บรรณานุกรม

ภาษาไทย

พรรณวดี โงวหจินตนา "การตรวจสอบความเหมาะสมของตัวแบบถดถอยเชิงเส้นโดยการแบ่งข้อมูลด้วยวิธีดูเพล็กซ์" วิทยานิพนธ์ ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2530.

เลิศสรรพ์ เมมสุต "การเปรียบเทียบอำนาจการทดสอบของตัวสถิติทดสอบการเท่ากันของค่าเฉลี่ยของประชากร 2 ชุด ที่มีการแจกแจงชนิดลอง-เทลด์" วิทยานิพนธ์ ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2530.

สมเกียรติ เกตุเอี่ยม "การเปรียบเทียบวิธีประมาณค่าพารามิเตอร์เพื่อการทดสอบการแจกแจงของประชากรที่ให้ค่าสถิติไคสแควร์ต่ำสุด" วิทยานิพนธ์ ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2529.

ภาษาอังกฤษ

Bolch, B.W. and Huang, C.J. Multivariate Statistical for business and Economics. Englewood Clilfs, New Jersey : Prentice-Hall, inc., 1974.

Hoaglin D.C. Mosteller F. and Tukey J.W. Understanding Robust and Exploratory Data Analysis. New York : John Wiley, 1983.

Huber, P.J. Robust Statistics New York; John Wiley, 1981.

Larvless, J.F. Statistical Models and Method for lifetime data. New York : John Wiley, 1982.

Kennedy, W.J. and Gentle, J.E. Statistical Computing. New York and Basel : Marcel Dekker, Inc., 1980.

Montgomery, D.C. and Peck, E.A. Introduction to liner Regression Analysis. New York : John Wiley, 1982.

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

Seber, G.A.F. Linear Regression Analysis. New York : John Wiley, 1976.

Articles

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

Carrol, R.J. "A Robust Method for Testing Transformations to Achieve Approximate Normality." Jornal of the Royal Statistical Society B42(1980) : 71-78.

Carroll, R.J. and Ruppert, David. "Transformations in Regression : A Robust Analysis." Technometrics 27(1985) : 1-12.

Forsythe, A.B. "Robust estimation of straight-line regression coefficients by minimizing p^{th} power deviations." Technometrics 14(1972) : 159-166.

Huber, P.J. "Robust estimation of a location parameter." Annual Mathmatic Statistics 35(1964) : 73-101.

Holland, P.W. and Welsch R.E. "Robust regression using iteratively reweighted least squares." Communication Statistics A6 (1977) : 813-828.

Jeremy M.G. and Taylor "Power Transformations to symmetry." Biometrika 72(1985) : 145-152.

Ramsay, J.O. "A comparative study of several robust estimates of slope, intercept, and scale in linear regression." Journal of the American Statistical Association 72(1977) 608-615.

Shapiro, S.S. and Wilk, M.B. "An analysis of variance test for normal (complete samples)." Biometrika 52(1965) 591-611.

Shapiro, S.S. and Francia, R.S. "An approximate analysis of variance test for nomality. Journal of the American Statistical Association 63(1972) : 215-216.

Shapiro, S.S. Wilk, M.B. and Chen J. "A comparative study of variance." Journal of the American Statistical Association 63(1968) : 1343-1372.

ภาคผนวก

ภาคผนวก ก.

ในการสร้างตัวแปรให้มีคุณสมบัติตามต้องการ วิธีการหนึ่งที่สามารถทำได้คืออาศัยเทคนิคของการผลิตเลขสุ่มโดยการใช้โปรแกรม

1. การผลิตเลขสุ่มโดยการใช้โปรแกรม

เป็นการผลิตเลขสุ่มจากความสัมพันธ์ที่ซ้ำ ๆ (recurrence relation) กล่าวคือ เลขถัดไปเกิดจากการดำเนินการทางคณิตศาสตร์และตรรกศาสตร์ของตัวเลขปัจจุบันหรือกลุ่มตัวเลขในอดีต อนุบรรพ (sequence) ของตัวเลขซึ่งผลิตได้จึงเป็นอนุบรรพของเลขสุ่มในความหมายที่แท้จริงไม่ได้และย่อมมีคาบ (period) แต่อย่างไรก็ตามเลขที่ผลิตในอนุบรรพเหล่านี้อาจจะผ่านการทดสอบความเป็นสุ่มทางสถิติได้หลายอย่างและเรียกว่าเลขคล้ายสุ่ม (Pseudo-Random Number)

การผลิตเลขสุ่มโดยการใช้โปรแกรม มีข้อดีหลายประการที่สำคัญคือ วิธีนี้สามารถผลิตอนุบรรพของเลขชุดเดิมออกมาได้ ซึ่งเป็นสิ่งจำเป็นอย่างยิ่งในกรณีที่ใช้การทดสอบแบบจำลอง และมีความประสงค์จะทบทวนการคำนวณโปรแกรม

ข้อบกพร่องของวิธีการผลิตนี้ก็คือ อนุบรรพของเลขสุ่มที่ผลิตออกมาเป็นอนุบรรพที่มีคาบทั้งสิ้นและการอนุมานคุณสมบัติเชิงสถิติของเลขสุ่มเหล่านี้ทางทฤษฎีกระทำได้ยากพอสมควร

เทคนิคการผลิตเลขสุ่มโดยการใช้โปรแกรมได้รับการพัฒนาอย่างรวดเร็ว Von Neuman และ Metropolis ได้เสนอวิธีตัวกลางกำลังสอง (Mid-Square method) ในปี ค.ศ. 1946 Forsythe ได้ปรับปรุงวิธีตัวกลางกำลังสอง และต่อมา Lehmer ได้เสนอวิธีผลิตเลขสุ่มด้วยการใช้เศษจากการหารผลคูณ (Multiplicative congruential method) ซึ่งเป็นวิธีที่ใช้กันอย่างแพร่หลายในคอมพิวเตอร์ ยุคต่อมา Rotenleerg ได้ปรับปรุงวิธีการผลิตของ Lehmer เป็นการใช้เศษของผลบวกของผลคูณกับค่าคงที่จากการหาร (Mixed congruential method)

วิธีการแบบ Multiplicative congruential method จะหาเลขสุ่มโดยทำการคำนวณจากสมการ

(1) $X_{i+1} \equiv X_i \cdot a \pmod{m}$

เมื่อ X_i เป็นเลขคล้ายสุ่มตัวที่ i

X_{i+1} เป็นเลขคล้ายสุ่มตัวที่ $i + 1$

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) จะได้เลขคล้ายสุ่มที่เป็นเลขจำนวนเต็ม ค่าใดค่าหนึ่งในช่วง 0, 1,.., m-1 หลังจากนั้นแล้วจะได้เลขคล้ายสุ่มชุดเดิมอีก ฉะนั้นคาบของเลขคล้ายสุ่มที่ได้จึงมีค่าไม่เกิน m (คาบของเลขคล้ายสุ่มมีค่าน้อยกว่า m เมื่อเลือกค่า a และ X_0 ไม่ดีนัก) การเลือกค่า m, a และ X_0 จึงมีความสำคัญในการผลิตเลขคล้ายสุ่มที่มีคาบใกล้เคียงกับ m มากที่สุด

Lehmer ได้มีการทดลองเลือกใช้ค่า m, a และ X_0 ที่จับคู่ต่างๆ กันเพื่อใช้ผลิตเลขคล้ายสุ่มตามสมการที่ (1) พบว่าถ้าเลือก X_0 เป็นเลขคี่ และ $m = 2^r$ (เมื่อ $r > 2$) และ $a = 8k \pm 3$ (เมื่อ k เป็นจำนวนเต็มบวกใด ๆ) 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 หรือจำนวน bit ใน 1 word ในการศึกษาครั้งนี้ใช้เครื่อง IBM 3031 system 32 bit binary machine ซึ่ง bit สุดท้าย 1 bit ใช้สำหรับแสดงเครื่องหมาย ดังนั้นเลขจำนวนเต็มที่ใหญ่ที่สุดใน 1 word และเป็นเลขคี่ที่คอมพิวเตอร์ได้รับคือ $2^{b-1}1$ เท่ากับ $2^{31}-1 = 2147483647$ นั่นคือ ค่า m ควรมีค่า = 2147483647

2) ค่า seed (X_0) ควรมีค่าที่เป็น prime กับค่า m(relatively prime to m) เมื่อ m เป็นค่ากำลังของ 2 (จาก $m = 2^b$) ดังนั้น X จึงควรมีค่าเป็นเลขจำนวนเต็มบวกที่เป็นเลขคี่ (ในกรณีที่ใช้ X_0 เป็นเลขคู่จะพบว่าทุก ๆ ค่า X_i ต่อไปจะเป็นเลขคู่เสมอ จึงไม่มีคุณสมบัติเป็นเลขสุ่ม)

3) ค่าคงที่ที่ใช้เป็นตัวคูณ a(constant multiplier) ควรมีค่าเป็น prime กับ m ด้วย นั่นคือ a ต้องเป็นเลขคี่ พบว่าวิธีเลือกที่ดีที่สุด สำหรับค่า a เมื่อใช้ความสัมพันธ์ $\equiv \pm 3$ (mod m) หรือ $a = 8t \pm 3$ เมื่อ t เป็นค่าบวกใด ๆ a จะมีค่าใกล้ $2^{b/2}$ ซึ่ง a จะเป็นเลขอันดับแรกของความสัมพันธ์ระหว่างเลขคล้ายสุ่ม ดังนั้น เครื่อง IBM 3031 system 32 bit binary machine เราเลือกใช้ค่า $a = 2^{16} + 3 = 65539$

ในกรณีของเครื่องคอมพิวเตอร์จะใช้หลักการ Multiplicative congruential method และ Shift register ดังนั้นการคำนวณ aX_0 มีผลคูณที่เกิน fixed length หรือ 1 word โดยผลคูณของเลขจำนวนเต็มจะประกอบด้วย 2b bits จากผลลัพธ์นี้ตัวเลขหลักค่าสูงใน b bit แรกจะถูกตัดทิ้งไปและตัวเลขหลักค่าต่ำ b bit หลังจะแทนด้วย X_1 ค่าตัวเลขซึ่งมีการแจกแจงแบบสม่ำเสมอ คือเศษจากการหาร $r_1 = \frac{X_1}{2^b}$ 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

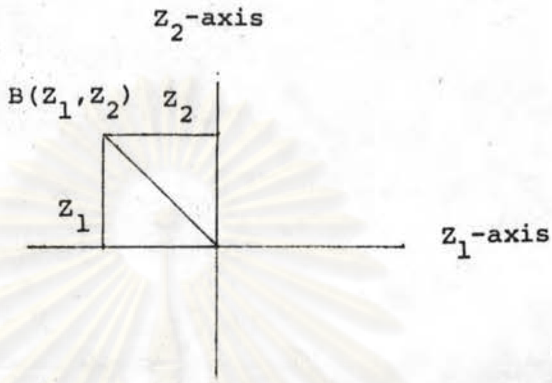
```
      SUBROUTINE RAND (IX, IY, YFL)
      IY  =  IX * 65539
      IF (IY) 5, 6, 6
5     IY  =  IY + 2147483647 + 1
6     YFL =  IY
      YFL =  YFL / 2147483647
      IX  =  IY
      RETURN
      END
```

3. <u>การผลิตเลขสุ่มที่มีการแจกแจงแบบปกติ</u>

การแจกแจงแบบปกติโดยใช้เทคนิคแบบการแปลงโดยตรงจาก

$$\phi(X) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{X^2}{2}} dX \quad ; \quad -\infty < X < \infty$$

Box และ Muller (ค.ศ. 1958) สร้างเลขสุ่มที่มีการแจกแจงแบบปกติมาตรฐาน ที่มีค่าเฉลี่ยเป็น 0 และค่าความแปรปรวนเป็น 1 พร้อม ๆ กัน 2 ค่า ดังนี้



$$Z_1 = B\cos\Theta$$

$$Z_2 = B\sin\Theta$$

$B^2 = Z_1^2 + Z_2^2$ มีการแจกแจงไคสแควร์ (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 mutually independent

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

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

ฟังก์ชันสำหรับจำลองแบบประชากรที่มีการแจกแจงเป็นแบบปกติ มีค่าเฉลี่ย μ และ ค่าเบี่ยงเบนมาตรฐาน σ จะเรียกใช้ FUNCTION NORMAL(DMEAN, SIGMA) ซึ่งจะได้จากค่า NORMAL = Z1 * SIGMA + DMEAN หรือ NORMAL = Z2 * 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 และส่วนเบี่ยงเบนมาตรฐานตามที่กำหนด จะใช้วิธีที่ Ramsay (ค.ศ. 1977) เสนอไว้ โดยพิจารณาการแจกแจงซึ่งแปลงมาจากการแจกแจงแบบปกติ ที่มีฟังก์ชันการแจกแจงอยู่ในรูปของ

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

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

μ และ σ^2 เป็นค่ากำหนดค่าเฉลี่ยและความแปรปรวน

p และ c เป็นค่ากำหนดเปอร์เซนต์การปลอมปน และสเกลแฟคเตอร์

ดังนั้น คำสั่งในการสร้างตัวแปรให้มีการแจกแจงแบบปกติปลอมปน คือ

```
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 ซึ่งเป็นตัวแปรสุ่มที่มีการแจกแจงแบบไคสแควร์ ที่มีระดับความเป็นอิสระ เป็น k โดยที่ตัวแปร X และ Y เป็นอิสระต่อกัน

ดังนั้น คำสั่งในการสร้างตัวแปรให้มีการแจกแจงแบบที คือ

```
      FUNCTION TDIST(NDF,DMEAN,SIGMA)
      REAL NORMAL
C     T-DISTRIBUTION : T = X(NORMAL)/CHISQ ; CHISQ (NDF=DF.)
      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. การสร้างการแจกแจงแบบลอกนอร์มอล

การแจกแจงแบบลอกนอร์มอลมีฟังก์ชันความหนาแน่นอยู่ในรูปของ

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

เมื่อ μ และ σ^2 เป็นค่าเฉลี่ยและความแปรปรวนของ Y ซึ่ง $Y = \log_e X$ และ Y มีการแจกแจงแบบปกติ โดยมี $\exp(\sigma^2)$ เป็น scale parameter และ μ เป็น shape parameter

ค่าคาดหวัง ค่าความแปรปรวนและค่าสัมประสิทธิ์ความแปรปรวนของการแจกแจงแบบลอกนอร์มอล คือ

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

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

$$CV(X) = \sqrt{\exp\{\sigma^2\} - 1}$$

ดังนั้น คำสั่งในการสร้างตัวแปรให้มีการแจกแจงแบบลอกนอร์มอล คือ

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

7. การสร้างการแจกแจงแบบแกมมา

การแจกแจงแบบแกมมา มีฟังก์ชันความหนาแน่นอยู่ในรูปของ

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

โดย β เป็น scale parameter α เป็น shape parameter

การสร้างตัวแปรสุ่มให้มีการแจกแจงแบบแกมมาใช้คุณสมบัติ reproductive property คือ เมื่อ X_i โดยที่ $i = 1,..,n$ เป็นตัวแปรสุ่มจากการแจกแจงแบบ GAMMA(G) แล้ว $X = \sum_{i=1}^{n} X_i$ มีรูปแบบเป็น G(α β) ซึ่ง $\alpha = \sum_{i=1}^{n} \alpha_i$ ดังนั้นเมื่อ α เป็นตัวเลขค่าเต็มหรือ $\alpha = m$ ตัวแปรจากการแจกแจงแบบแกมมา G(m,β) สามารถผลิตได้โดยการรวมตัวแปรสุ่มแบบเอ็กซ์โปเนนเชียลที่เป็นอิสระกัน m ตัว ดังนี้

$$X = \beta \sum_{i=1}^{n} (-\log_e U_i)$$

$$= -\beta \log_e \prod_{i=1}^{n} U_i$$

เมื่อ U_i เป็นตัวแปรสุ่มที่มาจากการแจกแจงแบบสม่ำเสมอมีพิสัยภายในช่วง 0 ถึง 1

ค่าคาดหวัง ค่าความแปรปรวนและค่าสัมประสิทธิ์ความแปรปรวนของการแจกแจงแบบแกมมาคือ

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

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

$$CV(X) = \frac{1}{\sqrt{\alpha}}$$

ดังนั้น คำสั่งในการสร้างตัวแปรให้มีการแจกแจงแบบแกมมา คือ

```
      FUNCTION GAMMA1(ALPHA1,BETA1)
      COMMON /SEED/ IX
C     GAMMA DISTRIBUTION : X = - BETA * SUM(LN(R(I)) ; I=ALPHA,...,1
C                          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>

การแจกแจงแบบไวบูลล์ มีฟังก์ชันความหนาแน่นอยู่ในรูปของ

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

โดย β เป็น scale parameter α เป็น shape parameter

ค่าคาดหวัง ค่าความแปรปรวนและค่าสัมประสิทธิ์ความแปรปรวนของการแจกแจงแบบไวบูลล์ คือ

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

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

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

การสร้างตัวแปรสุ่มให้มีการแจกแจงแบบไวบูลล์ อาศัยเทคนิคการแปลงผกผัน (inverse transformation)

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

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

ขั้นที่ 3 หาค่าของ x ในเทอมของ R ได้เป็น $X = \beta\left[-\log_e(1-R)\right]^{\frac{1}{\alpha}}$

ดังนั้น คำสั่งในการสร้างตัวแปรให้มีการแจกแจงแบบไวบูลล์ คือ

```
      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
```

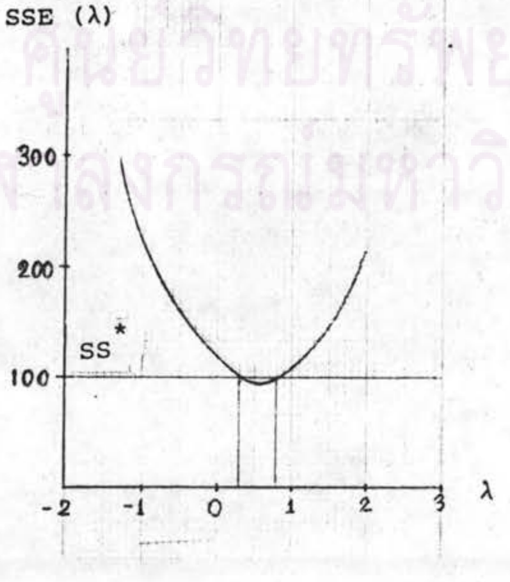
ภาคผนวก ข.

1. <u>วิธีการค้นหาค่า $\hat{\lambda}$ ซึ่งทำให้ $\sigma^2_{\varepsilon|\lambda}$ มีค่าน้อยที่สุด สำหรับการแปลงที่ใช้การยกกำลังของ Box และ Cox</u>

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

$$SS^* = SSE(\lambda)\left(1 + \frac{t^2_{\alpha/2,\nu}}{\nu}\right)$$

เมื่อ ν เป็นจำนวนของระดับความเป็นอิสระของค่าผิดพลาด



รูปที่ 1 การแสดงกราฟของผลบวกของค่าผิดพลาดกำลังสองกับ λ

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

1) กำหนดช่วงพิสัยของ λ ที่จะใช้ในการหา $\hat{\sigma}^2_{\varepsilon|\lambda}$ โดยให้

   λ_1 เป็นจุดเริ่มต้น λ_2 เป็นจุดกลาง และ λ_3 เป็นจุดสุดท้าย

   ระหว่างจุด λ_1 กับ λ_2 สามารถแบ่งเป็นจุดเล็ก ๆ ที่ยอมรับได้ห่างกัน $d = 0.1$

   จำนวนจุดที่อยู่ระหว่าง λ_1 กับ $\lambda_2 = MR = (|\lambda_1| + |\lambda_2|)/d$

   โดย MR เป็นเลขจำนวนเต็มคู่มีค่าเป็น 2^b

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

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

3) เปรียบเทียบ $\hat{\sigma}^2_{\varepsilon|\lambda_1}$ กับ $\hat{\sigma}^2_{\varepsilon|\lambda_2}$ และ $\hat{\sigma}^2_{\varepsilon|\lambda_3}$ โดยหาค่า $\hat{\sigma}^2_{\varepsilon|\lambda}$ ที่มีค่าน้อยที่สุด เมื่อค่า $\hat{\sigma}^2_{\varepsilon|\lambda}$ มีค่าน้อยที่สุดจะกำหนดให้เป็น $\hat{\lambda}$ นี้เป็น λ_2 เปลี่ยนค่า $\hat{\sigma}^2_{\varepsilon|\lambda}$ และ $\hat{\beta}_{\varepsilon|\lambda}$ ใด ๆ มาเป็น $\hat{\sigma}^2_{\varepsilon|\lambda_2}$ และ $\hat{\beta}_{\varepsilon|\lambda_2}$

4) เปรียบเทียบค่า MR และ 1

   ถ้า MR = 1 จะยอมรับค่า λ_2 เป็นค่าที่ให้ $\hat{\sigma}^2_{\varepsilon|\lambda}$ น้อยที่สุด และจะได้ค่า $\hat{\beta}_{\varepsilon|\lambda_2}$ ด้วย ถือว่าจบขบวนการ ถ้า MR $\neq$ 1 ให้ทำข้อ 5)

5) คำนวณค่า MR (ใหม่) = MR (เก่า)/2

   คำนวณ $\lambda_1 = \lambda_2 - 0.1 * MR$ (ใหม่)

   คำนวณ $\lambda_3 = \lambda_2 + 0.1 * MR$ (ใหม่)

6) กลับไป 2)

สำหรับโปรแกรมนี้ λ_1 และ λ_3 สามารถเคลื่อนย้ายออกไปจากจุดที่กำหนดตั้งแต่เริ่มต้นได้ ถ้าหากว่าค่า $\hat{\lambda}$ ที่ต้องการไม่ได้อยู่ในพิสัยตั้งแต่เริ่มต้น และจำนวนครั้งของการค้นหาจะขึ้นอยู่กับจำนวนของ MR ครั้งแรก ซึ่งถ้า MR = 2^b จำนวนครั้งของการค้นหาเท่ากับ $2b + 3$ ครั้ง

2. การทดสอบว่าข้อมูลมีการแจกแจงแบบปกติของ Shapiro และ Wilk

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

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

โดย k เป็นค่าของเลขนับที่ใหญ่ที่สุดของ $\frac{n}{2}$

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

ก. ถ้า n เป็นเลขคู่ โดยที่ n = 2k คำนวณ

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

ถ้า n เป็นเลขคี่ โดยที่ n = 2k + 1 คำนวณค่า b ดังนี้ เมื่อ $a_{k+1} = 0$

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

เมื่อ ค่า y_{k+1} ค่ามัธยฐาน(median) ไม่ถูกนำไปคำนวณค่า b ด้วย

ข. คำนวณค่า $$S^2 = \sum_{i=1}^{n} (Y_i - \bar{Y})^2$$

ค. คำนวณ W ในรูปของ $$W = \frac{b^2}{S^2}$$

ง. เปรียบเทียบค่า W (คำนวณ) กับ W (ตาราง) ถ้า W (คำนวณ) มีค่าน้อยอย่างมีนัยสำคัญ แสดงว่าข้อมูลไม่ได้มาจากการแจกแจงแบบปกติ

Shapiro, Wilk และ Chen (1964 : 1343-1372) ได้ศึกษาเปรียบเทียบตัวสถิติทดสอบ W กับ$\sqrt{b_1}$ (โมเมนต์ที่ 3), b_2 , KS(Kolmogorov - Smirnov), CM(Cramer-Von Mises), WCM (Weighted CM), D(modified KS), CS(Chi-squared) และ U(Studentized rang) ในการแจกแจง 45 แบบต่างๆ กันและขนาดตัวอย่างต่างๆ กัน วิธีการทดสอบของ Shapiro และ Wilk เป็นตัววัดความเป็นปกติที่เหนือกว่าวิธีการอื่น ๆ

สำหรับขนาดตัวอย่าง n>50 , Shapiro และ Francia (1972) ได้กล่าวถึงค่าสถิติ W ที่ปรับปรุงขึ้น ในรูปประมาณเป็นการทดสอบ W'

โดย $$W' = \frac{\left(\sum_{i=1}^{n} b_i Y_i\right)^2}{\sum_{i=1}^{n} (Y_i - \bar{Y})^2}$$

เมื่อ $$b' = (b_1, b_2 \ldots, b_n) = \frac{m'}{(m'm)^{\frac{1}{2}}}$$

มี $m' = (m_1, \ldots, m_n)$ แทนเวกเตอร์ของค่าปกติมาตรฐานของค่าสถิติอันดับ (Order Statistics)

ตารางที่ 1 แสดงค่าสัมประสิทธิ์ (a_{n-i+1}) สำหรับการทดสอบ W ของการแจกแจงปกติ สำหรับ n เท่ากับ 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 แสดงค่าสัมประสิทธิ์ (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	.868	.884	.905	.920	.959	.979	.983	.986	.968
50	.930	.938	.947	.955	.974	.985	.988	.990	.991

ตารางที่ 4 แสดงค่า b โดย $b = \dfrac{m'}{(m'm)^{\frac{1}{2}}}$ ใช้กับขนาดของ n = 100

k	b	k	b	k	b	k	b	k	b
1	0.2543	11	0.1267	21	0.0833	31	0.0516	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	0.0731	34	0.0431	44	0.0161
5	0.1711	15	0.1070	25	0.0698	35	0.0404	45	0.0166
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 หาได้จาก หนังสือวารสารของ H. LEON HARTER ชื่อ Expected Value of normal order statistic วารสาร Biometrika (1961), 48, หน้า p.151.

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

ตารางที่ 5 แสดง Percentage Point ของการทดสอบ w สำหรับ n = 100 และ 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 = $\frac{m'}{(m'\ m)^{1/2}}$ 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	0.0352	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

```
C ********************************************************************
C ***                                                             ***
C ***           E S T I M A T I O N   O F                         ***
C ***    R E G R E S S I O N   C O E F F I C I E N T              ***
C ***     B Y   P R A N E E   R A T T A N U N G                   ***
C ***                                                             ***
C ********************************************************************
C
C       METHOD     1)  ORDINARY LEAST  SQUARE METHOD (OLS)
C                  2)  M-ESTIMATOR WITH RAMSAY CRITERIA (MEST)
C       MAIN PROGRAM 1
C            WHEN RESIDUALS HAVE LONGER TAIL DISTRIBUTION THAN NORMAL
C              USE METHOD     1)  CALL OLS
C                             2)  CALL MEST
C       MAIN PROGRAM 2
C            WHEN RESIDUALS HAVE SKEWNESS DISTRIBUTION
C              USE METHOD     1)  CALL BOXCOX
C                               (BOX & COX POWER TRANSFORMATION & OLS)
C                             2)  CALL BOXC
C                               (BOX & COX POWER TRANSFORMATION & MEST)
C       DISTRIBUTION :II
C                  II = 1  - CONTAMINATED NORMAL
C                     = 2  - T
C                     = 3  - LOGNORMAL
C                     = 4  - GAMMA
C                     = 5  - WEIBULL
C ____________________________________________________________________
C WHEN USE MAIN PROGRAM 1 YOU MUST INSERT C(COMMENT) IN COL.1 INTO MAIN
C PROGRAM 2 , WHEN USE MAIN PROGRAM 2 YOU MUST DO INVERSELY
C AND USE SUBROUTINE CONSISTENCY WITH MAIN PROGRAM
C--------------------------------------------------------------------
      MAIN PROGRAM 1

      REAL MSE1,MSE2
      DOUBLE PRECISION A,S,B,X,B1,BM
      COMMON /REGRS/ A(11,11),S(12,12)
     *       /COEFF/ B(11)
     *       /CONTA/ P,IC /INTERV/XBAR1(11),SG(11)
     *       /DATAXY/ X(12,150) /SEED/ IX /SELECT/ KK
     *       /DIST/ II
     *       /VARIAB/ N,M,M2
      DIMENSION B1(11),BM(11)
      DATA    IK1,IK2,SMO1,SMO2,SMM1,SMM2,S1,SS1,S2,SS2
     *        S3,SS3,S4,SS4/14*0/
      CALL INIT
      DO 100 I = 1,200
      CALL DATA
      WRITE(4,10) (B(J),J=1,M)
 10   FORMAT(' VECTOR B              ',/,11(1X,F11.3))
      CALL OLS(B1)
      WRITE(4,25) (B1(J),J=1,M)
 25   FORMAT(' VECTOR B - USE OLS',/,11(1X,F11.3))
      MSE1 = 0
      DO 30 J = 1,M
 30   MSE1 = MSE1 + (B1(J)-B(J))*(B1(J)-B(J))
      MSE1 = MSE1 / M
```

```
      SMO1 = SMO1 + MSE1
      SMO2 = SMO2 + MSE1 * MSE1
50    CALL MEST(B1,BM)
      WRITE(4,55) (BM(J),J=1,M)
55    FORMAT(' VECTOR B - USE M-EST',/,11(1X,F11.3))
      MSE2 = 0
      DO 90 J  = 1,M
90    MSE2 = MSE2 + (BM(J)-B(J))*(BM(J)-B(J))
      MSE2 = MSE2 / M
      SMM1 = SMM1 + MSE2
      SMM2 = SMM2 + MSE2 * MSE2
      IF(  MSE1 - MSE2 ) 100,100,110
100   RMSE1 = MSE2/MSE1
      DIFF1 = MSE1/MSE2  - 1.0
      IK1    = IK1 + 1
      S1     = S1  + RMSE1
      SS1    = SS1 + RMSE1 * RMSE1
      S2     = S2  + DIFF1
      SS2    = SS2 + DIFF1 * DIFF1
      GOTO 115
110   RMSE2 = MSE1/MSE2
      DIFF2 = MSE2/MSE1  - 1.0
      IK2    = IK2 + 1
      S3     = S3  + RMSE2
      SS3    = SS3 + RMSE2 * RMSE2
      S4     = S4  + DIFF2
      SS4    = SS4 + DIFF2 * DIFF2
115   WRITE(4,116) MSE1,MSE2,RMSE1,RMSE2,DIFF1,DIFF2
116   FORMAT(' MSE1,MSE2,RMSE1,RMSE2,DIFF1,DIFF2=',6F10.3)
100   CONTINUE
      XX1  = S1 / FLOAT(IK1)
      XX2  = S2 / FLOAT(IK1)
      XX3  = S3 / FLOAT(IK2)
      XX4  = S4 / FLOAT(IK2)
      SD1  = SQRT((SS1 - (S1*XX1))/FLOAT(IK1-1))
      SD2  = SQRT((SS2 - (S2*XX2))/FLOAT(IK1-1))
      SD3  = SQRT((SS3 - (S3*XX3))/FLOAT(IK2-1))
      SD4  = SQRT((SS4 - (S4*XX4))/FLOAT(IK2-1))
      WRITE(4,150) S1,XX1,SS1,SD1,IK1
      WRITE(4,150) S2,XX2,SS2,SD2,IK1
      WRITE(4,150) S3,XX3,SS3,SD3,IK2
      WRITE(4,150) S4,XX4,SS4,SD4,IK2
150   FORMAT('  SUM OF',F15.2,/,' AVERAGE =',F10.2,/,
     *'  SUM SQUARE OF=',F20.4,/,'  SD. = ',F10.3,/,'  IK=',I3,/,
     *'*****************************************************')
      XBD = ( SMO1 - SMM1 ) /SQRT(200.0)
      XBD1= SMO1/200.0
      XBD2= SMM1/200.0
      SD1 = SMO2-(SMO1*SMO1)/200.0
      SD2 = SMM2-(SMM1*SMM1)/200.0
      SD11= (SMO2-(SMO1*SMO1)/200.0)/199.0
      SD22= (SMM2-(SMM1*SMM1)/200.0)/199.0
      ZD  = XBD / SQRT((SD1+SD2)/199.0)
      WRITE(6,160) IX,SMO1,SMO2,SMM1,SMM2,ZD,XBD1,XBD2,SD11,SD22
160   FORMAT(' IX=',I16,9F12.4)
      STOP
      END
```

```
      MAIN PROGRAM 2

      REAL MSE1,MSE2
      DOUBLE PRECISION Y1,FL,A,S,B,X,W,A2,SW,Y,T,B1,BM,SLG,BB
      COMMON /REGRS/ A(11,11),S(12,12)
     *       /COEFF/ B(11)
     *       /CONTA/ P,IC /INTERV/XBAR1(11),SG(11) /SKEWED/ ALPHA,BETA
     *       /DATAXY/ X(12,150) /SEED/ IX /SELECT/ KK /DATAY/ Y(150)
     *       /TRANSF/ Y1(150),FL,ST,FIN /WEIGHT/ W(150) /DIST/ II
     *       /VARIAB/ N,M,M2 /TABLE/ A2(150),SW(9) /STORE/ BB(3,11)
      DIMENSION T(150),B1(11),BM(11)
      DATA    III,IK1,IK2,SMO1,SMO2,SMM1,SMM2,S1,SS1,S2,SS2,
     *        S3,SS3,S4,SS4,FL1,FLL1,F3,FL3/19*0/
      CALL INIT
      DO 100 I = 1,200
      CALL DATA
      WRITE(6,10) (B(J),J=1,M)
10    FORMAT(' VECTOR B           ',/,11(1X,F11.3))
      CALL BOXCOX(B1,FL1)
      WRITE(6,25) FL1,(B1(J),J=1,M)
25    FORMAT('FL =',F11.3,/,' VECTOR B - USE OLS',/,11(1X,F11.3))
      MSE1 = 0
      DO 30 J = 1,M
30    MSE1 = MSE1 + (B1(J)-B(J))*(B1(J)-B(J))
      MSE1 = MSE1 / M
      SMO1 = SMO1 + MSE1
      SMO2 = SMO2 + MSE1 * MSE1
      F1   = F1 + FLL1
      FLL1 = FLL1 + FLL1*FL1
      FL   = FL1
      CALL BCOX
      DO 40 J = 1,N
40    T(J) = X(M2,J)
      CALL SHAPWK(N,T,ITEST)
      IF(ITEST.EQ.2) GOTO 50
      III = III + 1
50    CALL BOXC(BM,FL1)
      WRITE(6,55) FL1,(BM(J),J=1,M)
55    FORMAT('FL =',F11.3,/,' VECTOR B - USE M-EST',/,11(1X,F11.3))
      MSE2 = 0
      DO 90 J  = 1,M
90    MSE2 = MSE2 + (BM(J)-B(J))*(BM(J)-B(J))
      MSE2 = MSE2 / M
      SMM1 = SMM1 + MSE2
      SMM2 = SMM2 + MSE2 * MSE2
      F3   = F3 + FL1
      FL3  = FL3 + FL1*FL1
      IF(  MSE1 - MSE2 ) 100,100,110
100   RMSE1 = MSE2/MSE1
      DIFF1 = MSE1/MSE2  - 1.0
      IK1   = IK1 + 1
      S1    = S1  + RMSE1
      SS1   = SS1 + RMSE1 * RMSE1
      S2    = S2  + DIFF1
      SS2   = SS2 + DIFF1 * DIFF1
      GOTO 115
```

```
110  RMSE2 = MSE1/MSE2
     DIFF2 = MSE2/MSE1  - 1.0
     IK2   = IK2 + 1
     S3    = S3  + RMSE2
     SS3   = SS3 + RMSE3 * RMSE2
     S4    = S4  + DIFF2
     SS4   = SS4 + DIFF2 * DIFF2
115  WRITE(6,116) MSE1,MSE2,RMSE1,RMSE2,DIFF1,DIFF2
116  FORMAT(' MSE1,MSE2,RMSE1,RMSE2,DIFF1,DIFF2=',6F10.3)
100  CONTINUE
     XX1  = S1 / FLOAT(IK1)
     XX2  = S2 / FLOAT(IK1)
     XX3  = S3 / FLOAT(IK2)
     XX4  = S4 / FLOAT(IK2)
     SD1  = SQRT((SS1 - (S1*XX1))/FLOAT(IK1-1))
     SD2  = SQRT((SS2 - (S2*XX2))/FLOAT(IK1-1))
     SD3  = SQRT((SS3 - (S3*XX3))/FLOAT(IK2-1))
     SD4  = SQRT((SS4 - (S4*XX4))/FLOAT(IK2-1))
     WRITE(6,150) S1,XX1,SS1,SD1,IK1
     WRITE(6,150) S2,XX2,SS2,SD2,IK1
     WRITE(6,150) S3,XX3,SS3,SD3,IK2
     WRITE(6,150) S4,XX4,SS4,SD4,IK2
150  FORMAT('  SUM OF',F15.2,/,' AVERAGE =',F10.2,/,
    *'  SUM SQUARE OF=',F20.4,/,'  SD. = ',F10.3,/,'  IK=',I3,/,
    *'**************************************************************')
     XBD = ( SMO1 - SMM1 ) /SQRT(200.0)
     XBD1= SMO1/200.0
     XBD2= SMM1/200.0
     SD1 = SMO2-(SMO1*SMO1)/200.0
     SD2 = SMM2-(SMM1*SMM1)/200.0
     SD11= (SMO2-(SMO1*SMO1)/200.0)/199.0
     SD22= (SMM2-(SMM1*SMM1)/200.0)/199.0
     ZD  = XBD / SQRT((SD1+SD2)/199.0)
     WRITE(6,160) IX,SMO1,SMO2,SMM1,SMM2,ZD,XBD1,XBD2,SD11,SD22
160  FORMAT(' IX=',I16,9F12.4)
     WRITE(6,170) III,FL1,FLL1,F3,FL3
170  FORMAT(' NO. NORMALITIES',I3,/,
    *' SUM OF FL,SUM SQ OF FL - USE OLS ',2(5X,FL15.4),/,
    *' SUM OF FL,SUM SQ OF FL - USE M-EST ',2(5X,F15.4))
     STOP
     END
C------------------------------------------------------------------------
     SUBPROGRAM FOR MAIN PROGRAM 1 & MAIN PROGRAM 2
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C    SOME SYMBOL IN SUBROUTINE INIT
C    N       NUMBER OF OBSERVATION
C    M       NUMBER OF INDEPENDENT VARIABLES(INCL. INTERCEPT)
C    P       PERCENT CONTAMINATED/100
C    C       SCALE FACTER
C    ALPHA   SHAPE PARAMETER
C    BETA    SCALE PARAMETER
C........................................................................
```

```
      SUBROUTINE INIT
      REAL NORMAL
      DOUBLE PRECISION B,X,A2,SW,A,S
      COMMON /SEED/ IX /DATAXY/ X(12,150) /COEFF/ B(11)
     *       /CONTA/ P,IC /INTERV/ XBAR1(11),SG(11) /SKEWED/ ALPHA,BETA
     *       /SELECT/ KK /VARIAB/ N,M,M2 /TABLE/ A2(150),SW(9)
     *       /REGRS/ A(11,11),S(12,12)
      READ(5,5) N,M
 5    FORMAT(2I3)
      WRITE(6,10) N,M,P,IC
 10   FORMAT('N , M, P,  IC',/,2I3,F5.2,I3)
      DO 15 I = 1,M
 15   READ(5,20) XBAR1(I),SG(I)
 20   FORMAT(2F7.3)
      M2 = M + 1
      DO 25 I = 1,M
 25   READ(5,30) B(I)
 30   FORAT(F10.4)
      WRITE(6,35) ALPHA,BETA
 35   FORMAT(' ALPHA=',F5.2,' BETA=',F5.2)
      KK  = 0
      K   = INIT(FLOAT(N/2))
      DO 50 I = 1,K
 50   READ(5,60) A2(N-I+1)
 60   FORAMT(F7.4)
      DO 70 I = 1,4
 70   READ(5,80) SW(I)
 80   FORAMT(F6.3)
C  GENERATE INDEPENDENT VARIABLES
      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  CAL X'X & X'Y FOR SUBROUTINE OLS
      DO 120 I = 1,M
      DO 120 K = I,M
      SIK = 0.0
      DO 110 J = 1,N
 110  SIK = SIK + X(I,J)*X(K,L)
      S(I,K) = SIK
 120  S(K,I) = SIK
C CAL INVERSE METRIX OF X'X
      DO 140 I = 1,M
      DO 140 J = I,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 ON DIAGONAL CANNOT USE SUBRORTINE INVS')
      STOP
 145  CONTINUE
      CALL INVS(M,A)
      RETURN
      END
```

```
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      SUBROUTINE DATA
      REAL NORMAL
      DOUBLE PRECISION B,X,Y,A2,SW,E,T
      COMMON /COEFF/ B(11) /SEED/ IX /SELECT/ KK /DIST/ II
     *       /CONTA/ P,IC /INTERV/XBAR1(11),SG(11) /SKEWD/ ALPHA,BETA
     *       /DATAXY/ X(12,150) /DATAY/ Y(150) /VARIA/ N,M,M2
     *       /TABLE/ A2(150),SW(9)
      DIMENSION E(150),T(150)
C         DISTRIBUTION :II
C                   II = 1  - CONTAMINATED NORMAL
C                      = 2  - T
C                      = 3  - LOGNORMAL
C                      = 4  - GAMMA
C                      = 5  - WEIBULL
C....................................................................
      GOTO(10,20,30,40,50),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)
      SIGMA = SG(M)
      NDF   = 4
      DO 25 J = 1,N
      E(J)  = TDIST(NDF,DMEAN,SIGMA)
 25   CONTINUE
      GOTO 60
 30   SIGMA = SG(M)
      DO 35 J = 1,N
      DMEAN1  = X(1,J)
      X(M2,J) = EXP(NORMAL(DMEAN1,SIGMA))
 35   CONTINUE
      GOTO 60
 40   DO 45 J = 1,N
      X(M2,J) = GAMMA1(ALPHA,BETA)
 45   CONTINUE
      GOTO 60
 50   DO 55 J = 1,N
      X(M2,J) = WEIBUL(ALPHA,BETA)
 55   CONTINUE
 60   GOTO(100,100,110,110,110),II
 100  DO 105 J = 1,N
      DO 105 I = 1,M
 105  X(M2,J)  = X(M2,J) + X(I,J) * B(I)
      X(M2,J)  = X(M2,J) + E(J)
      GOTO 120
 110  DO 115 J = 1,N
 115  Y(J)   = X(M2,J)
```

```
 120  WRITE(6,125)
 125  FORMAT(' MATRIX X & VECTOR Y')
      DO 130 J = 1,N
 130  WRITE(6,135) (X(I,J),I=1,M2)
 135  FORMAT(12(1X,F10.2))
      RETURN
      END
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:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      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
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      FUNCTION TDIST(NDF,DMEAN,SIGMA)
      REAL NORMAL
C     T-DISTRIBUTION : T = X(NORMAL)/CHISQ ; CHISQ (NDF=DF.)
      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:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      FUNCTION GAMMA1(ALPHA1,BETA1)
      COMMON /SEED/ IX
C     GAMMA DISTRIBUTION : X = - BETA * SUM(LN(R(I)) ; I=ALPHA,...,1
C                          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
C
 10   GAMMA1 = BETA1 * U
      RETURN
      END
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      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
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      SUBROUTINE OLS(B)
      DOUBLE PRECISION A,S,X,B
      COMMON /REGRS/ A(11,11),S(12,12)  /VARIAB/ N,M,M2
     *       /DATAXY/ X(12,150)
      DIMENSION B(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,I)  = SIK
      DO 50 I = 1,M
      B(I)     = 0.0
      DO 50 J = 1,M
 50   B(I)     = B(I) + A(J,I)*S(M2,J)
      RETURN
      END
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      SUBROUTINE OLS1(B)
      DOUBLE PRECISION A1,S1,B,X,W
      COMMON /REGRS1/ A1(11,11),S1(12,12)  /VARIAB/ N,M,M2
     *       /DATAXY/ X(12,150) /WEIGHT/ W(150)
      DIMENSION B(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(I,K)  = SIK
 20   S1(K,I)  = SIK
      DO 40 I = 1,M
      DO 40 J = 1,M
 40   A1(I,J)  = 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 USE SUBROUTINE INVS')
      STOP
 45   CONTINUE
      CALL INVS(M,A1)
      DO 50 I = 1,M
      B(I)     = 0.0
      DO 50 J = 1,M
```

```
 50   B(I)    = B(I) + A1(J,I)*S1(M2,J)
      RETURN
      END
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      SUBROUTINE YRESID(X,N,M,YHAT,YRES,B)
      DOUBLE PRECISION X,YHAT,YRES,B
      DIMENSION X(12,150),YHAT(150),YRES(150),B(11)
      M2  = M + 1
      DO 10 J = 1,N
      YHAT(J) = 0.0
      DO 20 I = 1,M
 20   YHAT(J) = YHAT(J) + B(I)*X(I,J)
 10   YRES(J) = X(M2,J) - YHAT(J)
      RETURN
      END
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      SUBROUTINE INVS(M,A)
      DOUBLE PRECISION A(11,11)
C THIS SUBROUTINE INVERTS A MATRIX BY SWEEPING. THE MATRX TO BE
C INVERTED NEED NOT BE SYMMETRIC BUT IT MUST BE SQUARE ,
C NONSIGULAR AND CAN NOT HAVE ANY ZERO ON ITS MAIN DIAGONAL
      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::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      SUBROUTINE MEST(B1,B)
      DOUBLE PRECISION A1,S1,X,W,S2,Z,B,B1,YHAT,YRES,E
      COMMON /REGRS1/ A1(11,11),S1(12,12)
     *        /DATAXY/ X(12,150) /WEIGHT/ W(150)
     *        /VARIAB/ N,M,M2
      REAL MEDIAN
      DIMENSION S2(150),Z(150),B(11),B1(11)
      DIMENSION YHAT(150),YRES(150),E(150)
C   M - ESTIMATOR BY RAMSAY A = 0.3
      L = 0
      CALL YRESID(X,N,M,YHAT,YRES,B1)
      DO 10 J = 1,N
 10   E(J)    = YRES(J)
      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) 15,17,15
 15      MEDIAN = E(N2)
      GOTO 18
 17      MEDIAN = (E(N3)+E(N4))/2
 18   DO 20 I = 1,N
 20   S1(I) = DABS(E(I)-MEDIAN)
      CALL RANK(N,S1)
      IF(N - N1) 25,27,25
 25      SIGMA  = S1(N2)/0.6745
      GOTO 777
 27      SIGMA  = (S1(N3)+S1(N4))/(2*0.6745)
 777  CALL YRESID(X,N,M,YHAT,YRES,B1)
      DO 30 I = 1,N
      Z(I)     = DABS(YRES(I) / SIGMA)
 30   W(I)     = DEXP(-0.3 * Z(I))
      L        = L + 1
      IF(L - 20) 60,60,90
 60   CALL OLS1(B)
      DO 70 I = 1,M
      C        = DABS(B1(I) - B(I)) / DABS(B1(I))
      IF( C - 0.001) 70,70,75
 75       DO 80 J = 1,M
 80       B1(J)   = B(J)
          GOTO 777
 70   CONTINUE
 90   RETURN
      END
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      SUBROUTINE BOXCOX(B,FL1)
      DOUBLE PRECISION Y1,FL,Y,B,SLG,BB
      COMMON /DATAY/ Y(150)
     *       /TRANSF/ Y1(150),FL,ST,FIN /VARIAB/ N,M,M2 /STORE/BB(3,11)
      DIMENSION B(11)
C  BOX & COX POWER TRANSFORMATION
      DO 30 J = 1,N
      IF(Y(J)) 20,20,30
 20      WRITE(6,25)
 25      FORMAT(' Y(J) IS NEGATIVE OR XERO THEN RETURN TO MAIN PROGRAM')
         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,SSE1,1,1)
      CALL SUMSQ(FD,SSE2,2,1)
      CALL SUMSQ(FIN,SSE3,3,1)
 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        BB(2,I) = BB(1,I)
          GOTO 100
72        FM = FD
          GOTO 100
73        FM = FIN
          SSE2 = SSE3
          DO 90 I = 1,M
90        BB(2,I) = BB(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,SSE1,1,1)
      CALL SUMSQ(FIN,SSE3,3,1)
      GOTO 70
110   FL1 = FM
      DO 120 I = 1,M
120   B(I) = BB(2,I)
      FL   = FM
      RETURN
      END
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      SUBROUTINE BOXC(B,FL1)
      DOUBLE PRECISION Y1,FL,B,BB
      COMMON /REGRS/ A(11,11),S(12,12)
     *       /DATAXY/ X(12,150)  /DATAY/ Y(150)
     *       /TRANSF/ Y1(150),FL,ST,FIN /VARIAB/ N,M,M2 /STORE/BB(3,11)
      DIMENSION B(11),B1(11),Z(16)
      L = 0
      MR = 3
      ST   = FL - 0.1 * MR
      FIN  = FL + 0.1 * MR
      FD   = FL
      CALL SUMSQ(ST,SSE1,1,3)
      CALL SUMSQ(FD,SSE2,2,2)
      CALL SUMSQ(FIN,SSE3,3,3)
70    IF(SSE1 .LE. SSE2 .AND. SSE1 .LE. SSE3) GOTO 80
      IF(SSE2 .LE. SSE1 .AND. SSE2 .LE. SSE3) GOTO 90
      IF(SSE3 .LE. SSE1 .AND. SSE3 .LE. SSE2) GOTO 100
80      ST = ST - 0.1 * MR
        FD = FD - 0.1 * MR
        FIN = FIN - 0.1 * MR
        SSE3 = SSE2
        SSE2 = SSE1
        DO 85 I = 1,M
        BB(3,I) = BB(2,I)
85      BB(2,I) = BB(1,I)
        CALL SUMSQ(ST,SSE1,1,3)
        GOTO 110
90      MR = 1
        ST = FD - 0.1
        FIN = FD + 0.1
        CALL SUMSQ(ST,SSE1,1,3)
        CALL SUMSQ(FIN,SSE3,3,3)
        GOTO 110
100     ST = ST + 0.1 * MR
        FD = FD + 0.1 * MR
```

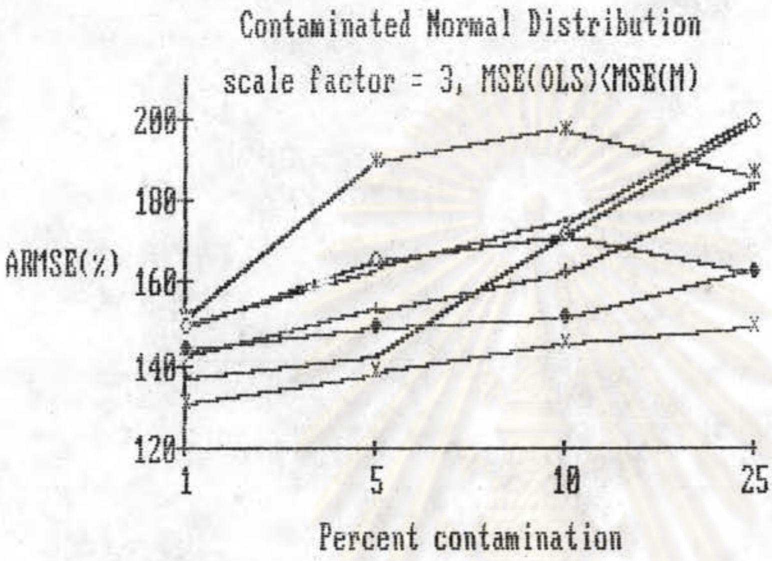
```
        FIN = FIN + 0.1 * MR
        SSE1 = SSE2
        SSE2 = SSE3
        DO 105 I = 1,M
        BB(1,I) = BB(2,I)
105     BB(2,I) = BB(3,I)
        CALL SUMSQ(FIN,SSE3,3,3)
110   L = L + 1
      IF( (L .GE. 16) .OR. (MR .EQ. 1 )) GOTO 120
      GOTO 70
120   IF(SSE1 .LE. SSE2 .AND. SSE1 .LE. SSE3) GOTO 130
      IF(SSE2 .LE. SSE1 .AND. SSE2 .LE. SSE3) GOTO 140
      IF(SSE3 .LE. SSE1 .AND. SSE3 .LE. SSE2) GOTO 150
130   FL1 = ST
      K = 1
      GOTO 160
140   FL1 = FD
      K = 2
      GOTO 160
150   FL1 = FIN
      K = 3
160   DO 170 I = 1,M
170   B(I) = BB(K,I)
      FL   = FM
      RETURN
      END
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      SUBROUTINE SUMSQ(FLX,SSE,K,L)
      DOUBLE PRECISION Y1,FL,A,S,Y,B,SLG,X,BB,FLX,W,A1,S1,B1,Z,ZZ
      COMMON /REGRS/ A(11,11),S(12,12)
     *       /REGRS1/ A1(11,11),S1(12,12) /WEIGHT/ W(150)
     *       /DATAXY/ X(12,150)  /DATAY/ Y(150)
     *       /TRANSF/ Y1(150),FL,ST,FIN /VARIAB/ N,M,M2 /STORE/BB(3,11)
      DIMENSION B(11),B1(11),Z(16)
      FL = FLX
      GOTO(10,20,30),L
10    CALL BCOX
      CALL OLS(B)
      DO 15 I = 1,M
15    Z(I) = S(M2,I)
      ZZ   = S(M2,M2)
      GOTO 50
20    CALL BCOX
      DO 25 I = 1,M
25    B1(I) = BB(2,I)
      CALL MEST(B1,B)
      DO 27 I = 1,M
27    Z(I) = S1(M2,I)
      ZZ   = S1(M2,M2)
      GOTO 50
30    CALL BCOX
      CALL OLS(B1)
      CALL MEST(B1,B)
      DO 37 I = 1,M
37    Z(I) = S1(M2,I)
      ZZ   = S1(M2,M2)
50    DO 80 I = 1,M
80    BB(K,I) = B(I)
      SSR = 0.0
```

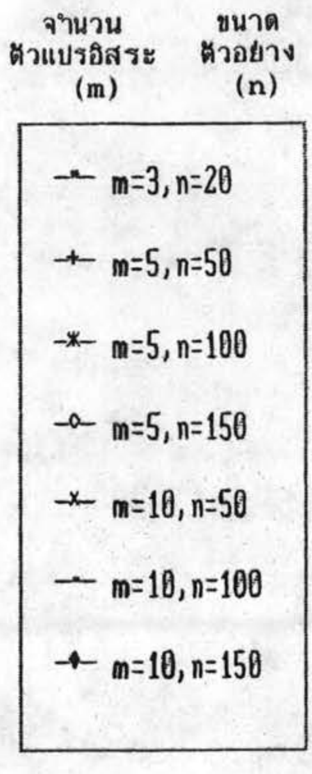
```
      DO 90 I = 1,M
 90   SSR  = SSR + B(I) * Z(I)
      SSE  = ZZ - SSR
      SSE  = SSE / FLOAT(N)
      RETURN
      END
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      SUBROUTINE BCOX
      DOUBLE PRECISION Y1,FL,X
      COMMON /DATAXY/ X(12,150)
     *       /TRANSF/ Y1(150),FL,ST,FIN  /VATIAB/ N,M,M2
      IF(DABS(FL)) 5,15,5
 5       DO 10 I = 1,N
 10      X(M2,I) = DLOG(Y1(I))
      GOTO 30
 15      DO 20 I = 1,N
 20      X(M2,I) = ((Y1(I)**FL)-1.0) /FL
 30   RETURN
      END
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::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      SUBROUTINE SHAPWK(N,Y,ITEST)
      DOUBLE PRECISION A2,SW,Y,S
      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
 20   B     = B    + A2(N-I+1) * (Y(N-I+1)-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::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

ภาคผนวก ง

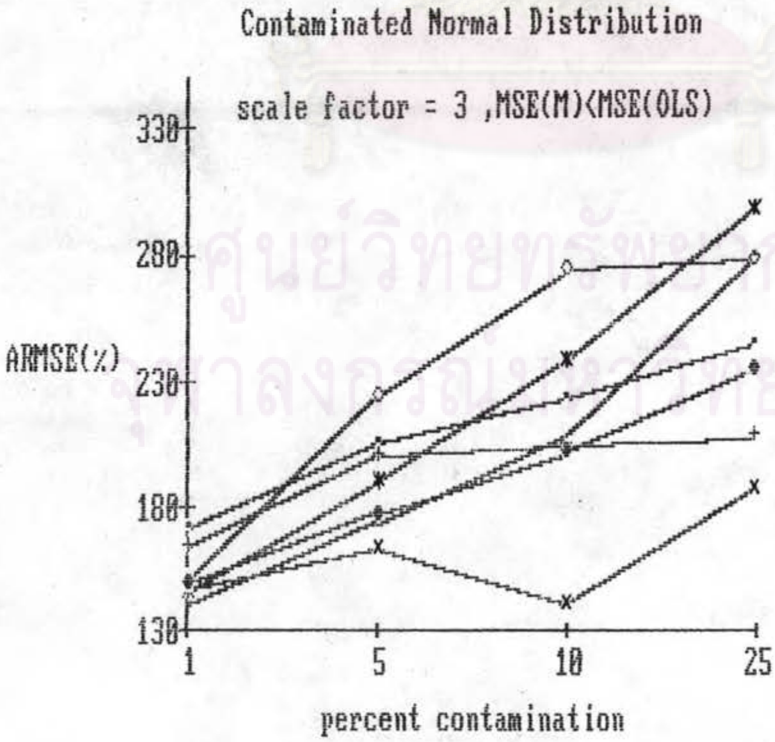
กราฟแสดงการศึกษาเปรียบเทียบวิธีกำลังสองน้อยที่สุดกับวิธี M-estimator

โดยใช้เกณฑ์การเปรียบเทียบของค่าเฉลี่ยของค่าสัมพัทธ์ของค่าเฉลี่ยความผิดพลาดกำลังสอง (ARMSE)

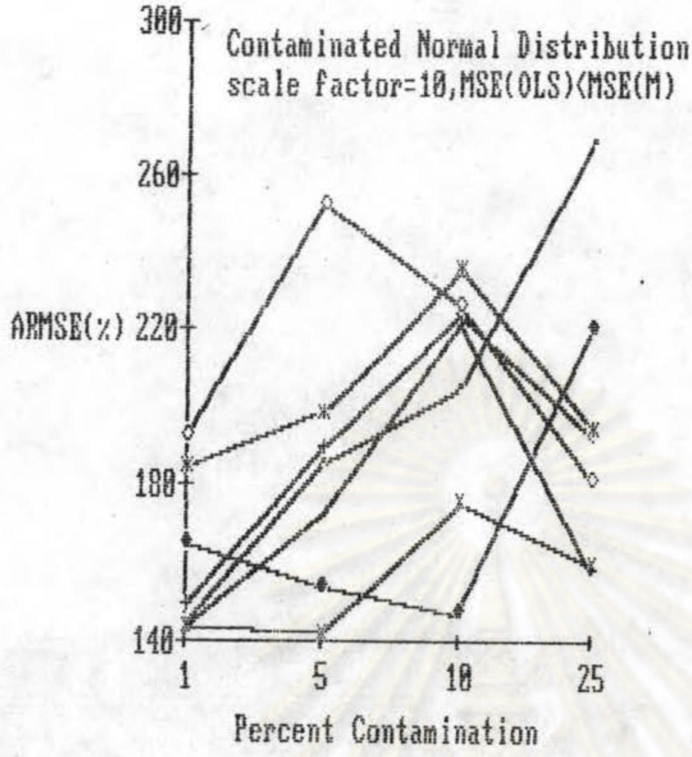


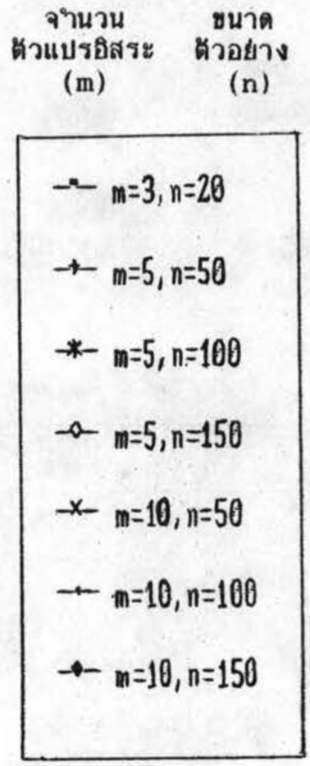


รูปที่ แสดงการแจกแจงแบบปกติปลอมปน

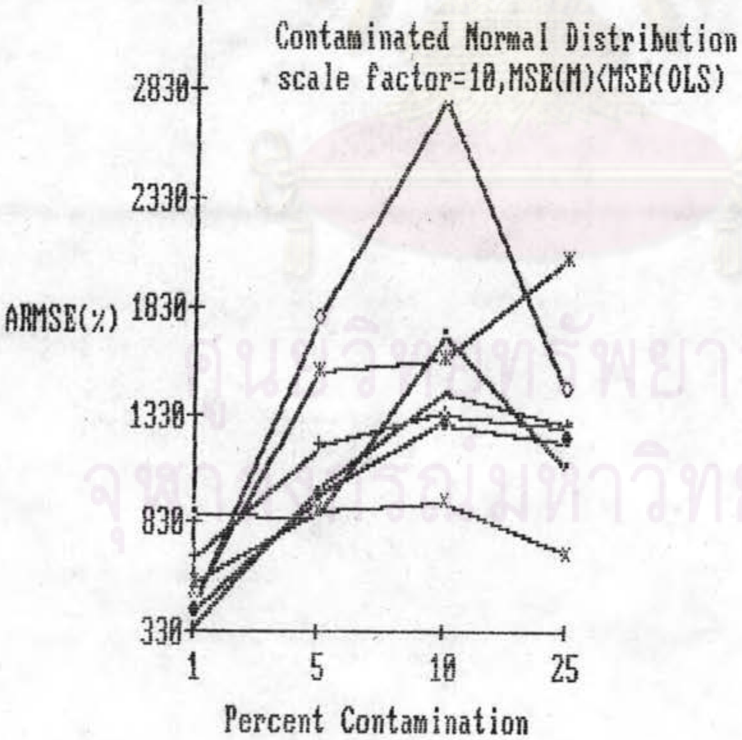


รูปที่ แสดงการแจกแจงแบบปกติปลอมปน

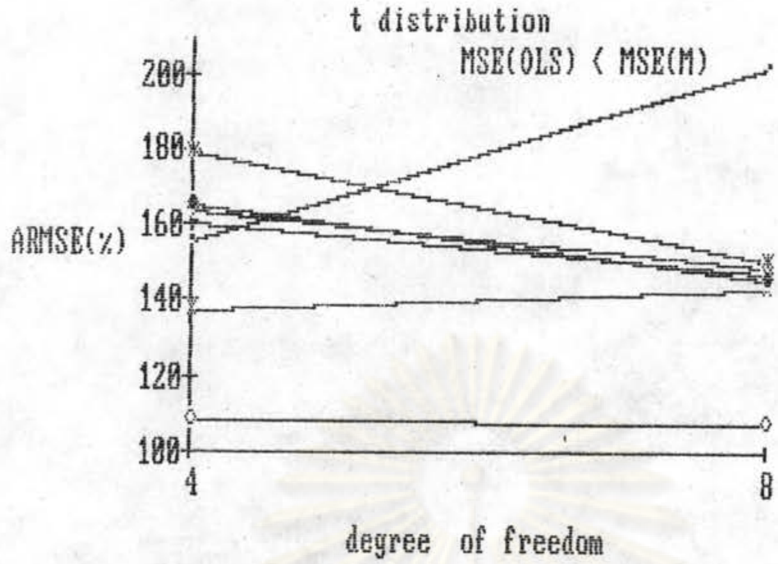




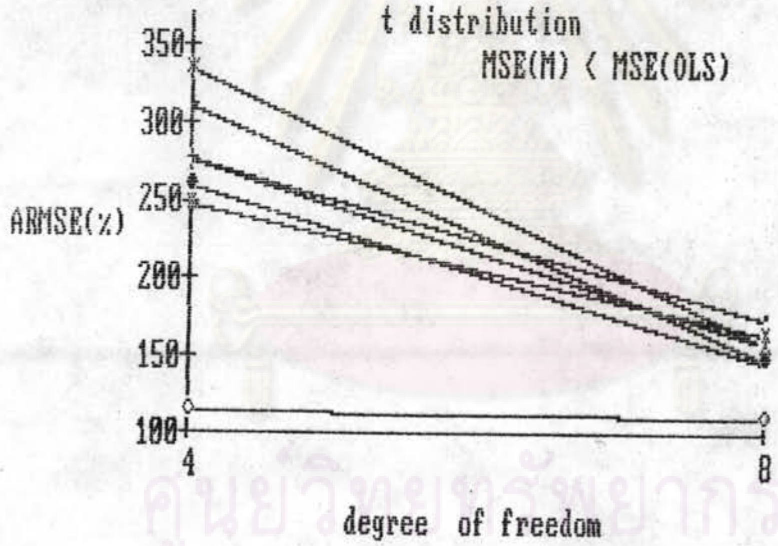
รูปที่ แสดงการแจกแจงแบบปกติปลอมปน



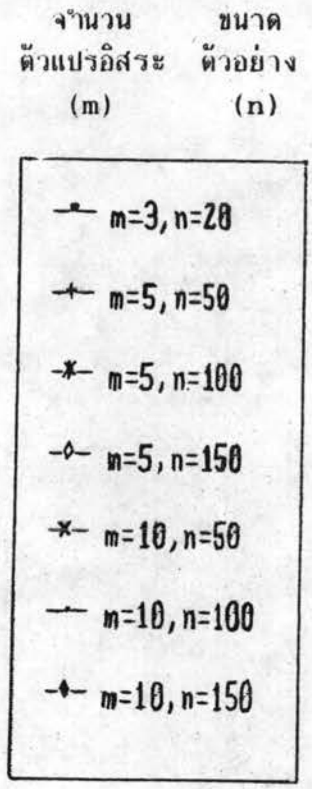
รูปที่ แสดงการแจกแจงแบบปกติปลอมปน

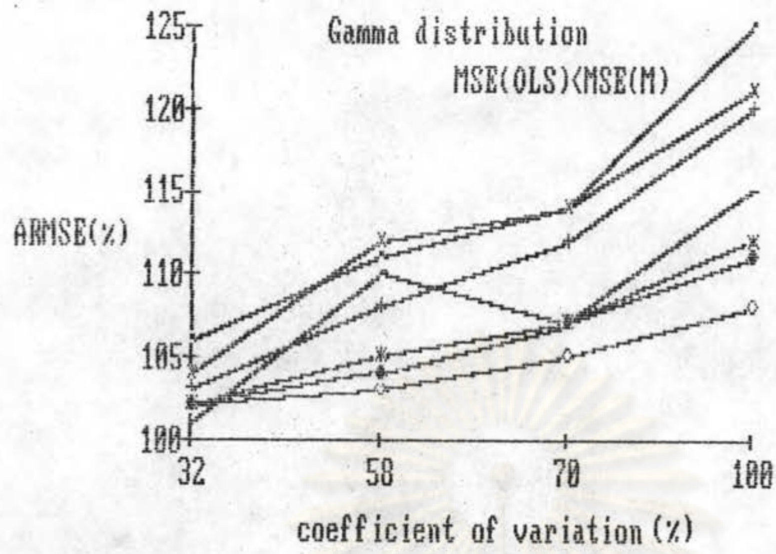


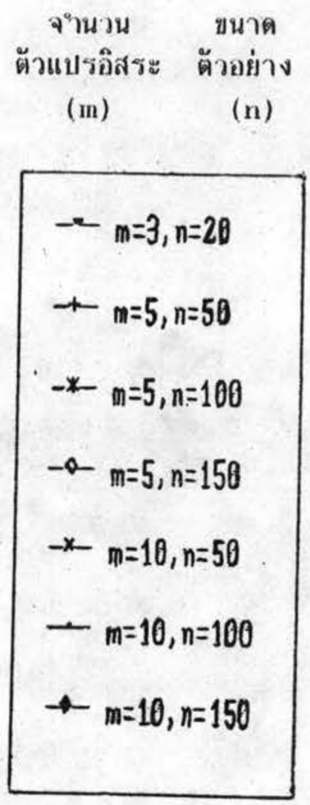
รูปที่ แสดงการแจกแจงแบบที



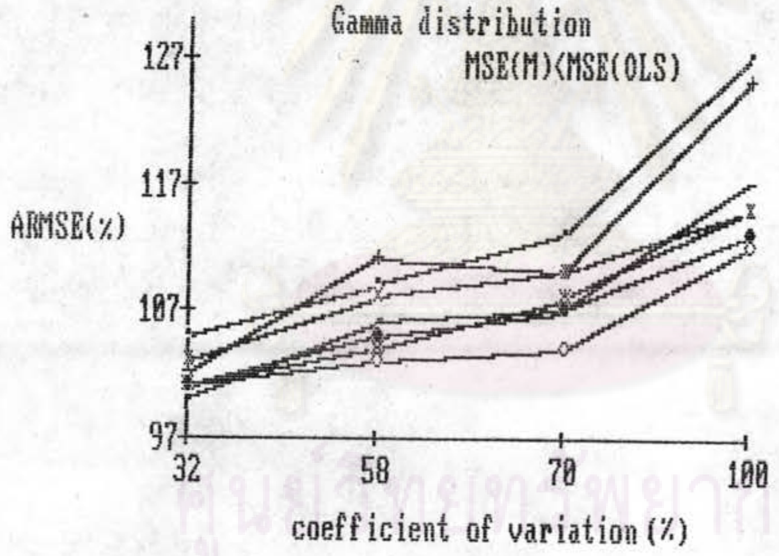
รูปที่ แสดงการแจกแจงแบบที



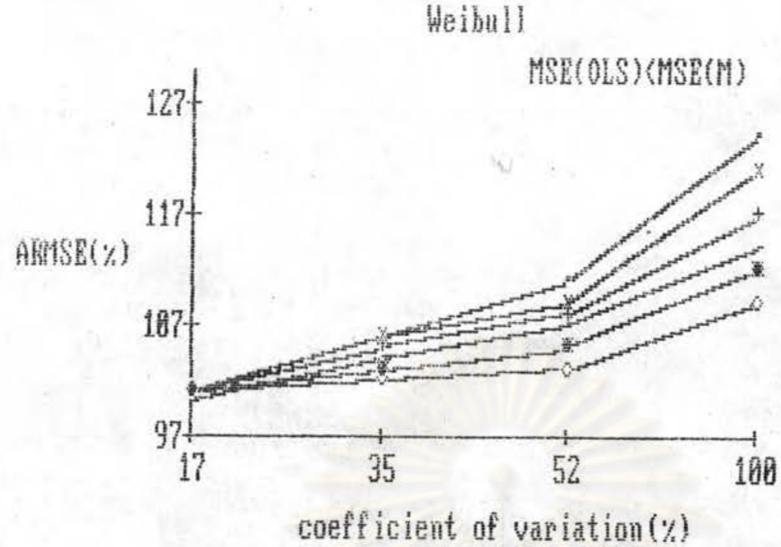


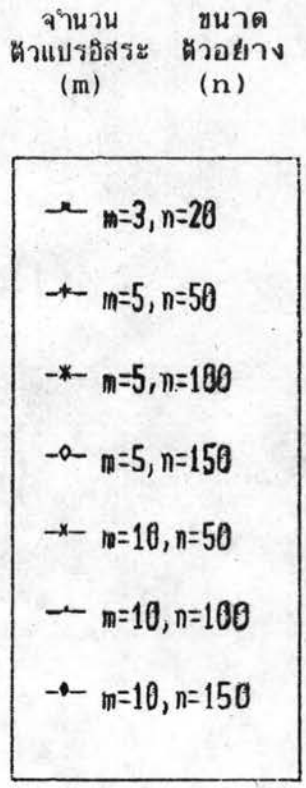


รูปที่ แสดงการแจกแจงแบบแกมมา

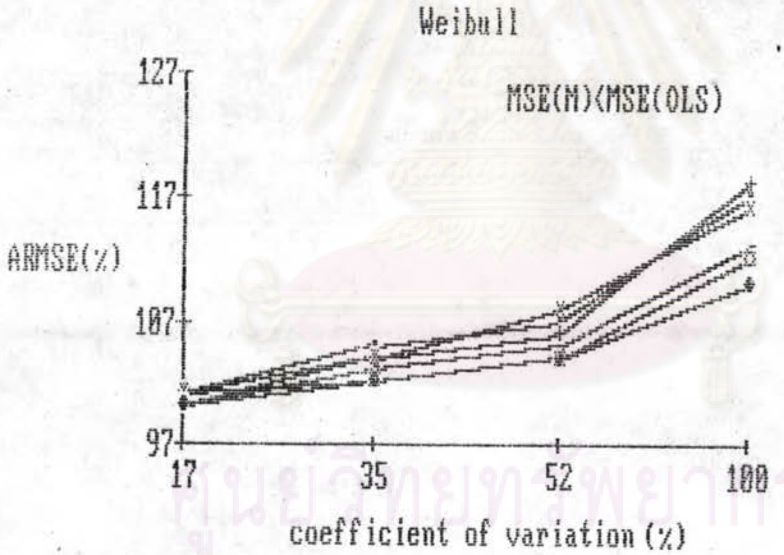


รูปที่ แสดงการแจกแจงแบบแกมมา





รูปที่ แสดงการแจกแจงแบบไวบูลล์



รูปที่ แสดงการแจกแจงแบบไวบูลล์

ประวัติผู้เขียน

นางสาวปราณี รัตนัง เกิดเมื่อวันที่ 22 สิงหาคม พ.ศ. 2503 ที่จังหวัดเชียงใหม่ สำเร็จการศึกษาปริญญาตรี วิทยาศาสตร์บัณฑิต(สถิติ) จากมหาวิทยาลัยเชียงใหม่ เมื่อปีการศึกษา 2525 และเข้าศึกษาต่อระดับปริญญาโท ในภาควิชาสถิติ บัณฑิตวิทยาลัยจุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2526 ปัจจุบันทำงานอยู่ที่ ฝ่ายวิชาการ ธนาคารแห่งประเทศไทย กรุงเทพมหานคร