บรรณานุกรม

ภาษาไทย

วิทยานิพนธ์

ปราณี รัตนัง "การประมาณสัมประสิทธิ์การถดถอยพหุ เมื่อความผิดพลาดมีการแจกแจงแบบเบ้และมีการแจกแจงแบบหางยาวกว่าการแจกแจงปกติ" วิทยานิพนธ์ ปริญญามหาบัณฑิต ภาควิชาสถิติบัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2530

ภาษาต่างประเทศ

หนังสือ

Andrews, D.F., Bickel, P.J., Hampel, F.R., Rogers, W.H. and Tukey, J.W. Robust Estimates of Location : Survey and Advance. Princeton University Press, 1972.

Frank, R. Hampel, Perter, J. Rousseeuw, Elvezio, M. Ronchette and Werner, A. Stahel. Robust Statistics The Approch Based on influence Function. New York : John Wiley, 1986.

Hoagtin, D.C., Marteller, F. and Tukey, J.W. Understanding Robust and Exploratory Data Analysis. New York : John Wiley, 1981.

Huber, P.J. Robust Statistics. New york : John Wiley, 1981.

Jude, G.G., Griffiths, W.E., Hill, R.C. and Lee, T.C. The Theory and Practice of Econometric. New York : John Wiley, 1980.

Robinstein, R.Y. Simulation and Monte Carlo method. New York : John Wiley, 1981.

วารสาร

Alkinson, A.C. "Diaganostic Regression Analysis and shilfed Power Transformation." Technometrics 25 (1983) : 23-33.

Andrews, D.F. "A Robust Method for Multiple Linear Regression." Technometrics 16 (November 1974) : 523-531.

Carroll, R.J. and David, R. "Transformation in Regression : A Robust Analysis." Technometrics 27 No.1 (February 1985) : 1-12.

David, A. Lax "Robust Estimators of Scale : Finite-Sample Performance in Long-tailed Symmetric Distributions." Journal of the American Statistic Association 80 (1985) : 736-741.

Gilstein, C.A. and Leamer, E.E. "Robust sets of Regression Estimates." Econometrica 51 No. 2 (1983) : 321-333.

Godambc, V.P. and Thanpson, M.E. "Robust Estimation through estimating equation." Biometrika 71 (1984) : 115-125.

Peter, J. Huber "Robust Estimation of Location Parameter." Annual Math. Stat (1964) : 73-101.

Ranchetti, E. "Robust Model Selection in Regression." Statistic & Probability Letter 3 (1985) : 21-23.

Shier, D.C. and Lawrence, K.D. "A Compairison of Robust Regression Technique for the Estimation of Weibull Parameter." Communication in Statistic Part B 13 (June 1984) : 743-750.

Silvapulle, M.J. "Asymtotic behavior of Robust estimator of Regression Scale parameter with fixed carriers." The Annauls of Statistics 13 No. 4 (1985) : 1490-1497.

Wolfgang, H. "Robust Regression function estimation." Journals of multivariate Analysis 14 (1984) : 169-180.

ภาคผนวก ก.

ภาคผนวก ก.

ในการสร้างตัวแปรให้มีคุณสมบัติตามต้องการ วิธีการหนึ่งที่สามารถทำได้คืออาศัยเทคนิคของการผลิตเลขสุ่มโดยการใช้โปรแกรม

1. <u>การผลิตเลขสุ่มโดยการใช้โปรแกรม</u>

เป็นการผลิตเลขสุ่มจากความสัมพันธ์ที่ซ้ำๆ (recurrence relation) กล่าวคือเลขถัดไปเกิดจากการดำเนินการทางคณิตศาสตร์และตรรกศาสตร์ของตัวเลขปัจจุบันหรือกลุ่มตัวเลขในอดีต อนุบรรพ (sequence) ของตัวเลขซึ่งผลิตได้จึงเป็นอนุบรรพของเลขสุ่มในความหมายที่แท้จริงไม่ได้และย่อมมีคาบ (period) แต่อย่างไรก็ตามเลขที่ผลิตในอนุบรรพเหล่านี้อาจจะผ่านการทดสอบความเป็นสุ่มทางสถิติได้หลายอย่างและเรียกว่าเลขคล้ายสุ่ม (pseudo-random number)

การผลิตเลขสุ่มโดยการใช้โปรแกรมมีข้อดีหลายประการที่สำคัญคือ วิธีนี้สามารถผลิตอนุบรรพของเลขชุดเดิมออกมาได้ซึ่งเป็นสิ่งจำเป็นอย่างยิ่งในกรณีที่ใช้การทดสอบแบบจำลอง และมีความประสงค์จะทบทวนการคำนวณโปรแกรม

ข้อบกพร่องของวิธีการผลิตนี้ก็คือ อนุบรรพของเลขสุ่มที่ผลิตออกมาเป็นอนุบรรพที่มีคาบทั้งสิ้นและการอนุมานคุณสมบัติเชิงสถิติของเลขสุ่มเหล่านี้ทางทฤษฎีกระทำได้ยากพอสมควร

เทคนิคการผลิตเลขสุ่มโดยการใช้โปรแกรมได้รับการพัฒนาอย่างรวดเร็ว Von Neuman และ Metropolis ได้เสนอวิธีตัวกลางกำลังสอง (mid-square method) ในปี ค.ศ. 1946 Poraythe ได้ปรับปรุงวิธีตัวกลางกำลังสอง และต่อมา Lehmer ได้เสนอวิธีผลิตเลขสุ่มด้วยการใช้เศษจากการหารผลคูณ (multiplicative congruential method) ซึ่งเป็นวิธีที่ใช้กันอย่างแพร่หลายในคอมพิวเตอร์ยุคต่อมา Rotenleerg ได้ปรับปรุงวิธีการผลิตของ Lehmer เป็นการใช้เศษของผลบวกของผลคูณกับค่าคงที่จากการหาร (mixed congruential method)

วิธีการแบบ multiplicative congruentail method จะหาเลขสุ่มโดยทำการคำนวณจากสมการ

(1) $$X_{i+1} = X_i \cdot a \ (\text{Modulo } m)$$

เมื่อ X_i เป็นเลขคล้ายสุ่มตัวที่ i

	X_{i+1}	เป็นเลขคล้ายสุ่มตัวที่ i+1
และ	a	เป็นตัวคูณคงที่ (Constant Multiplier)

Modulo m หมายความว่า ค่า (X_i,a) ถูกหารด้วย m จนกระทั่งเหลือเศษน้อยกว่าค่า m เลขที่เหลือเศษจึงเป็นเลขคล้ายสุ่มตัวต่อไปคือ X_{i+1}

วิธีการเริ่มต้นโดยค่าเริ่มต้น X เรียกว่า initial value หรือ seed จากการใช้สมการ (1) จะได้เลขคล้ายสุ่มที่เป็นเลขจำนวนเต็มค่าใดค่าหนึ่งในช่วง 0,1,...,m-1 หลังจากนั้นแล้วจะได้เลขคล้ายสุ่มชุดเดิมอีก ฉะนั้นคาบของเลขคล้ายสุ่มที่ได้จึงมีค่าไม่เกิน m (คาบของเลขคล้ายสุ่มมีค่าน้อยกว่า m เมื่อเลือกค่า a และ X_o ไม่ดีนัก) การเลือกค่า m,a และ X_o จึงมีความสำคัญในการผลิตเลขคล้ายสุ่มที่มีคาบใกล้เคียงกับ m มากที่สุด

Lehmer ได้ทดลองเลือกใช้ค่า m,a และ X_o ที่จับคู่ต่างๆ กันเพื่อใช้ผลิตเลขคล้ายสุ่มตามสมการที่ (1) พบว่าถ้าเลือก X_o เป็นเลขคี่ และ $m = 2^r$ (เมื่อ r>2) และ a = 8k 3 (เมื่อ k เป็นจำนวนเต็มบวกใดๆ) จะได้คาบของเลขคล้ายสุ่มมากที่สุด และเท่ากับ 2^{r-2} วิธีการดังกล่าวเป็นวิธีการเลือกค่า parameter ทั้ง 3 ตัว เพื่อจะได้กลุ่มของเลขคล้ายสุ่มที่ดีและมีการแจกแจงสม่ำเสมอในช่วง (0,1)

2) <u>การผลิตเลขสุ่มที่มีการแจกแจงสม่ำเสมอ</u>

จากสมการการผลิตเลขสุ่มสามารถผลิตเลขสุ่มที่มีการแจกแจงสม่ำเสมอได้โดยตรง คือ

1) ค่า m เป็นค่าของจำนวนเต็มที่ใหญ่ที่สุด (largest integer) และเป็นเลขคี่ที่สามารถคำนวณได้จากเครื่องคอมพิวเตอร์ จาก $m = 2^b$ เมื่อ b เป็นค่าความยาว 1 word หรือ จำนวน bit ใน 1 word ในการศึกษาครั้งนี้ใช้เครื่อง IBM 3031 system 32 bit binary และเป็นเลขคี่ที่คอมพิวเตอร์ได้รับคือ 2^{b-1} เท่ากับ $2^{31} - 1 = 2147483647$ นั่นคือค่า m ควรมีค่า 2147483647

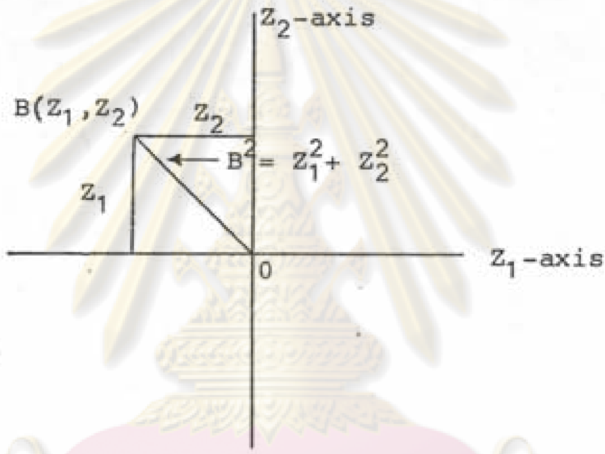
2) ค่า seed (X_o) ควรมีค่าที่เป็น prime กับค่า m (relatively prime to m) เมื่อ m เป็นค่ากำลังของ 2 (จาก $m = 2^b$) ดังนั้น X จึงควรมีค่าเป็นเลขจำนวนเต็มบวกที่เป็นเลขคี่ (ในกรณีที่ใช้ X_o เป็นเลขคู่จะพบว่าทุกๆ ค่า X_i ต่อไปจะเป็นเลขคู่เสมอ จึง

3. <u>การผลิตเลขสุ่มที่มีการแจกแจงแบบปกติ</u>

การแจกแจงแบบปกติโดยใช้เทคนิคแบบการแปลงโดยตรงจากสมการในรูปของ

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

Box และ Muller (ค.ศ. 1958) สร้างเลขสุ่มที่มีการแจกแจงแบบปกติมาตรฐานที่มีค่าเฉลี่ยเป็น 0 และค่าความแปรปรวนเป็น 1 พร้อมๆ กัน 2 ค่า ดังนี้



$$Z_1 = B \cos \theta$$

$$Z_2 = B \sin \theta$$

B^2 มีการแจกแจงไคสแควร์ (chi-square distribution) ด้วยระดับความเป็นอิสระ 2 ซึ่งเทียบเท่า (equivalent) กับการแจกแจงแบบเอ็กซ์โปเนนเชียล (exponential distribution) ด้วยค่าเฉลี่ย 2 ดังนั้นรัศมี B มีค่าดังนี้

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

โดยการสมมาตร (symmetry) ของการแจกแจงแบบปกติ (normal distribution) เมื่อ θ มีการแจกแจงแบบสม่ำเสมอ (uniform distribution) ระหว่าง 0 กับ 2π เรเดียน

ไม่มีคุณสมบัติเป็นเลขสุ่ม)

3) ค่าคงที่ที่ใช้เป็นตัวคูณ a (constant multiplier) ควรมีค่าเป็น prime กับ m ด้วยนั่นคือ a ต้องเป็นเลขคี่ พบว่าวิธีเลือกที่ดีที่สุดสำหรับค่า a เมื่อใช้ความสัมพันธ์ 3 (mod m) หรือ a 8t + 3 เมื่อ t เป็นค่าบวกใดๆ a จะมีค่าใกล้ $2^{b/2}$ ซึ่ง a จะเป็นเลขอันดับแรกของความสัมพันธ์ระหว่างเลขคล้ายสุ่ม ดังนั้นเครื่อง IBM 3031 system 32 bit binary machine เราเลือกใช้ค่า $a = 2^{16} + 3 = 65539$

ในกรณีของเครื่องคอมพิวเตอร์จะใช้หลักการ multiplicative congruential method และ Shift register ดังนั้นการคำนวณ aX_o มีผลคูณที่เกิน fixed length หรือ 1 word โดยผลคูณของเลขจำนวนเต็มจะประกอบด้วย 2b bits จากผลลัพธ์นี้ตัวเลขหลักค่าสูงใน b bit แรกจะถูกตัดทิ้งไปและตัวเลขหลักค่าต่ำ b bit หลังจะแทนด้วย X_1 ค่าตัวเลขซึ่งมีการแจกแจงแบบสม่ำเสมอคือเศษจากการหาร $r1 = X_1/(2^b)$ ซึ่งจะมีค่าอยู่ระหว่าง 0 กับ 1 และค่า $0 < X_i < m$

โปรแกรมย่อยที่ใช้คือ RAND(IX, IY, YFL) ซึ่ง IX คือเลขสุ่มที่เป็นค่าเริ่มต้นที่เข้าไปในโปรแกรมย่อย IY คือเลขสุ่มตัวถัดไปที่คำนวณได้จากเลขสุ่มตัวเริ่มต้น YFL คือเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอในช่วง (0,1) สำหรับโปรแกรมย่อย RAND เขียนได้ดังนี้

```
      SUBROUTINE RAND (IX, IY, YFL)
      IY  = IX x 65539
      IF (IY) 5,6,6
5     IY  = IY + 2147483647 + 1
6     YFL = IY
      YFL = YFL / 2147483647
      IX  = IY
      RETURN
      END
```

ซึ่งค่า B และ θ เป็น mutually independent

$$Z_1 = (-2 \ln R_1)^{1/2} \cos(2\pi R_2)$$
$$Z_2 = (-2 \ln R_1)^{1/2} \text{sine}(2\pi R_2)$$

ฟังก์ชันสำหรับจำลองแบบประชากรที่มีการแจกแจงเป็นแบบปกติ มีค่าเฉลี่ย μ และ ค่าเบี่ยงเบนมาตรฐาน σ จะเรียกใช้ FUNCTION NORMAL(DMEAN, SIGMA) ซึ่งจะได้จากค่า NORMAL = Z_1x SIGMA + DMEAN หรือ NORMAL = Z_2x SIGMA + DMEAN ในแต่ละครั้ง ดังนั้น คำสั่งในการสร้างตัวแปรให้มีการแจกแจงแบบปกติ คือ

```
      FUNCTION NORMAL(DMEAN,SIGMA)
      REAL NORMAL
      COMMON /SEED/IX/SELECT/KK
      PI    = 3.1415926
      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
```

4. <u>การสร้างการแจกแจงแบบปกติปลอมปน</u>

การสร้างตัวแปรสุ่มที่มีการแจกแจงแบบปกติปลอมปนที่มีค่าเฉลี่ย μ และส่วนเบี่ยงเบนมาตรฐาน σ^2 ตามที่กำหนดจะใช้วิธีที่ Ramsay (ค.ศ. 1977) เสนอไว้โดยพิจารณาการแจกแจงซึ่งแปลงมาจากการแจกแจงแบบปกติที่มีฟังช์การแจกแจงอยู่ในรูปของ

$$F(x) = (1-p)N(\mu,\sigma^2) + pN(\mu,c^2\sigma^2)$$

หมายความว่าตัวแปรสุ่ม X มาจากการแจกแจง N (μ,σ^2) ด้วยความน่าจะเป็น 1-p และจากการแจกแจง N $(\mu,c^2\sigma^2)$ ด้วยความน่าจะเป็น p โดยที่

และ เป็นค่ากำหนดค่าเฉลี่ย และความแปรปรวน

p และ c เป็นค่ากำหนดเปอร์เซนต์การปลอมปน และสเกลแฟคเตอร์

ดังนั้นคำสั่งในการสร้างตัวแปรให้มีการแจกแจงแบบปกติปลอมปนคือ

```
10    DMEAN = XBAR1(M)
      SIGMA = SG(M)
      SG2   = IC * SIGMA
      DO 15 J = 1,N
      CALL RAND(IX,IY,YFL)
      IF(YFL - P) 11,11,12
11    E(J)  = NORMAL(DMEAN,SG2)
      GOTO 15
12    E(J)  = NORMAL(DMEAN,SIGMA)
```

```
15    CONTINUE
```

5. <u>การสร้างการแจกแจงแบบที</u>

การแจกแจงแบบที เป็นการสร้างจากตัวแปรสุ่ม X ซึ่งมีการแจกแจงแบบปกติมาตรฐาน และ Y ซึ่งเป็นตัวแปรสุ่มที่มีการแจกแจงแบบไคสแควร์ที่มีระดับความเป็นอิสระเป็น k โดยที่ตัวแปร X และ Y เป็นอิสระต่อกัน

ดังนั้น คำสั่งในการสร้างตัวแปรให้มีการแจกแจงแบบทีคือ

```
      FUNCTION TDIST(NDF,DMEAN,SIGMA)
      REAL NORMAL
C     T-DISTRIBUTION : T = X(NORMAL(DMEAN,SIGMA)**2)
      SQNOR = 0.0
      DO 10 I = 1,NDF
10    SQNOR = SQNOR + (NORMAL(DMEAN,SIGMA)**2)
      CHISQ = SQRT(SQNOR/FLOAT(NDF))
      TDIST = NORMAL(DMEAN,SIGMA)/CHISQ
      RETURN
      END
```

6. <u>การสร้างการแจกแจงแบบลอกนอร์มอล</u>

การแจกแจงแบบลอกนอร์มอลมีฟังก์ชันความหนาแน่นอยู่ในรูปของ

$$f(x) = \begin{cases} \dfrac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\ln(x-\mu/\sigma^2))} & ; \quad x>0, \sigma>0, -\infty<\mu<\infty \\ 0 & ; \quad \text{อื่น ๆ} \end{cases}$$

เมื่อ μ และ σ^2 เป็นค่าเฉลี่ยและความแปรปรวนของ Y ซึ่ง Y = ln X และ Y มีการแจกแจงแบบปกติ

โดยมี $\exp(\sigma^2)$ เป็น scale parameter

และ μ เป็น shape parameter

ค่าคาดหวัง ค่าความแปรปรวนและค่าสัมประสิทธิ์ความแปรปรวนของการแจกแจงแบบลอกนอร์มอล คือ

$$E(X) = \exp\{\mu + \sigma^2/2\}$$

$$V(X) = \exp\{2\mu + \sigma^2\} \exp\{\sigma^2\} - 1$$

$$C.V.(X) = \exp\{\sigma^2\} - 1$$

ดังนั้นคำสั่งในการสร้างตัวแปรให้มีการแจกแจงแบบลอกนอร์มอลคือ

```
30    SIGMA = SG(M)
      DO 35 J = 1,N
      DMEAN1 = X(1,J)
      X(M2,J) = EXP(NORMAL(DMEAN1,SIGMA))
35    CONTINUE
```

7. <u>การสร้างการแจกแจงแบบแกมมา</u>

การแจกแจงแบบแกมมา มีฟังก์ชันความหนาแน่นอยู่ในรูปของ

$$f(x) = \begin{cases} \dfrac{x^{\alpha-1}\exp\{-x/\beta\}}{\beta^{\alpha}\,\Gamma(\alpha)} & ; x>0, \alpha>0, \beta>0 \\ 0 & ; \text{อื่น ๆ} \end{cases}$$

เมื่อ β เป็น scale parameter

และ α เป็น shape parameter

การสร้างตัวแปรสุ่มให้มีการแจกแจงแบบแกมมาใช้คุณสมบัติ reproductive property เมื่อ X_i โดยที่ $i = 1, \ldots, n$ เป็นตัวแปรสุ่มจากการแจกแจงแบบ GAMMA(G) แล้ว $X = \sum_i^n X_i$ มีรูปแบบเป็น $G(\alpha, \beta)$ ซึ่ง $\alpha = \sum_i^n \alpha_i$ ดังนั้นเมื่อ α เป็นตัวเลขค่าเต็มหรือ $\alpha = m$ ตัวแปรจากการแจกแจงแบบแกมมา $G(m, \beta)$ สามารถผลิตได้โดยการรวมตัวแปรสุ่มแบบเอ็กซ์โปเนนเชียลที่เป็นอิสระ m ตัวดังนี้

$$X = \beta \sum_i^n (-\ln U_i)$$
$$= -\beta \ln \prod_i^n U_i$$

เมื่อ U_i เป็นตัวแปรสุ่มที่มาจากการแจกแจงแบบสม่ำเสมอมีพิสัยภายในช่วง 0 ถึง 1

ค่าคาดหวัง ค่าความแปรปรวนและค่าสัมประสิทธิ์ความแปรปรวนของการแจกแจงแบบแกมมาคือ

$$E(X) = \beta\alpha$$

$$V(X) = \beta^2\alpha$$

$$C.V.(X) = 1/\sqrt{\alpha}$$

ดังนั้นคำสั่งในการสร้างตัวแปรให้มีการแจกแจงแบบแกรมมาคือ

```
      FUNCTION GAMMA1(ALPHA1,BETA1)
      COMMON /SEED/ IX
C     GAMMA DISTRIBUTION : X = -BETA * SUM(LN(R(I)) ; I = ALPHA,....,1
                          R IS RANDOM VARIABLE FROM U(0,1)
      ALPHA = ALPHA1
      U = 0,0
5     CALL RAND(IX,IY,YFL)
      V = -ALOG(YFL)
      U = U + V
```

```
      IF(ALPHA.EQ.1.0)GOTO 10
      ALPHA = ALPHA - 1.0
      GOTO 5
   10 GAMMA1 = BETA1 * U
      RETURN
      END
```

8. <u>การสร้างการแจกแจงแบบไวบูลล์</u>

การแจกแจงแบบไวบูลล์มีฟังก์ชันความหนาแน่นอยู่ในรูปของ

$$f(x) = \begin{cases} \dfrac{x^{\alpha-1}\ \exp\{-x_{/\beta}\}^2}{\beta^{\alpha}} & ;\quad x>0,\ \alpha>0,\beta>0 \\ 0 & ;\quad \text{อื่น ๆ} \end{cases}$$

โดย β เป็น scale parameter

และ α เป็น shape parameter

ค่าคาดหวัง ค่าความแปรปรวนและค่าสัมประสิทธิ์ความแปรปรวนของการแจกแจงแบบไวบูลล์คือ

$$E(X) = \beta\Gamma(1 + \frac{1}{\alpha})$$

$$V(X) = \beta^2 [\ \Gamma(1 + \frac{2}{\alpha}) - \Gamma^2 (1 + \frac{1}{\alpha})\ ]$$

$$C.V.(X) = \left[\frac{\Gamma(1 + 2/\alpha)}{\Gamma^2 (1 + 1/\alpha)} - 1 \right]^{\frac{1}{2}}$$

การสร้างตัวแปรสุ่มให้มีการแจกแจงแบบไวบูลล์อาศัยเทคนิคการแปลงผกผัน (inverse transformation) ดังนี้

ขั้นที่ 1 cdf. เขียนเป็น $F(X) = 1-e^{-(X/\beta)^{\alpha}}$; $X>0$

ขั้นที่ 2 ให้ $F(X) = 1-e^{-(X/\beta)^{\alpha}}$

$= R$

ขั้นที่ 3 หาค่าของ X ในเทอมของ R ได้เป็น $X = \beta(-\ln(1-R))^{1/\alpha}$

ดังนั้น คำสั่งในการสร้างตัวแปรให้มีการแจกแจงแบบไวบูลล์คือ

```
      FUNCTION WEIBUL(ALPHA1,BETA1)
      COMMON /SEED/ IX
C     WEIBULL DISTRIBUTION : X = BETA(-LN(1-R)**1/ALPHA
C                            R IS RANDOM VARIABLE FROM U(0,1)
      CALL RAND(IX,IY,YFL)
      WEIBUL = BETA1*(-ALOG(1.0-YFL))**(1.0/ALPHA1)
      RETURN
      END
```

ภาคผนวก ข.

ภาคผนวก ข.

1. วิธีการค้นหาค่า $\hat{\lambda}$ ซึ่งทำให้ $\sigma^2_{\varepsilon|\lambda}$ มีค่าน้อยที่สุด สำหรับการแปลงที่อยู่ในรูปยกกำลังของ BOX และ COX

วิธีการค้นหาค่า $\hat{\lambda}$ ที่ทำให้ $\sigma^2_{\varepsilon|\lambda}$ มีค่าน้อยที่สุด สำหรับการแปลงที่อยู่ในรูปยกกำลังของ Box และ Cox นี้ ผู้วิจัยได้เขียนขึ้นเพื่อจะช่วยให้การค้นหา $\hat{\lambda}$ ได้รวดเร็วขึ้น จาก Montgomery (ค.ศ.1982 หน้า 94-96) ให้ข้อสังเกตว่าโดยปกติเราจะหาค่า $\hat{\lambda}$ จำนวน 10 ถึง 20 ค่าให้เพียงพอสำหรับการประมาณค่าที่เหมาะสม เราไม่สามารถเลือก $\hat{\lambda}$ ที่จะเปรียบเทียบโดยตรงกับผลบวกของค่าผิดพลาดกำลังสองจากการถดถอย $Y^{\hat{\lambda}}$ บน X เพราะว่าแต่ละ $\hat{\lambda}$ มีค่าผลบวกของค่าผิดพลาดกำลังสองที่วัดได้ที่สเกลแตกต่างกัน ดังนั้นการหาค่า $\hat{\lambda}$ จึงอ่านจากกราฟและช่วงความเชื่อมั่น 100 (1-α) ของ λ สามารถหาได้จากการคำนวณในรูปสมการดังนี้

ดังนั้นวิธีการค้นหา $\hat{\lambda}$ ซึ่งทำให้ $\sigma^2_{\varepsilon|\lambda}$ มีค่าน้อยที่สุดสำหรับการแปลงที่อยู่ในรูปยกกำลังของ Box และ Cox มีขั้นตอนดังนี้

1) กำหนดช่วงพิสัยของ λ ที่จะใช้ในการหา $\sigma^2_{\varepsilon|\lambda}$ โดยให้ λ_1 เป็นจุดเริ่มต้น λ_2 เป็นจุดกลาง และ λ_3 เป็นจุดสุดท้าย ระหว่างจุด λ_1 กับ λ_2 สามารถแบ่งเป็นจุดเล็กๆ ที่ยอมรับได้ห่างกัน $d = 0.1$ จำนวนจุดที่อยู่ระหว่าง λ_1 กับ $\lambda_2 = MR = (|\lambda_1| + |\lambda_2|)/d$ โดย MR เป็นเลขจำนวนเต็มคู่มีค่าเป็น 2^b

2) หาค่า $\sigma^2_{\varepsilon|\lambda}, \hat{\beta}_{\varepsilon|\lambda}$ ของทั้ง 3 จุด จะได้เป็น

$$\sigma^2_{\varepsilon|\lambda_1}, \quad \hat{\beta}_{\varepsilon|\lambda_1}$$
$$\sigma^2_{\varepsilon|\lambda_2}, \quad \hat{\beta}_{\varepsilon|\lambda_2}$$
$$\sigma^2_{\varepsilon|\lambda_3}, \quad \hat{\beta}_{\varepsilon|\lambda_3}$$

3) เปรียบเทียบ $\sigma^2_{\varepsilon|\lambda_1}$ กับ $\sigma^2_{\varepsilon|\lambda_2}$ และ $\sigma^2_{\varepsilon|\lambda_3}$ โดยหาค่า $\sigma^2_{\varepsilon|\lambda}$ ที่มีค่าน้อยที่สุด เมื่อค่า $\sigma^2_{\varepsilon|\lambda}$ มีค่าน้อยที่สุดจะกำหนดให้เป็น $\hat{\lambda}$ คือ λ_2 เปลี่ยนค่า $\sigma^2_{\varepsilon|\lambda}$ และ $\beta_{\varepsilon|\lambda}$ ใดๆ มาเป็น $\sigma^2_{\varepsilon|\lambda_2}$ และ $\beta_{\varepsilon|\lambda_2}$

4) เปรียบเทียบค่า MR และ 1

   ถ้า MR = 1 จะยอมรับค่า λ_2 เป็นค่าที่ให้ $\sigma^2_{\varepsilon|\lambda}$ น้อยที่สุด และจะได้ค่า $\beta_{\varepsilon|\lambda_2}$ ด้วย ถือว่าจบกระบวนการ

   ถ้า MR $\neq$ 1 ให้ไปทำข้อ 5)

5) คำนวณค่า MR(ใหม่) = MR(เก่า)/2

   คำนวณ $\lambda_1 = \lambda_2 - 0.1 \times MR(\text{ใหม่})$

   คำนวณ $\lambda_3 = \lambda_2 + 0.1 \times MR(\text{ใหม่})$

6) กลับไปทำข้อ 2)

สำหรับโปรแกรมนี้ λ_1 และ λ_2 สามารถเคลื่อนย้ายออกไปจากจุดที่กำหนดตั้งแต่เริ่มต้นได้ถ้าหากว่าค่า $\hat{\lambda}$ ที่ต้องการไม่ได้อยู่ในพิสัยตั้งแต่เริ่มต้น และจำนวนครั้งของการค้นหาจะขึ้น

อยู่กับจำนวนของ MR ครั้งแรก ซึ่งถ้า MR = 2^b จำนวนครั้งของการค้นหาเท่ากับ 2b + 3 ครั้ง

2. การทดสอบว่าข้อมูลมีการแจกแจงแบบปกติของ Shapiro และ Wilk

การทดสอบของ Shapiro และ Wilk (1965 หน้า 591-611) มีค่าสถิติทดสอบ คือ

$$W = \frac{\left[\sum_{i}^{n} a_{n-i+1}(Y_{n-i+1} - Y_i)\right]^2}{\sum_{i}^{n}(Y_i - \bar{Y})^2}$$

โดย k เป็นค่าของเลขนับที่ใหญ่ที่สุดของ n/2

และ a_{n-i+1} เป็นสัมประสิทธิ์จากตารางของ Shapiro และ Wilk สำหรับขนาดตัวอย่าง n < 50 และมีขั้นตอนการทำงานดังนี้

ก. ถ้า n เป็นเลขคู่ โดยที่ n = 2k คำนวณค่า b ดังนี้

$$b = \sum_{i}^{k} a_{n-i+1}(Y_{n-i+1} - Y_i)$$

ถ้า n เป็นเลขคี่ โดยที่ n = 2k+1 คำนวณค่า b ดังนี้

$$b = a_n(Y_n - Y_1) + \ldots + a_{k+2}(Y_{k+2} - Y_k)$$

เมื่อ $a_{k+1} = 0$

และค่า Y_{k+1} เป็นค่ามัธยฐาน (median) จะไม่ถูกนำไปคำนวณค่า b ด้วย

ข. คำนวณค่า $S^2 = \sum_{i}^{n}(Y_i - \bar{Y})^2$

ค. คำนวณ W ในรูปของ $W = b^2/S^2$

ง. เปรียบเทียบค่า W(คำนวณ) กับ W(ตาราง) ถ้า W(คำนวณ) มีค่าน้อยอย่างมีนัยสำคัญแสดงว่าข้อมูลไม่ได้มาจากการแจกแจงแบบปกติ

Shapiro Wilk และ Chen (1964 หน้า 1343-1372) ได้ศึกษาเปรียบเทียบตัวสถิติ

ทดสอบ W กับ$\sqrt{b_1}$(โมเมนต์ที่ 3) b2 KS(Kolmogorov-Smirnov) CM(Cramer Von Mises) WCM(Weighted CM) D(modified KS) CS(Chi-squared) และ U (Studentized range) ในการแจกแจง 45 แบบต่างๆ กันและขนาดตัวอย่างต่างๆ กับ วิธีการทดสอบของ Shipiro และ Wilk เป็นตัววัดความเป็นปกติที่เหนือกว่าวิธีการอื่นๆ

สำหรับขนาดตัวอย่าง n > 50, Shapiro และ Francia (1972) ได้กล่าวถึงค่าสถิติ W ที่ปรับปรุงขึ้นในรูปประมาณเป็นการทดสอบ W' โดยที่

$$W' = \frac{[\sum_{i}^{n} b_i Y_i]^2}{\sum_{i}^{n} (Y_i - \bar{Y})^2}$$

เมื่อ $b' = (b_1, b_2, \ldots, b_n)$

$= m'/[(m'n)^{1/2}]$

และ $m' = (m_1, \ldots, m_n)$ แทนเวกเตอร์ของค่าปกติมาตรฐานของค่าสถิติอันดับ (order statistics)

ตารางที่ 1 แสดงค่าสัมประสิทธิ์ (a_{n-i+1}) สำหรับการทดสอบ W ของการแจกแจงปกติ สำหรับ n = 20

i	a_{n-i+1}	i	a_{n-i+1}
1	0.4734	6	0.1334
2	0.3211	7	0.1013
3	0.2565	8	0.0711
4	0.2085	9	0.0422
5	0.1686	10	0.0140

ตารางที่ 2 แสดงสัมประสิทธิ์ (a_{n-i+1}) สำหรับการทดสอบ W ของการแจกแจงปกติ สำหรับ n = 50

i	a_{n-i+1}	i	a_{n-i+1}
1	0.3751	14	0.0846
2	0.2574	15	0.0764
3	0.2260	16	0.0685
4	0.2032	17	0.0608
5	0.1847	18	0.0532
6	0.1691	19	0.0459
7	0.1554	20	0.0386
8	0.1430	21	0.0314
9	0.1317	22	0.0244
10	0.1212	23	0.0174
11	0.1113	24	0.0104
12	0.1020	25	0.0035
13	0.0932		

ตารางที่ 3 แสดง Percentage point ของการทดสอบ W สำหรับ n = 20 และ 50

n	ระดับนัยสำคัญ								
	0.01	0.02	0.05	0.10	0.50	0.90	0.95	0.98	0.99
20	.869	.884	.905	.920	.959	.979	.983	.986	.968
50	.930	.938	.947	.955	.974	.985	.988	.990	.991

ตารางที่ 4 แสดงค่า b โดย $b = m'/\sqrt{m'm}$ ใช้กับขนาดของ n = 100

k	b	k	b	k	b	k	b	k	b
1	0.2543	11	0.1267	21	0.0833	31	0.0510	41	0.0243
2	0.2178	12	0.1213	22	0.0798	32	0.0487	42	0.0217
3	0.1974	13	0.1163	23	0.0764	33	0.0459	43	0.0191
4	0.1827	14	0.1115	24	0.0698	34	0.0431	44	0.0166
5	0.1711	15	0.1070	25	0.0698	35	0.0404	45	0.0161
6	0.1613	16	0.1026	26	0.0666	36	0.0376	46	0.0140
7	0.1529	17	0.0985	27	0.0635	37	0.0349	47	0.0114
8	0.1454	18	0.0945	28	0.0605	38	0.0322	48	0.0089
9	0.1386	19	0.0907	29	0.0575	39	0.0296	49	0.0038
10	0.1324	20	0.0869	30	0.0545	40	0.0269	50	0.0013

หมายเหตุ ค่าของ m หาได้จากหนังสือวารสารของ H.LEON HARTER ชื่อ Expected Value of normal order statistic วารสาร Biometrika (1961), 48, หน้า p.151.

ในการทดสอบนัยสำคัญในที่นี้ Shapiro และ Francia ได้สร้างตารางเริ่มต้น n = 35 ถึง n = 99 กระโดดที่ละ 2 ขั้น สำหรับกรณีเมื่อต้องการขนาดตัวอย่าง n = 100 และ n = 150 ผู้วิจัยจึงได้สร้างตารางค่า b และตารางของ Percentage Point ของการทดสอบ w สำหรับ n = 100 และ 150 ขึ้น โปรแกรมที่เขียนขึ้นกระทำโดยการหา moving average ที่ละ 7 period ได้ผลดังนี้

ตารางที่ 5 แสดง Percentage Point ของการทดสอบ w สำหรับ n = 100และ 150

n	ระดับนัยสำคัญ			
	0.01	0.05	0.10	0.50
100	0.968	0.976	0.981	0.990
150	0.985	0.987	0.991	0.995

ตารางที่ 6 แสดงค่า b โดย $b = m'/\sqrt{m'm}$ ใช้กับขนาดของ n = 150

k	b	k	b	k	b	k	b	k	b
1	0.2184	16	0.1039	31	0.0683	46	0.0424	61	0.0201
2	0.1901	17	0.1009	32	0.0664	47	0.0408	62	0.0187
3	0.1744	18	0.0980	33	0.0645	48	0.0393	63	0.0173
4	0.1632	19	0.0953	34	0.0626	49	0.0377	64	0.0159
5	0.1544	20	0.0927	35	0.0608	50	0.0362	65	0.0145
6	0.1471	21	0.0901	36	0.0590	51	0.0347	66	0.0131
7	0.1408	22	0.0877	37	0.0572	52	0.0332	67	0.0117
8	0.1352	23	0.0853	38	0.0555	53	0.0317	68	0.0103
9	0.1302	24	0.0830	39	0.0538	54	0.0302	69	0.0090
10	0.1256	25	0.0807	40	0.0521	55	0.0288	70	0.0076
11	0.1213	26	0.0785	41	0.0504	56	0.0273	71	0.0062
12	0.1174	27	0.0764	42	0.0488	57	0.0259	72	0.0048
13	0.1137	28	0.0743	43	0.0472	58	0.0244	73	0.0034
14	0.1103	29	0.0722	44	0.0455	59	0.0230	74	0.0021
15	0.1070	30	0.0702	45	0.0440	60	0.0216	75	0.0007

ภาคผนวก ค.

ภาคผนวก ค.

```
C***************************************************************
C************************ MAIN PROGRAM **************************
C***************************************************************
      DOUBLE PRECISION A,S,B,B1,BMMAD,BMMOD,BMMBAR,B2,X,X1,Y1,FL
     * ,T
      COMMON /REGRS/A(11,11),S(12,12)/CONTA/P,IC,SKEWED/ALPHA,BETA
     *       /COEFF/B(11)/COEFF2/BMBAR(11),BMMAD(11),BMMOD(11)
     *       /INTERV/XBAR1(11),SG(11)/SELECT/KK/COEFF3/B1(11)
     *       /DATAXY/X(12,150)/SEED/IX,IX2/VARIAB/N,M,M2
     *       /DIST/II/MEANSQ/SD(3)/DATXY1/X1(12,150)
     *       /TABLE/A2(150),SW(9)/TRANSF/Y1(150),FL(3),ST,FIN
     *       /MSE/AMSE(3),AMSE2(3)
      DIMENSION B2(11),STD(3),T(150),S11(2),S12(2),S13(2),STDD(3),
     *  SS1(2),SS2(2),SS3(2),XX1(2),XX2(2),XX3(2),SSD1(2),SSD2(2)
     *  ,SSD3(2),IK(3)
      DO  1 I   = 1,3
      AMSE2(I) = 0.0
      STD(I)   = 0.0
      IK(I)    = 0.0
    1 AMSE(I)  = 0.0
      DO  3 I  = 1,2
      S11(I)   = 0.0
      S12(I)   = 0.0
      S13(I)   = 0.0
      SS1(I)   = 0.0
      SS2(I)   = 0.0
      SS3(I)   = 0.0
      SSD1(I)  = 0.0
      SSD2(I)  = 0.0
    3 SSD3(I)  = 0.0
```

```
      DO  5 I = 1,4
      READ (5,10) SW(I)
   10 FORMAT(F6.3)
    5 CONTINUE
      READ(5,15)N,M
   15 FORMAT (I3,I2)
      M2        = M+1
      IX        = 973253
      KK        = 0
      K         = INT(FLOAT(N/2))
      JJ        = 100
      DO 16 I = 1,K
      READ (5,17)A2(N-I+1)
   17 FORMAT(F6.4)
   16 CONTINUE
      DO 20 I = 1,M
   20 READ(5,25) B(I),XBAR1(I),SG(I)
   25 FORMAT(F10.4,2F7.4)
      WRITE(6,26)N,M
   26 FORMAT(10X,' NO.OF OBSERVATION = ',I3,' NO.OF VARIABLE
     * = ',I2)
      WRITE(6,27)(B(I),XBAR1(I),SG(I),I=1,M)
   27 FORMAT(F10.4,2F7.4)
      READ(5,30)II
   30 FORMAT (I1)
      GOTO (60,70,80),II
   60 WRITE(6,61)
   61 FORMAT(34X,'#### BETA FROM LOG NORMAL DIST. ####')
      GOTO 105
   70 WRITE(6,71)
```

```
 71 FORMAT(34X,'#### BETA FROM GAMMA DISTRIBUTION ####')
    READ(5,95) ALPHA,BETA
 95 FORMAT (2F5.2)
    WRITE (6,96)ALPHA,BETA
 96 FORMAT(33X,'     ALPHA = ',F5.2,' & BETA = ',F5.2)
 80 WRITE(6,81)
 81 FORMAT(34X,'#### BETA FROM WEIBULL DISTRIBUTION ####')
    READ(5,100) ALPHA,BETA
100 FORMAT (2F5.2)
    WRITE (6,101)ALPHA,BETA
101 FORMAT(33X,'     ALPHA = ',F5.2,' & BETA = ',F5.2)
105 DO 130 I = 1,JJ
    WRITE(6,106) I
106 FORMAT(10X,I3,'TH ROUND',/10X,-------------')
    CALL INIT
    CALL DATA
    DO 1000 J = 1,M2
    DO 1000 K = 1,N
1000 X1(J,K)    = X(J,K)
    CALL BOXCOX(B1,FL1)
    F1         = F1 + FL1
    FLL1       = FLL1 + FL1**2
    FL(I)      = FL1
    CALL BCOX(FL(I),I)
    DO 210  J = 1,N
210 T(J)       = X1(M2,J)
    CALL SHAPWK(N,T,ITEST)
    IF(ITEST.EQ.2) GOTO 220
    III        = II + 1
220 CALL SEE(B1,SD(1))
```

```
      CALL MEST2
      DO 425 J = 1,3
      AMSE2(J) = AMSE2(J) + SD(J) ** 2
  425 AMSE(J)  = AMSE(J) + SD(J)
      CALL ADMSE(IK,S11,S12,S13,SS1,SS2,SS3)
  130 CONTINUE
      DO 450 I = 1,3
      STD(I) = (AMSE2(I) - AMSE(I) **2/JJ)/(JJ-1)
  450 AMSE(I) = AMSE(I)/JJ
      WRITE(6,475)(AMSE(I),I=1,3)
  475 FORMAT('===> AVERAGE MEAN SQUARE ERROR FOR B(I)    ',
     * 3(1XF10.4))
      WRITE(6,476)(STDD(I),I=1,3)
  476 FORMAT('===> STANDARD ERROR OFMEAN SQUARE FOR B(I) *',
     * 3(1XF10.4))
      CALL ARMS
      DO 4 I =  1,2
      IF (IK(1).EQ.0.OR.(IK(I)-1).EQ.0) GOTO 899
        XX1(I) = S11(I)/FLOAT(IK(1))
        SSD1(I) = SQRT(SS1(I) - S11(I)*XX1(I)))/FLOAT(IK(1)-1
      IF (IK(3).EQ.0.OR.(IK(3)-1).EQ.0) GOTO 4
        XX1(I) = S12I)/FLOAT(IK(3))
        SSD1(I) = SQRT(SS1(I) - S11(I)*XX1(I)))/FLOAT(IK(1)-1
    4 CONTINUE
      DO 904 I = 1,2
      WRITE (6,905)XX1(I),XX2(I),XX3(I)
  904 WRITE (6,906)SSD1(I),SSD2(I),SSD3(I)
  905 FORMAT ('===> SD.(STD.DEV)           * ',3(1X,F15.5))
  906 FORMAT ('===> AVERAGE SUM OF DIFF   * ',3(1X,F15.5))
      WRITE (6,907)IK(1),IK(2),IK(3)
```

```
  907 FORMAT ('IK1 = ',I3,' IK2 = ',I3,' IK3 = ',I3)
      DO 950 I = 1,2
  950 WRITE(6,951) S11(I),S12(I),S13(I)
  951 FORMAT ('===> SUMS(I) ON I ',3(1X,F15.5))
      DO 952 I = 1,2
  952 WRITE(6,953) SS1(I),SS2(I),SS3(I)
  953 FORMAT ('===> SUMS SQUARE SS(I) ON I ',3(1X,F15.5))
      STOP
      END
C*********************************************************************
C****************** SUBROUTINE CREATE VARIABLE **********************
C*********************************************************************
      SUBROUTINE INIT
      REAL NORMAL
      DOUBLE PRECISION B,X,A2,SW,A,S,B1
      COMMON /SEED/IX,IX2/DATAXY/X(12,150)/COEFF/B(11)
     *       /COEFF3/B1(11)/SKEWED/ALPHA,BETA
     *       /CONTA/P,IC/INTERV/XBAR1(11),SG(11)
     *       /SELECT/KK/VALIAB/N,M,M2/TABLE/A2(150),SW(9)
     *       /REGRS/A(11,11),S(12,12)/SKEWED/ALPHA,BETA
      DO 90 J = 1,N
   90 X(1,J)  = 1.0
      DO 100 I = 2,M
      DMEAN   = XBAR1(I - 1)
      SIGMA   = SG(I - 1)
      DO 100 J = 1,N
      X(I,J)   = NORMAL(DMEAN,SIGMA)
  100 CONTINUE
C*********************************************************************
C****************** CALCULATE X'X & X'Y FOR  OLS ********************
```

```
C***********************************************************************
      DO 120 I = 1,M
      DO 120 K = 1,M
      SIK      = 0.0
      DO 110 J = 1,N
  110 SIK      = SIK + X(I,J) * X(K,J)
      S(I,K)   = SIK
  120 S(K,I)   = SIK
C***********************************************************************
C***************** CALCULATE INV. METRIX OF X'X ******************
C***********************************************************************
      DO 140 I = 1,M
      DO 140 J = 1,M
      A(I,J)   = S(I,J)
  140 A(J,I)   = S(I,J)
      DO 145 K = 1,M
      IF (A(K,K)) 145,146,145
  146 WRITE(6,150)
  150 FORMAT('A(K,K) HAS ZERO IN DIAGONAL CANNOT USE THIS
     * METRIX')
      STOP
  145 CONTINUE
      CALL INVS(M,A)
      RETURN
      END
C***********************************************************************
C****************** CREATE OUTLINER DISTRIBUTION ******************
C***********************************************************************
      SUBROUTINE DATA
      REAL NORMAL
```

```
      DOUBLE PRECISION B,X,Y,A2,SW,E,T,B1
      COMMON /COEFF/B(11)/SEED/IX,IX2/SELECT/KK/DIST/II
     *       /CONTA/P,IC/INTREV/XBAR1(11),SG(11)
     *       /DATAXY/X(12,150)/DATAY/Y(150)/VARIAB/N,M,M2
     *       /TABLE/A2(150),SW(9)/COEFF3/B1(11)
     *       /SKEWED/ALPHA,BETA
      DIMENSION E(150),T(150)
      GOTO (10,20,30)II
   10 DMEAN     = XBAR1(M)
      SIGMA     = SG(M)
      SG2       = IC * SIGMA
      DO 15 J   = 1,N
      CALL RAND(IX,IY,YFL)
      IF (YFL - P) 11,11,12
   11 E(J)      = NORMAL(DMEAN,SG2)
      GOTO 15
   12 E(J)      = NORMAL(DMEAN,SIGMA)
   15 CONTINUE
      GOTO 60
   20 DMEAN     = XBAR1(M)
      NDF       = N - M
      SIGMA     = SG(M)
      DO 25 J   = 1,N
      E(J)      = TDIST(NDF,DMEAN,SIGMA)
   25 CONTINUE
      DO 106 J = 1,N
      SUM       = 0.0
      DO 105 I = 1,M
      SUM       = SUM + X(I,J) * B(I)
  105 CONTINUE
```

```
      X(M2,J)  = SUM + E(J)
  106 CONTINUE
      WRITE(6,125)
  125 FORMAT(30X,'** MATRIX X & VECTOR Y **')
      DO 130 J = 1,N
  130 WRITE(6,135)(X(I,J),I = 1,M2)
  135 FORMAT(12(2X,F11.4))
      RETURN
      END
C*****************************************************************
C************************* SUBROUTINE RANDOM ********************
C*****************************************************************
      SUBROUTINE RAND(IX,IY,YFL)
      IY    = IX*65539
      IF (IY) 5,6,6
    5 IY    = IY + 2147483647 + 1
    6 YFL   = IY
      YFL   = YFL/2147483647
      IX    = IY
      RETURN
      END
C*****************************************************************
C******************* FUNCTION NORMAL DISTRIBUTION ****************
C*****************************************************************
      FUNCTION NORMAL(DMEAN,SIGMA)
      REAL NORMAL
      COMMON /SEED/IX,IX2/SELECT/KK
      PI     = 3.1415926
      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 T-DISTRIBUTION *********************
C******************************************************************
       FUNCTION TDIST(NDF,DMEAN,SIGMA)
       REAL NORMAL
       SQNOR      = 0.0
       DO 10 I    = 1,NDF
    10 SQNOR      = SQNOR + (NORMAL(DMEAN,SIGMA)**2)
       CHISQ      = SQRT(SQNOR/FLOAT(NDF))
       TDIST      = NORMAL(DMEAN,SIGMA)/CHISQ
       RETURN
       END
C******************************************************************
C****************** FUNCTION WEIBULL DISTRIBUTION ****************
C******************************************************************
       FUNCTION GAMMA(ALPHA1,BETA1)
       COMMON /SEED/IX,IX2
       ALPHA      = ALPHA1
```

```
      U        = 0.0
    5 CALL RAND(IX,IY,YFL)
      V        = -ALOG(YFL)
      U        = U+V
      IF(ALPHA.EQ1.0) GOTO 10
         ALPHA = ALPHA - 1.0
         GOTO 5
   10 GAMMA1   = BETA1*U
      RETURN
      END
C**********************************************************************
C*********************** SUBROUTINE LEAST SQUARE (B) *****************
C**********************************************************************
      SUBROUTINE OLS(B2)
      DOUBLE PRECISION A,S,X,B2
      COMMON /REGRS/A(11,11),S(12,12)/VARIAB/N,M,M2
     *       /DATAXY/X(12,150)
      DIMENSION B2(11)
      DO 20 I = 1,M2
      SIK     = 0.0
      DO 10 J = 1,N
   10 SIK     = SIK + X(I,J) * X(M2,J)
   20 S(M2,1) = SIK
      DO 50 I = 1,M
   50 B2(I)   = B2(I) + A(J,I) * S(M2,J)
      RETURN
      END
C**********************************************************************
C*********** SUBROUTINE FOR M-ESTIMATOR WITH CHANGE SCALE ***********
C**********************************************************************
```

```
      SUBROUTINE MEST2
      DOUBLE PRECISION A1,S1,X1,W,S2,Z,B,B1,YHAT,YRES,BMMAD,
     *       X,BMMOD,BMBAR,SIGMA,SIGMA1
      COMMON /REGRS1/A1(11,1),S1(12,12)/DATAXY/X(12,150)
     *       /VARIAB/N,M,M2/WEIGHT/W(150)/MEANSQ/SD(3)
     *       /DATXY1/X1(12,150)/COEFF/B(11)/COEFF3/B1(11)
      REAL MEDEAN,MEAN1
      DIMENSION S2(150),Z(150),YHAT(150),YRES(150)
     *         ,BMMAD(11),BMMOD(11),BMBAR(11)
      DO 10 I = 1,M2
      DO 10 J = 1,N
   10 X1(I,J) = X(I,J)
      DO 20 I = 1,M2
      DO 20 J = 1,N
   20 X1(I,J) = X(I,J)
      CALL BOXC2(BMMAD,FL1,SIGMA)
      CALL SEE (BMMAD,SD(2))
      SIGMA1  = SIGMA
      KCC = 10
      DO 40 I = 1,M2
      DO 40 J = 1,N
   40 X1(I,J) = X(I,J)
      CALL BOXC3(BMMOD,FL1,KCC,SIGMA1,SIGMA,BMMAD)
      CALL SEE(BMMOD,SD(3))
   30 CONTINUE
      RETURN
      END
C**********************************************************************
C************ SUBROUTINE OLS1(B) FOR M-ESTIMATOR *********************
C**********************************************************************
```

```
      SUBROUTINE OLS1(B2)
      DOUBLE PRECISION A1,S1,B2,X,W
      COMMON /REGRE1/A1(11,11),S1(12,12)/VARIAB/N,M,M2
     *       /DATAXY/X(12,150)/WEIGHT/W(150)
      DIMENSION B2(11)
      DO 20 I = 1,M2
      DO 20 K = I,M2
      SIK     = 0.0
      DO 10 J = 1,N
   10 SIK     = SIK + X(I,J) * X(K,J) * W(J)
      S1(1,K) = SIK
   20 S1(K,I) = SIK
      DO 40 I = 1,M
      DO 40 J = 1,M
      A1(I,J) = S1(I,J)
   40 A1(J,I) = S1(I,J)
      DO 45 K = 1,M
      IF (A1(K,K)) 45,46,45
   46 WRITE(6,100)
  100 FORMAT('A(K,K) HAS ZERO  ON DIAGONAL & CANNOT CREATE INVERSE')
      STOP
   45 CONTINUE
      CALL INVS(M,A1)
      DO 50 I = 1,M
      B2(I)   = 0.0
      DO 50 J = 1,M
   50 B2(I)   = B2(I) + A1(J,I) * S1(M2,J)
      RETURN
      END
C*********************************************************************
```

```
C****************** SUBROUTINE COMPUTE Y-HAT *************************
C*********************************************************************
      SUBROUTINE YRESID(X,N,M,YHAT,YRES,B2)
      DOUBLE PRECISION YHAT,YRES,B2,X
      DIMENSION X(12,150),YHAT(150),YRES(150),B2(11)
      M2        = M+1
      DO 10 J   = 1.N
      YHAT(J)   = 0.0
      DO 20 I   = 1,M
   20 YHAT(J)   = YHAT(J) + B2(I) * X(I,J)
   10 YRES(J)   = X(M2,J) - YHAT(J)
      RETURN
      END
C*********************************************************************
C******************* SUBROUTINE INVERST METRIX X'X ******************
C*********************************************************************
      SUBROUTINE INVS(M,A)
      DOUBLE PRECISION A(11,11)
      DO 20 K    = 1,M
      A(K,K)     = -1.0/A(K,K)
      DO 5 I     = 1,M
      IF (I-K) 3,5,3
    3 A(I,K)     = -A(I,K) * A(K,K)
    5 CONTINUE
      DO 10 I    = 1,M
      DO 10 J    = 1,M
      IF ((I-K) * (J-K)) 9,10,9
    9 A(I,J)     = A(I,J) - A(I,K) * A(K,J)
   10 CONTINUE
      DO 20 J    = 1,M
```

```
      IF (J-K)  18,20,18
   18 A(K,J)    = -A(K,J) * A(K,K)
   20 CONTINUE
      DO 25 I   = 1,M
      DO 25 J   = 1,M
   25 A(I,J)    = - A(I,J)
      RETURN
      END
C*********************************************************************
C****** SUBROUTINE BOX & COX FOR TRANFORMATION SKEWED DIST. ********
C*********************************************************************
      SUBROUTINE BOXCOX(B,FL1)
      DOUBLE PRECISION Y1,Y,B,SLG,BMEST,FL
      COMMON /DATAY/Y(Y150)/TRANSF/Y1(150),FL(3),ST,FIN
     *       /VARIAB/N,M,M2/STORE/BMEST(3,11)
      DIMENSION B(11)
      DO 30 J = 1,N
      IF (Y(J)) 20,20,30
   20 WRITE (6,25)
   25 FORMAT('*Y(J) IS NEGATIVE OR ZERO THEN RETURN TO MAIN PROG*')
      RETURN
   30 CONTINUE
      SLG = 0.0
      DO 50 J = 1,N
   50 SLG =SLG + DLOG(Y(J))
      SLG = SLG/N
      G = DEXP(SLG)
      DO 60 J = 1,N
   60 Y1(J) = Y(J)/G
      ST = -1.0
```

```
      FIN = 2.0
      FD = 0.5
      MR = 16
      CALL SUMSQ(ST,SSE1,1)
      CALL SUMSQ(FD,SSE2,2)
      CALL SUMSQ(FIN,SSE3,3)
   70 IF (SSE1.LE.SSE2.AND.SSE1.LE.SSE3) GOTO 71
      IF (SSE2.LE.SSE1.AND.SSE2.LE.SSE3) GOTO 72
      IF (SSE3.LE.SSE1.AND.SSE3.LE.SSE2) GOTO 73
   71 FM = ST
      SSE2 = SSE1
      DO 80 I = 1,M
   80 BMEST (2,I = BMEST(1,I)
      GOTO 100
   72 FM = FD
      GOTO 100
   73 FM = FIN
      SSE2 = SSE3
      DO 90 I = 1,M
   90 BMEST(2,I) = BMEST(3,I)
  100 IF (MR.EQ.1) GOTO 110
      MR = MR/2
      ST = FM - MR * 0.1
      FD = FM
      FIN = FM + MR * 0.1
      CALL SUMSQ(ST,SSE1,1)
      CALL SUMSQ(FIN,SSE3,3)
      GOTO 70
  110 FL1  = FM
      DO 120 I = 1,M
```

```
  120 B(I) = BMEST(2,I)
      FL(1) = FM
      RETURN
      END
C*****************************************************************************
C*************** SUBROUTINE FOR TEST MULTICOLINEARITY ***************
C*****************************************************************************
      SUBROUTINE SHAPWK (N,Y,ITEST)
      DOUBLE PRECISION A2,SW,Y,S2
      COMMON /TABLE/A2(150),SW(9)
      DIMENSION Y(150),S(11)
      CALL RANK(N,Y)
      YSUM = 0.0
      YSS  = 0.0
      DO 5 I = 1,N
      YSUM = YSUM + Y(I)
    5 YSS  = YSS + Y(I) * Y(I)
      S2   = YSS - (YSUM * YSUM / FLOAT (N))
      K    = INT(FLOAT(N/2))
      B    = 0.0
      DO 20 I = 1,K
      JJ = N - 1 + 1
   20 B    = B + A2(JJ) * (Y(JJ)- Y(I))
      W    = B * B/S2
      IF (W-SW(3)) 30,30,40
   30 ITEST = 2
      GOTO 50
   40 ITEST =1
   50 RETURN
      END
```

```
C*********************************************************************
C****************** SUBROUTINE FOR RANKING RESIDUAL *****************
C*********************************************************************
      SUBROUTINE RANK(N,X)
      DOUBLE PRECISION X,T
      DIMENSION X(150)
      N1      = N - 1
      DO 10 I = 1,N1
      II      = I + 1
      DO 10 K = II,N
      IF (X(I).LE.X(K)) GOTO 10
         T    = X(I)
         X(I) = X(K)
         X(K) = T
   10 CONTINUE
      RETURN
      END
C*********************************************************************
C****************** SUBROUTINE FOR RANKING MSE **********************
C*********************************************************************
      SUBROUTINE RANK2(N,X)
      DOUBLE PRECISION X,T
      DIMENSION X(3)
      N1      = N - 1
      DO 10 I = 1,N1
      II      = I + 1
      DO 10 K = II,N
      IF (X(I).LE.X(K)) GOTO 10
         T    = X(I)
         X(I) = X(K)
```

```
      X(K) = T
   10 CONTINUE
      RETURN
      END
C**********************************************************************
C************************* SUBROUTINE BCOX ****************************
C**********************************************************************
      SUBROUTINE BCOX(FLI,K)
      DOUBLE PRECISION Y1,X1,FL,FLI
      COMMON /DATXY1/X1(12,150)
     *       /TRANSF/Y1(150),FL(3),ST,FIN/VARIAB/N,M,M2
      FL(K)  = FLI
      IF (DABS(FL(K))) 5,15,5
    5 DO 10 I = 1,N
   10 X1(M2,I) = DLOG(Y1(I))
      GOTO 30
   15 DO 20 I = 1,N
   20 X1(M2,I) = ((Y1(I) ** FL(K)) - 1.0 / FL(K)
   30 RETURN
      END
C**********************************************************************
C*************** SUBROUTINE TO COMPUTE SUMSQUARE ERROR ***************
C**********************************************************************
      SUBROUTINE SUMSQ(FLX,SSE,K)
      DOUBLE PRECISION Y1,A,S,Y,B,X1,BMEST,W,A1,S1,Z,ZZ,FL
      COMMON /REGRS/A(11,11),S(12,12)
     *       /REGRS1/A1(11,11),S1(12,12)/WEIGHT/W(150)
     *       /DATXY1/X1(12,150)/DATAY/Y(150)/STORE/BMEST(3,11)
     *       /TRANSF/Y1(150),FL(3),ST,FIN/VARIAB/N,M,M2
      DIMENSION B(11),Z(11)
```

```
      FL(1) = FLX
      CALL BCOX(FL(1),1)
      CALL OLS(B)
      DO 15 I = 1,M
   15 Z(I)     = S(M2,I)
      ZZ       = S(M2,M2)
      DO 80 I = 1,M
   80 BMEST(K,I) = B(I)
      SSR      = 0.0
      DO 90 I = 1,M
   90 SSR      = SSR + B(I) * Z(i)
      SSE      = ZZ - SSR
      SSE      = SSE/FLOAT(N)
      RETURN
      END
C*********************************************************************
C**************** M-ESTIMATOR WITH STANDARD DIVIATION ****************
C*********************************************************************
      SUBROUTINE BAR(B1,BMBAR,SIGMA)
      DOUBLE PRECISION E,BMBAR,B2,SUME,Z,YHAT,YRES,W,B1,EXBAR,
     *       X1,SIGMA
      COMMON /DATXY1/X1(12,1500/WEIGHT/W(150)/VARIAB/N,M,M2
      DIMENSION Z(150),YHAT(150),YRES(150),E(150),BMBAR(11),B2(11),
     *          B1(11)
      SUME = 0.0
      CALL YRESID (X1,N,M,YHAT,YRES,B1)
      DO 5 I = 1,N
    5 E(i)    = YRES(I)
      DO 10 J = 1,N
      SUME    = SUME + E(J)
```

```
 10 CONTINUE
    EXBAR    = SUME/N
    SIGMA1   = 0.0
    DO 20 J = 1,N
    SIGMA1   = SIGMA1 + (E(J)- EXBAR)**2
 20 CONTINUE
    SIGMA    = SQRT(SIGMA1)
    L = 0
    CALL YRESID(X1,N,M,YHAT,YRES,B1)
    DO 25 I = 1,M
 25 B2(I)    = B1(I)
777 DO 30 I = 1,N
    Z(I)     = DABS(YRES(I)/SIGMA)
    W(I)     = 2.0/Z(I)
    IF (W(I).GE.2.0) GOTO 555
       GOTO 30
555 W(I) = 1.0
 30 CONTINUE
    L    = L + 1
    IF (L-20) 60,60,90
 60 CALL OLS1(BMBAR)
    DO 70 I = 1,M
    C        = DABS(B2(I)-BM BAR(I)/DABS(I))
    IF (C-0.001) 70,70,75
 75 CALL YRESID(X1,N,M,YHAT,YRES,BMBAR)
    DO 80 J = 1,M
 80 B2(J)    = BMBAR(J)
    GOTO 777
 70 CONTINUE
 90 RETURN
```

```
      END
C***********************************************************************
C****************** M-ESTIMATOR WITH MEDIAN ABSOLUTE *******************
C***********************************************************************
      SUBROUTINE MAD(B1,BMMAD,SIGMA)
      DOUBLE PRECISION E,BMMAD,B2,SUME,Z,YHAT,YRES,W,B1,SIG1,
     *       X1,SIGMA
      COMMON /DATXY1/X1(12,1500/WEIGHT/W(150)/VARIAB/N,M,M2
      DIMENSION Z(150),YHAT(150),YRES(150),E(150),BMMAD(11),B2(11),
     *          B1(11),SIG1(150)
      REAL MEDIAN
      CALL YRESID (X1,N,M,YHAT,YRES,B1)
      DO 5 I = 1,N
    5 E(I)    = YRES(I)
      N1      = INT(FLOAT(N/2)) * 2
      N2      = (N+1)/2
      N3      = INT(FLOAT(N/2))
      N4      = N3+1
      CALL    RANK(N,E)
      IF (N-N1) 10,15,10
   10 MEDIAN = E(N2)
      GOTO 18
   15 MEDIAN = (E(N3)+E(N4))/2
   18 DO 20 I = 1,N
   20 SIG1(I) = DABS(E(I)-MEDIAN)
      CALL RANK(N,SIG1)
      IF (N-N1) 25,27,25
   25 SIGMA = SIG1(N2)/0.6745
      GOTO 28
   27 SIGMA = (SIG1(N3)+SIG1(N4))/(2*0.6745)
```

```
   28 L = 0
      CALL YRESID(X1,N,M,YHAT,YRES,B1)
      DO 29 I = 1,M
   29 B2(I)   = B1(I)
  777 DO 30 I = 1,N
      Z(I)    = DABS(YRES(I)/SIGMA)
      W(I)    = 2.0/Z(I)
      IF (W(I).GE.2.0) GOTO 555
         GOTO 30
  555 W(I) = 1.0
   30 CONTINUE
      L     = L + 1
      IF (L-20) 60,60,90
   60 CALL OLS1(BMBAR)
      DO 70 I = 1,M
      C       = DABS(B2(I)-BM BAR(I)/DABS(I))
      IF (C-0.001) 70,70,75
   75 CALL YRESID(X1,N,M,YHAT,YRES,BMBAR)
      DO 80 J = 1,M
   80 B2(J)   = BMBAR(J)
      GOTO 777
   70 CONTINUE
   90 RETURN
      END
C**********************************************************************
C********* M-ESTIMATOR WITH MODIFIED BIWEIGHT A-ESTIMATES *************
C**********************************************************************
      SUBROUTINE MMOD(BMMAD,KCC,SIGMA1,BMMOD,SIGMA)
      DOUBLE PRECISION YHAT,YRES,B2,BMMOD,BMMAD,WB,W,
     *       U,SU,Z,SIGMA1,E,C,X1,SIGMA
```

```
      COMMON /DATXY1/X1(12,150)/WEIGHT/W(150)/VARIAB/N,M,M2
      DIMENSION WB(150),U(150),BMMAD(11),B2(11),BMMOD(11),YRES(150),
     *       YHAT(150),Z(150)
      CALL YRESID(X1,N,M,YHAT,YRES,BMMAD)
      WEB      = 0.0
      SUMU     = 0.0
      DO 20 I = 1,N
      U(I)     = DABS(YRES(I)/(KCC*SIGMA1))
      IF (U(I)-1) 30,40,40
   30 WB(I)    = (1-U(I)**2)**2
      WEB      = WEB + WB(I)
      SU       = (YRES(I)**2*WB(I))**2
      SUMU     = SUMU + SU
      GOTO 20
   40 U(I)     = 0.0
      WB(I)    = 0.0
   20 CONTINUE
      SQU      = SQRT(SUMU)
      SIGMA    = (N/SQRT(FLOAT(N-1)))*(SQU/WEB)
      L        = 0
      CALL YRESID(X1,N,M,YHAT,YRES,BMMAD)
      DO 50 I = 1,M
   50 B2(I)    = BMMAD(I)
  777 DO 60 I = 1,N
      Z(I)     = DABS(YRES(I)/SIGMA)
      W(I)     = 2.0/Z(I)
      IF (W(I).GE.2.0) GOTO 555
         GOTO 30
  555 W(I) = 1.0
   60 CONTINUE
```

```
      L        = L+1
      IF (L-20)  70,70,100
   70 CALL OLS1(BMMOD)
      DO 80 I = 1,M
      C        = DABS(B2(I)-BMMOD(I))/DABS(B2(I))
      IF (C-0.001) 80,80,85
   85 CALL YRESID(X1,N,M,YHAT,YRES,BMMOD)
      DO 90 J = 1,M
   90 B2(J)    = BMMOD(J)
      GOTO 777
   80 COTINUE
  100 RETURN
      END
C*******************************************************************
C********* SUBROUTINE  FOR  MEANSQUARE ERROR ***********************
C*******************************************************************
      SUBROUTINE SEE(B2,SE)
      DOUBLE PRECISION B,B2
      COMMON /COEFF/B(11)/VARIAB/N,M,M2
      DIMENSION B2(11)
      SE       = 0.0
      DO 10 I = 1,M
   10 SE       = SE + (B(I)-B2(I))**2
      SE       = SE/M
      RETURN
      END
C*******************************************************************
C************** BOX & COX FOR STANDARD VARIABLE ********************
C*******************************************************************
      SUBROUTINE BOXC1(BMBAR,FL1,SIGMA)
```

```
      DOUBLE PRECISION Y1,BMBAR,BMEST,Y,A,A1,S,S1,X1,SIGMA,FL
      COMMON /REGRES/A(11,11),S(12,12)
     *       /DATXY1/X1(12,150)/DATAY/Y(150)/STORE/BMEST(3,11)
     *       /TRANSF/Y1(150),FL(5),ST,FIN/VARIAB/N,M,M2
      DIMENSION BMBAR(11)
      L = 0
      MR = 3
      ST = FL(1) - 0.1 * MR
      FIN = FL(1) + 0.1 * MR
      FD = FL(1)
      CALL SUMBAR(ST,SSE1,1,SIGMA)
      CALL SUMBAR(FD,SSE2,2,SIGMA)
      CALL SUMBAR(FIN,SSE3,3,SIGMA)
   10 IF (SSE1.LE.SSE2.AND.SSE1.LE.SSE3) GOTO 20
      IF (SSE2.LE.SSE1.AND.SSE2.LE.SSE3) GOTO 30
      IF (SSE3.LE.SSE1.AND.SSE3.LE.SSE2) GOTO 40
   20 ST = ST - 0.1 * MR
      FIN = FD - 0.1 * MR
      FD = FIN - 0.1 * MR
      SSE3 = SSE2
      SSE2 = SSE1
      DO 25 I = 1,M
      BMEST(3,I) = BMEST(2,I)
   25 BMEST(2,I) = BMEST(1,I)
      CALL SUMBAR(ST,SSE1,1,SIGMA)
      GOTO 50
   30 MR = 1
      ST = FD - 0.1
      FIN = FD + 0.1
      CALL SUMBAR(ST,SSE1,1,SIGMA)
```

```
      CALL SUMBAR(FIN,SSE3,3,SIGMA)
      GOTO 50
   40 ST = ST + 0.1 * MR
      FIN = FD + 0.1 * MR
      FD = FIN + 0.1 * MR
      SSE1 = SSE2
      SSE2 = SSE3
      DO 45 I = 1,M
      BMEST(1,I) = BMEST(2,I)
   45 BMEST(2,I) = BMEST(3,I)
      CALL SUMBAR(FIN,SSE3,3,SIGMA)
   50 L = L + 1
      IF ((L.GE.16).OR.(MR.EQ.1)) GOTO 60
      GOTO 10
   60 IF (SSE1.LE.SSE2.AND.SSE1.LE.SSE3) GOTO 70
      IF (SSE2.LE.SSE1.AND.SSE2.LE.SSE3) GOTO 80
      IF (SSE3.LE.SSE1.AND.SSE2.LE.SSE1) GOTO 90
   70 FL1 = ST
      K = 1
      GOTO 100
   80 FL1 = FD
      K = 2
      GOTO 100
   90 FL1 = FIN
      K = 3
  100 DO 110 I = 1,M
  110 BMBAR(I) = BMEST(K,I)
      FL(2) = FL1
      RETURN
      END
```

```
C***********************************************************************
C************************* BOX & COX FOR MAD ***************************
C***********************************************************************
      SUBROUTINE BOXC2(BMMAD,FL1,SIGMA)
      DOUBLE PRECISION Y1,BMMAD,BMEST,Y,A,A1,S,S1,X1,SIGMA,FL
      COMMON /REGRES/A(11,11),S(12,12)
     *       /DATXY1/X1(12,150)/DATAY/Y(150)/STORE/BMEST(3,11)
     *       /TRANSF/Y1(150),FL(5),ST,FIN/VARIAB/N,M,M2
      DIMENSION BMMAD(11)
      L = 0
      MR = 3
      ST = FL(1) - 0.1 * MR
      FIN = FL(1) + 0.1 * MR
      FD = FL(1)
      CALL SUMMAD(ST,SSE1,1,SIGMA)
      CALL SUMMAD(FD,SSE2,2,SIGMA)
      CALL SUMMAD(FIN,SSE3,3,SIGMA)
   10 IF (SSE1.LE.SSE2.AND.SSE1.LE.SSE3) GOTO 20
      IF (SSE2.LE.SSE1.AND.SSE2.LE.SSE3) GOTO 30
      IF (SSE3.LE.SSE1.AND.SSE3.LE.SSE2) GOTO 40
   20 ST = ST - 0.1 * MR
      FIN = FD - 0.1 * MR
      FD = FIN - 0.1 * MR
      SSE3 = SSE2
      SSE2 = SSE1
      DO 25 I = 1,M
      BMEST(3,I) = BMEST(2,I)
   25 BMEST(2,I) = BMEST(1,I)
      CALL SUMMAD(ST,SSE1,1,SIGMA)
      GOTO 50
```

```
 30 MR = 1
    ST = FD - 0.1
    FIN = FD + 0.1
    CALL SUMMAD(ST,SSE1,1,SIGMA)
    CALL SUMMAD(FIN,SSE3,3,SIGMA)
    GOTO 50
 40 ST = ST + 0.1 * MR
    FIN = FD + 0.1 * MR
    FD = FIN + 0.1 * MR
    SSE1 = SSE2
    SSE2 = SSE3
    DO 45 I = 1,M
    BMEST(1,I) = BMEST(2,I)
 45 BMEST(2,I) = BMEST(3,I)
    CALL SUMMAD(FIN,SSE3,3,SIGMA)
 50 L = L + 1
    IF ((L.GE.16).OR.(MR.EQ.1)) GOTO 60
    GOTO 10
 60 IF (SSE1.LE.SSE2.AND.SSE1.LE.SSE3) GOTO 70
    IF (SSE2.LE.SSE1.AND.SSE2.LE.SSE3) GOTO 80
    IF (SSE3.LE.SSE1.AND.SSE2.LE.SSE1) GOTO 90
 70 FL1 = ST
    K = 1
    GOTO 100
 80 FL1 = FD
    K = 2
    GOTO 100
 90 FL1 = FIN
    K = 3
100 DO 110 I = 1,M
```

```
  110 BMMAD(I) = BMEST(K,I)
      FL(3) = FL1
      RETURN
      END
C*******************************************************************
C**************** BOX & COX FOR MODIFIED A-ESTIMATOR **************
C*******************************************************************
      SUBROUTINE BOXC3(BMMOD,FL1,,IZZ,KCC,SIGMA1,SIGMA,BMMAD)
      DOUBLE PRECISION Y1,BMMOD,BMEST,Y,A,A1,S,S1,X1,SIGMA,FL,
     *       SIGMA1,BMMAD
      COMMON /REGRES/A(11,11),S(12,12)
     *       /DATXY1/X1(12,150)/DATAY/Y(150)/STORE/BMEST(3,11)
     *       /TRANSF/Y1(150),FL(5),ST,FIN/VARIAB/N,M,M2
      DIMENSION BMMAD(11),BMMOD(11)
      L = 0
      MR = 3
      ST = FL(1) - 0.1 * MR
      FIN = FL(1) + 0.1 * MR
      FD = FL(1)
      CALL SUMMOD(ST,SSE1,1,BMMAD,SIGMA1,KCC,IZZ,SIGMA)
      CALL SUMMOD(FD,SSE2,2,BMMAD,SIGMA1,KCC,IZZ,SIGMA)
      CALL SUMMOD(FIN,SSE3,3,BMMAD,SIGMA1,KCC,IZZ,SIGMA)
   10 IF (SSE1.LE.SSE2.AND.SSE1.LE.SSE3) GOTO 20
      IF (SSE2.LE.SSE1.AND.SSE2.LE.SSE3) GOTO 30
      IF (SSE3.LE.SSE1.AND.SSE3.LE.SSE2) GOTO 40
   20 ST = ST - 0.1 * MR
      FIN = FD - 0.1 * MR
      FD = FIN - 0.1 * MR
      SSE3 = SSE2
      SSE2 = SSE1
```

```
      DO 25 I = 1,M
      BMEST(3,I) = BMEST(2,I)
   25 BMEST(2,I) = BMEST(1,I)
      CALL SUMMOD(ST,SSE1,1,BMMAD,SIGMA1,KCC,IZZ,SIGMA)
      GOTO 50
   30 MR = 1
      ST = FD - 0.1
      FIN = FD + 0.1
      CALL SUMMOD(ST,SSE1,1,BMMAD,SIGMA1,KCC,IZZ,SIGMA)
      CALL SUMMOD(FIN,SSE3,3,BMMAD,SIGMA1,KCC,IZZ,SIGMA)
      GOTO 50
   40 ST = ST + 0.1 * MR
      FIN = FD + 0.1 * MR
      FD = FIN + 0.1 * MR
      SSE1 = SSE2
      SSE2 = SSE3
      DO 45 I = 1,M
      BMEST(1,I) = BMEST(2,I)
   45 BMEST(2,I) = BMEST(3,I)
      CALL SUMMOD(FIN,SSE3,3,BMMAD,SIGMA1,KCC,IZZ,SIGMA)
   50 L = L + 1
      IF ((L.GE.16).OR.(MR.EQ.1)) GOTO 60
      GOTO 10
   60 IF (SSE1.LE.SSE2.AND.SSE1.LE.SSE3) GOTO 70
      IF (SSE2.LE.SSE1.AND.SSE2.LE.SSE3) GOTO 80
      IF (SSE3.LE.SSE1.AND.SSE2.LE.SSE1) GOTO 90
   70 FL1 = ST
      K = 1
      GOTO 100
   80 FL1 = FD
```

```
      K = 2
      GOTO 100
   90 FL1 = FIN
      K = 3
  100 DO 110 I = 1,M
  110 BMMOD(I) = BMEST(K,I)
      FL(IZZ) = FL1
      RETURN
      END
C*********************************************************************
C*************** SUMSQUARE FOR STANDARD DIVIATION ******************
C*********************************************************************
      SUBROUTINE SUMBAR(FLX,SSE,K,SIGMA)
      DOUBLE PRECISION BMEST,BMBAR,A,A1,S,S1,X1,Y,Y1,W,Z,ZZ,
     *       SIGMA,FL,B1,B
      COMMON /STORE/BMEST(3,11)/VARIAB/N,M,M2/DATAY/Y(150)
     *       /REGRS1/A1(11,11),S1(12,12)/WEIGHT/W(150)
     *       /TRANSF/Y1(150),FL(5),ST,FIN/DATXY1/X1(12,150)
     *       /REGRS/A(11,11),S(12,12)/COEFF/B(11)/COEFF3/B1(11)
      DIMENSION BMBAR(11),Z(11)
      FL(2) = FLX
      CALL BCOX(FL(2),2)
      CALL OLS(B1)
      CALL BAR(B1,BMBAR,SIGMA)
      DO 10 I = 1,M
   10 Z(I) = S1(M2,I)
      ZZ   = S1(M2,M2)
      DO 20 I = 1,M
   20 BMEST(K,I) = BMBAR(I)
      SSR = 0.0
```

```
      DO 30 I = 1,M
   30 SSR = SSR + BMBAR(I) * Z(I)
      SSE = ZZ - SSR
      SSE = SSE/FLOAT(N)
      RETURN
      END
C*********************************************************************
C*************** SUMSQUARE FOR MEDIAN ABSOLUTE *********************
C*********************************************************************
      SUBROUTINE SUMMAD(FLX,SSE,K,SIGMA)
      DOUBLE PRECISION BMEST,BMMAD,A,A1,S,S1,X1,Y,Y1,W,Z,ZZ,
     *       SIGMA,FL,B1,B
      COMMON /STORE/BMEST(3,11)/VARIAB/N,M,M2/DATAY/Y(150)
     *       /REGRS1/A1(11,11),S1(12,12)/WEIGHT/W(150)
     *       /TRANSF/Y1(150),FL(5),ST,FIN/DATXY1/X1(12,150)
     *       /REGRS/A(11,11),S(12,12)/COEFF/B(11)/COEFF3/B1(11)
      DIMENSION BMMAD(11),Z(11)
      FL(3) = FLX
      CALL BCOX(FL(3),3)
      CALL OLS(B1)
      CALL BAR(B1,BMMAD,SIGMA)
      DO 10 I = 1,M
   10 Z(I) = S1(M2,I)
      ZZ   = S1(M2,M2)
      DO 20 I = 1,M
   20 BMEST(K,I) = BMMAD(I)
      SSR = 0.0
      DO 30 I = 1,M
   30 SSR = SSR + BMMAD(I) * Z(I)
      SSE = ZZ - SSR
```

```
      SSE = SSE/FLOAT(N)
      RETURN
      END
C**********************************************************************
C*********** SUMSQUARE FOR MODIFIED A-ESTIMATOR ******************
C**********************************************************************
      SUBROUTINE SUMMOD(FLX,SSE,K,BMMAD,SIGMA1,KCC,IZZ,SIGMA)
      DOUBLE PRECISION BMEST,BMMAD,A,A1,S,S1,X1,Y,Y1,W,Z,ZZ,
     *       SIGMA,FL,B1,B,SIGMA1,BMMOD
      COMMON /STORE/BMEST(3,11)/VARIAB/N,M,M2/DATAY/Y(150)
     *       /REGRS1/A1(11,11),S1(12,12)/WEIGHT/W(150)
     *       /TRANSF/Y1(150),FL(5),ST,FIN/DATXY1/X1(12,150)
     *       /REGRS/A(11,11),S(12,12)/COEFF/B(11)/COEFF3/B1(11)
      DIMENSION BMMAD(11),Z(11),BMMOD(11)
      FL(IZZ) = FLX
      CALL BCOX(FL(IZZ),IZZ)
      CALL MMOD(BMMAD,KCC,SIGMA1,BMMOD,SIGMA)
      DO 10 I = 1,M
   10 Z(I) = S1(M2,I)
      ZZ   = S1(M2,M2)
      DO 20 I = 1,M
   20 BMEST(K,I) = BMMOD(I)
      SSR = 0.0
      DO 30 I = 1,M
   30 SSR = SSR + BMMOD(I) * Z(I)
      SSE = ZZ - SSR
      SSE = SSE/FLOAT(N)
      RETURN
      END
C**********************************************************************
```

```
C************** AVERAGE DIFFERANCE MEAN SQUARE ERROR ***************
C*******************************************************************
      SUBROUTINE ADMSE(IK,S11,S12,S13,SS1,SS2,SS3)
      COMMON /MEANSQ/SD(3)
      DIMENSION  S11(2),S12(2),S13(2),SS1(2),SS2(2),SS3(2),IK(3),
     * DIFF(3,2)
      IF(SD(1).LE.SD(2).AND.SD(1).LE.SD(3)) GOTO 10
      IF(SD(2).LE.SD(1).AND.SD(2).LE.SD(3)) GOTO 20
      DIFF(3,1) = SD(1)/SD(3) - 1.0
      DIFF(3,2) = SD(2)/SD(3) - 1.0
      S13(1) = S13(1) + DIFF(3,1)
      S13(2) = S13(2) + DIFF(3,2)
      SS3(1) = SS3(1) + DIFF(3,1) ** 2
      SS3(2) = SS3(2) + DIFF(3,2) ** 2
      IK(3) = IK(3) + 1
      GOTO 50
   10 DIFF(1,1) = SD(2)/SD(1) - 1.0
      DIFF(1,2) = SD(3)/SD(1) - 1.0
      S11(1) = S11(1) + DIFF(1,1)
      S11(2) = S11(2) + DIFF(1,2)
      SS1(1) = SS1(1) + DIFF(1,1) ** 2
      SS13(2) = SS1(2) + DIFF(1,2) ** 2
      GOTO 50
   20 DIFF(2,1) = SD(1)/SD(2) - 1.0
      DIFF(2,2) = SD(3)/SD(2) - 1.0
      S12(1) = S12(1) + DIFF(2,1)
      S12(2) = S12(2) + DIFF(2,1)
      SS2(1) = SS2(1) + DIFF(2,1) ** 2
      SS2(2) = SS2(2) + DIFF(2,2) ** 2
      IK(2) = IK(2) + 1
```

```
   50 RETURN
      END
C*******************************************************************
C******************* AVERAGE SUM TOTAL MEAN SQUARE *****************
C*******************************************************************
      SUBROUTINE ARMS
      DOUBLE PRECISION AMSE,E,ARDIF
      COMMON /MSE/AMSE(3),AMSE2(3)
      DIMENSION E(3),ARDIF(2)
      DO 10 I = 1,3
   10 E(I) = AMSE(I)
      K = 3
      CALL RANK2(K,E)
      J = 1
      DO 20 I = 2,3
   20 ARDIF(I-1) = (E(I)-E(J))/E(J)
      DO 30 I = 1,2
   30 WRITE (6,40) I,ARDIF(I)
   40 FORMAT('THE  ',I3,'TH LARGE FROM MIN & AVERAGE DIFFERANCE = '
     * ,F13.5)
     RETURN
     END

C*******************************************************************
C*********** E N D               O F               P R O G R A M *********
C*******************************************************************
```

ประวัติผู้เขียน

นายทรงพันธุ์ ชุณหสวัสดิกุล เกิดเมื่อวันที่ 2 กุมภาพันธ์ พ.ศ. 2506 จบการศึกษาระดับปริญญาตรีจากมหาวิทยาลัยธรรมศาสตร์ ในสาขาสถิติ คณะศิลปศาสตร์ เมื่อปี พ.ศ. 2528 และเข้าศึกษาต่อระดับปริญญาโท ในภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2529