF*(ALOG(XW)**2)*EXP(-OK2)*(1.-OK2))
     *    -XS*(DF*(ALOG(XD)**2)*EXP(-OK1)*(1.-OK1))
      P12=AF*EXP(-OK4)*ALOG(CCI)*(1.-OK4)
     *  -BF*EXP(-OK3)*ALOG(CCJ)*(1.-OK3)
      XP12=CF*EXP(-OK2)*ALOG(XW)*(1.-OK2)
     *  -DF*EXP(-OK1)*ALOG(XD)*(1.-OK1)
      P2=AF*EXP(-OK4)-BF*EXP(-OK3)
      XP2=CF*EXP(-OK2)-DF*EXP(-OK1)
      P22=(BF*BF)*EXP(-OK3)
     *    -(AF*AF)*EXP(-OK4)
      XP22=(DF*DF)*EXP(-OK1)
     *    -(CF*CF)*EXP(-OK2)
      A1=P1/P
      B1=XP1/XP
      C1=A1-B1
        A2=P2/P
        B2=XP2/XP
        C2=A2-B2
      A11=((P*P11)-(P1**2))/(P**2)
      B11=((XP*XP11)-(XP1**2))/(XP**2)
      C11=A11-B11
```

```
        A12=((P*P12)-(P1*P2))/(P**2)
        B12=((XP*XP12)-(XP1*XP2))/(XP**2)
        C12=A12-B12
      A22=((P*P22)-(P2**2))/(P**2)
      B22=((XP*XP22)-(XP2**2))/(XP**2)
      C22=A22-B22
              XG1=C1*FQ(I)
              XG2=C2*FQ(I)
              XG11=C11*FQ(I)
              XG12=C12*FQ(I)
              XG22=C22*FQ(I)
67      RETURN
      END
C*****************************************************************C
C******* LEAST SQUARE METHOD OF WEIBULL *********C
C*****************************************************************C
      SUBROUTINE LSL(I,XM,XS,XG1,XG11,XG12,XG2,XG22)
      DOUBLE PRECISION Z,Z1,Z2,Z11,Z12,Z22
      COMMON/SON/X(1000),XBAR(200)
     *      /SAM/C(100),FQ(100),F(100)
     *      /PAR/T,CT,PT,PCT
     *      /PAR1/XD,XW,K
     *      /SIT/DIVG
     *      /FIN/HOP1,HOP2,SUC1,SUC2
     *      /SEED/IU,N
      CHARACTER DIVG*1
            J=I-1
          CCI=C(I)
        IF(J.EQ.0) THEN
          CCJ=XD
         ELSE
          CCJ=C(J)
         END IF
       IF((XM.GE.20.0).OR.(XM.LE.0)) THEN
          DIVG='Y'
          GOTO 67
        ELSE
```

```
      GOTO 76
    END IF
76    DF=(XD**XM)
     CF=(XW**XM)
     BF=(CCJ**XM)
     AF=(CCI**XM)
    IF((DF.LE.-174.673).OR.(DF.GE.174.673)) THEN
       DIVG='Y'
       GOTO 67
     END IF
    IF((CF.LE.-174.673).OR.(CF.GE.174.673)) THEN
       DIVG='Y'
       GOTO 67
     END IF
    IF((BF.LE.-174.673).OR.(BF.GE.174.673)) THEN
       DIVG='Y'
       GOTO 67
     END IF
    IF((AF.LE.-174.673).OR.(AF.GE.174.673)) THEN
       DIVG='Y'
       GOTO 67
     END IF
      OK1=DF*XS
      OK2=CF*XS
      OK3=BF*XS
      OK4=AF*XS
    IF((OK1.LE.-174.673).OR.(OK1.GE.174.673)) THEN
        DIVG='Y'
        GOTO 67
     END IF
    IF((OK2.LE.-174.673).OR.(OK2.GE.174.673)) THEN
        DIVG='Y'
        GOTO 67
     END IF
    IF((OK3.LE.-174.673).OR.(OK3.GE.174.673)) THEN
        DIVG='Y'
        GOTO 67
```

```
      END IF
      IF((OK4.LE.-174.673).OR.(OK4.GE.174.673)) THEN
        DIVG='Y'
        GOTO 67
      END IF
90      P=EXP(-OK3)-EXP(-OK4)
      XP=EXP(-OK1)-EXP(-OK2)
66      P1=XS*(AF*EXP(-OK4)*ALOG(CCI)
     *   - BF*EXP(-OK3)*ALOG(CCJ))
      XP1=XS*(CF*EXP(-OK2)*ALOG(XW)
     *   - DF*EXP(-OK1)*ALOG(XD))
      P11=XS*(AF*(ALOG(CCI)**2)*EXP(-OK4)*(1.-OK4))
     *   - XS*(BF*(ALOG(CCJ)**2)*EXP(-OK3)*(1.-OK3))
      XP11=XS*(CF*(ALOG(XW)**2)*EXP(-OK2)*(1.-OK2))
     *   -XS*(DF*(ALOG(XD)**2)*EXP(-OK1)*(1.-OK1))
      P12=AF*EXP(-OK4)*ALOG(CCI)*(1.-OK4)
     *   -BF*EXP(-OK3)*ALOG(CCJ)*(1.-OK3)
      XP12=CF*EXP(-OK2)*ALOG(XW)*(1.-OK2)
     *   -DF*EXP(-OK1)*ALOG(XD)*(1.-OK1)
      P2=AF*EXP(-OK4)-BF*EXP(-OK3)
      XP2=CF*EXP(-OK2)-DF*EXP(-OK1)
      P22=(BF*BF)*EXP(-OK3)
     *   -(AF*AF)*EXP(-OK4)
      XP22=(DF*DF)*EXP(-OK1)
     *   -(CF*CF)*EXP(-OK2)
      Z=(P/XP)
      Z1=((XP*P1)-(P*XP1))/(XP**2)
      Z2=((XP*P2)-(P*XP2))/(XP**2)
      Z11=((XP*P11)-(P*XP11))/(XP**2)
     *   -(2*XP1*((XP*P1)-(P*XP1)))/(XP**3)
      Z12=((XP2*P1)+(XP*P12)-(P2*XP1)-(P*XP12))/(XP**2)
     *   -(2*XP2*((XP*P1)-(P*XP1)))/(XP**3)
      Z22=((XP*P22)-(P*XP22))/(XP**2)
     *   -(2*XP2*((XP*P2)-(P*XP2)))/(XP**3)
        AB=(N*Z)-FQ(I)
      XG1=2*N*AB*Z1
      XG11=2*N*((AB*Z11)+(N*(Z1**2)))
```

```
      XG12=2*N*((AB*Z12)+(N*Z1*Z2))
      XG22=2*N*((AB*Z22)+(N*(Z2**2)))
       XG2=2*N*AB*Z2
67    RETURN
      END
C**********************************************************C
C***** MINIMUM CHI-SQUARE METHOD OF WEIBULL ******C
C**********************************************************C
      SUBROUTINE CS(I,XM,XS,XG1,XG11,XG12,XG2,XG22)
      COMMON/SON/X(1000),XBAR(200)
     *      /SAM/C(100),FQ(100),F(100)
     *      /PAR/T,CT,PT,PCT
     *      /PAR1/XD,XW,K
     *      /SIT/DIVG
     *      /FIN/HOP1,HOP2,SUC1,SUC2
     *      /SEED/IU,N
      CHARACTER DIVG*1
           J=I-1
         CCI=C(I)
        IF(J.EQ.0) THEN
          CCJ=XD
         ELSE
          CCJ=C(J)
         END IF
       IF((XM.GE.20.0).OR.(XM.LE.0)) THEN
         DIVG='Y'
         GOTO 67
       END IF
        DF=(XD**XM)
        CF=(XW**XM)
        BF=(CCJ**XM)
        AF=(CCI**XM)
       IF((DF.LE.-174.673).OR.(DF.GE.174.673)) THEN
          DIVG='Y'
          GOTO 67
        END IF
       IF((CF.LE.-174.673).OR.(CF.GE.174.673)) THEN
```

```
        DIVG='Y'
        GOTO 67
      END IF
      IF((BF.LE.-174.673).OR.(BF.GE.174.673)) THEN
        DIVG='Y'
        GOTO 67
      END IF
      IF((AF.LE.-174.673).OR.(AF.GE.174.673)) THEN
        DIVG='Y'
        GOTO 67
      END IF
       OK1=DF*XS
       OK2=CF*XS
       OK3=BF*XS
       OK4=AF*XS
      IF((OK1.LE.-174.673).OR.(OK1.GE.174.673)) THEN
         DIVG='Y'
          GOTO 67
      END IF
      IF((OK2.LE.-174.673).OR.(OK2.GE.174.673)) THEN
          DIVG='Y'
          GOTO 67
      END IF
      IF((OK3.LE.-174.673).OR.(OK3.GE.174.673)) THEN
          DIVG='Y'
          GOTO 67
      END IF
      IF((OK4.LE.-174.673).OR.(OK4.GE.174.673)) THEN
        DIVG='Y'
        GOTO 67
      END IF
90          P=EXP(-OK3)-EXP(-OK4)
        XP=EXP(-OK1)-EXP(-OK2)
66      P1=XS*(AF*EXP(-OK4)*ALOG(CCI)
     *    - BF*EXP(-OK3)*ALOG(CCJ))
      XP1=XS*(CF*EXP(-OK2)*ALOG(XW)
     *    - DF*EXP(-OK1)*ALOG(XD))
```

```
   P11=XS*(AF*(ALOG(CCI)**2)*EXP(-OK4)*(1.-OK4))
*     - XS*(BF*(ALOG(CCJ)**2)*EXP(-OK3)*(1.-OK3))
  XP11=XS*(CF*(ALOG(XW)**2)*EXP(-OK2)*(1.-OK2))
*     -XS*(DF*(ALOG(XD)**2)*EXP(-OK1)*(1.-OK1))
 P12=AF*EXP(-OK4)*ALOG(CCI)*(1.-OK4)
*   -BF*EXP(-OK3)*ALOG(CCJ)*(1.-OK3)
 XP12=CF*EXP(-OK2)*ALOG(XW)*(1.-OK2)
*   -DF*EXP(-OK1)*ALOG(XD)*(1.-OK1)
  P2=AF*EXP(-OK4)-BF*EXP(-OK3)
 XP2=CF*EXP(-OK2)-DF*EXP(-OK1)
  P22=(BF*BF)*EXP(-OK3)
*     -(AF*AF)*EXP(-OK4)
  XP22=(DF*DF)*EXP(-OK1)
*     -(CF*CF)*EXP(-OK2)
            Z=P/XP
 IF(Z.LE.0.000001) THEN
      DIVG='Y'
     GOTO 67
 END IF
   Z1=((XP*P1)-(P*XP1))/(XP**2)
   Z2=((XP*P2)-(P*XP2))/(XP**2)
  Z11=((XP*P11)-(P*XP11))/(XP**2)
*      -(2*XP1*((XP*P1)-(P*XP1)))/(XP**3)
  Z12=((XP2*P1)+(XP*P12)-(P2*XP1)-(P*XP12))/(XP**2)
*      -(2*XP2*((XP*P1)-(P*XP1)))/(XP**3)
  Z22=((XP*P22)-(P*XP22))/(XP**2)
*      -(2*XP2*((XP*P2)-(P*XP2)))/(XP**3)
          ST=FQ(I)-(N*Z)
         RST=ST/Z
     XG1=-RST*Z1*(2*N+RST)
     XG2=-RST*Z2*(2*N+RST)
      FOR1=-((2*N/Z)*(Z11*ST-Z1*N*Z1)-2*Z1*Z1*RST/Z)
      FOR2=(RST/Z)*(Z11*ST-2*N*Z1*Z1)-2*Z1*Z1*RST/(Z**2)
   XG11=FOR1-FOR2
      FOR3=-((2*N/Z)*(Z22*ST-Z2*N*Z2)-2*Z2*Z2*RST/Z)
      FOR4=(RST/Z)*(Z22*ST-2*N*Z2*Z2)-2*Z2*Z2*RST/(Z**2)
   XG22=FOR3-FOR4
```

```
          FOR5=-((2*N/Z)*(Z12*ST-Z1*N*Z2)-2*Z1*Z2*RST/Z)
          FOR6=(RST/Z)*(Z12*ST-2*N*Z1*Z2)-2*Z1*Z2*RST/(Z**2)
        XG12=FOR5-FOR6
67      RETURN
      END
C*****************************************************C
C* NEWTON RAPHSON'RULE FOR PARAMETER SEARCH   *C
C*****************************************************C
      SUBROUTINE CRAM(IZ,MAX)
      COMMON/PAR/RMEW,SIGMA,XMUE,XVAR
     *     /SSD/XEW(1000),XIG(1000),X1C(1000),X2C(1000)
     *     /PAR1/XD,XW,K
     *     /SIT/DIVG
     *     /FIN/HOP1,HOP2,SUC1,SUC2
      CHARACTER DIVG*1
          XSD=SQRT(XVAR)
          PAR1=XMUE
          PAR2=XSD
         RPAR1=RMEW
         RPAR2=SIGMA
          HOP1=0.
          HOP2=0.
           IC=0
      DO 65 M=1,MAX
            G1=0.
           G11=0.
           G12=0.
           G21=0.
            G2=0.
           G22=0.
       DO 70 I=1,K
        CALL MLI(I,PAR1,PAR2,XG1,XG11,XG12,XG2,XG22)
          IF(DIVG.EQ.'Y') THEN
             GOTO 12
           ELSE
              GOTO 69
          END IF
```

```
69          G1=G1+XG1
          G11=G11+XG11
          G12=G12+XG12
          G21=G12
           G2=G2+XG2
          G22=G22+XG22
70      CONTINUE
          DET=(G11*G22)-(G12*G21)
        IF(DET.EQ.0) THEN
           DIVG='Y'
           GOTO 12
        END IF
       HOP1=PAR1+((G12*G2)-(G22*G1))/DET
       HOP2=PAR2+((G1*G21)-(G2*G11))/DET
           IC=IC+1
     IF((ABS(HOP1-PAR1).LE.0.0001).AND.(ABS(HOP2-PAR2).LE.0.0001))THEN
           GOTO 600
       ELSE
          PAR1=HOP1
          PAR2=HOP2
         IF(IC.GE.MAX) THEN
             DIVG='Y'
              GOTO 12
          END IF
     END IF
65      CONTINUE
600             SUC1=(HOP1-RPAR1)**2
              SUC2=(HOP2-RPAR2)**2
            XEW(IZ)=HOP1
            XIG(IZ)=HOP2
            X1C(IZ)=SUC1
            X2C(IZ)=SUC2
C       WRITE(6,5)
C5      FORMAT(2X,'*** FINAL PARAMETER AND SQUAR ERROR (PAR-FPAR)1,2 **')
C      WRITE(6,10) HOP1,HOP2,SUC1,SUC2
C10     FORMAT(4(2X,E12.5))
12        RETURN
```

```
      END
C*****************************************************C
C************  EVERAGE MUE AND SIGMA   **************C
C*****************************************************C
      SUBROUTINE VALUE (IZ)
      COMMON/SSD/XEW(1000),XIG(1000),X1C(1000),X2C(1000)
     *     /PAR/RMEW,SIGMA,XMUE,XVAR
     *     /SSU/SUMM,SUMS,SUMD1,SUMD2
     *     /SS/IR,CC
     *     /FIN/HOP1,HOP2,SUC1,SUC2
           SUM=0.
           SUS=0.
      IF(IZ.EQ.IR) THEN
          DO 27 I=1,IZ
             SUMM=SUMM+XEW(I)
             SUMS=SUMS+XIG(I)
             SUMD1=SUMD1+X1C(I)
             SUMD2=SUMD2+X2C(I)
27         CONTINUE
C            WRITE(6,18)SUMM,SUMS
C18           FORMAT(2X,' SUMM=',E12.7,2X,' SUMS=',E12.7)
C            WRITE(6,19)SUMD1,SUMD2
C19           FORMAT(2X,'SUMD1=',E12.7,2X,'SUMD2=',E12.7)
             YMW=SUMM/IZ
             YSG=SUMS/IZ
            RMUE=SUMD1/IZ
            RMSG=SUMD2/IZ
           DO 20 I=1,IZ
            SUM=SUM+(XEW(I)**2)
            SUS=SUS+(XIG(I)**2)
20           CONTINUE
              A=(SUMM**2)/IZ
              B=(SUMS**2)/IZ
           BIAS1=(YMW-RMEW)**2
           BIAS2=(YSG-SIGMA)**2
            VAR1=(SUM-A)/IZ
            VAR2=(SUS-B)/IZ
```

```
      WRITE(6,51)
51     FORMAT(2X,'****** BIAS1, BIAS2, VAR1, VAR2 ******')
      WRITE(6,50)BIAS1,BIAS2,VAR1,VAR2
50     FORMAT(4(2X,E12.7))
       RMSEME=(RMUE+RMSO)/2
     WRITE(6,55)YMW,YSO
55    FORMAT(2X,'PAR1=',E12.5,2X,'PAR2=',E12.5/60('-'))
     WRITE(6,35)RMUE,RMSO,RMSEME
35   FORMAT(2X,'MSE1=',E12.7,2X,'MSE2=',E12.7,2X,'MSE=',E12.7/60('-'))
     ELSE
       GOTO 30
    END IF
30     RETURN
    END
```

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้วิจัย

นางสาวทศพร แถลงธรรม เกิดเมื่อวันที่ 25 มิถุนายน พ.ศ. 2516 อำเภอเมือง จังหวัดร้อยเอ็ด สำเร็จการศึกษา ปริญญาวิทยาศาสตรบัณฑิต สาขาคณิตศาสตร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ในปีการศึกษา 2536 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาการประกันภัย ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2537