บรรณานุกรม

ภาษาไทย

หนังสือ

มนตรี พิริยะกุล. เทคนิคการวิเคราะห์สมการถดถอย เล่ม 1. พิมพ์ครั้งที่ 4. กรุงเทพมหานคร : บริษัทรุ่งศิลป์การพิมพ์, 2529.

________. เทคนิคการวิเคราะห์สมการถดถอย เล่ม 2. พิมพ์ครั้งที่ 4. กรุงเทพมหานคร : โรงพิมพ์สำนักพิมพ์มหาวิทยาลัยรามคำแหง.

ศิริจันทร์ ทองประเสริฐ. การจำลองแบบปัญหา. พิมพ์ครั้งที่ 2. กรุงเทพมหานคร : โรงพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2533.

เอกสารอื่น ๆ

สมชัย ยืนนาน. "การศึกษาโดยวิธีมอนติคาร์โล เปรียบเทียบการทดสอบการเท่ากันของความแปรปรวนระหว่างประชากร 2 กลุ่ม". วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2528.

จีรพร วีระพันธุ์. "การศึกษาเปรียบเทียบวิธีการนอนพาราเมตริกสำหรับการประมาณค่าและการทดสอบสมมติฐานเกี่ยวกับพารามิเตอร์ของความถดถอยเชิงเส้นอย่างง่าย". วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2530.

มาลี ตระการศิรินนท์. "การเปรียบเทียบการประมาณค่าพารามิเตอร์ของรูปแบบสมการถดถอยเชิงเส้น ด้วยวิธีกำลังสองต่ำสุด และวิธีบูตสแตรป". วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2531.

ปราณี รัตนัง. "การประมาณสัมประสิทธิ์ถดถอยพหุเมื่อความผิดพลาดมีการแจกแจงแบบเบ้และแบบหางยาวกว่าปกติ". วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ จุฬาลงกรณ์มหาวิทยาลัย, 2531.

ทรงพันธ์ ชุณหสวัสดิกุล. "การประมาณค่าสัมประสิทธิ์การถดถอยพหุโดยที่ค่าประมาณสเกล-เปลี่ยนไป". วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2532.

ภาษาต่างประเทศ

หนังสือ

Frank, R. Hampel, Peter, J. Rousseeuw, Elvezis, M. Ronchette and Werner, A. Stahel. Rubust Statistics The Approch Based on influence Function. New York : Wiley, 1986.

Irwin Guttman. Linear Models : An Introduction. New York:John Wiley & Dpmd Inc, 1982.

Steven F. Arnold. The Theory of Linear Models and Multivariate Analysis" New York : John Wiley & Sons, 1981.

Thomas H. Wonnacott and Ronald J. Wonnacott. Regression a Second Course in Statistics. New York:John Wiley & Sons, 1981.

บทความในวารสาร

Dabid V. Hinkley. "Bootstrap Methods".Royal Statistical Society 50-(March 1988) : 321-369.

Raymond J. Carroll, C.F. Jeff Wu, and David Ruppert "The Effect of Estimating Weights in Weighted Least Squares" Journal of the American Statistical Association 83 (December 1988) : 1045-1054.

Bickel, P.J., and Freedman, D.A. "Some Asymptotic Theory for the Bootstrap. "The Annals of Statistics 9 (March 1981) : 1196-1217.

Freedman, D.A. "Bootstrapping Regression Models." The Annals of Statistics 9 (April 1981) : 1218-1228.

Freedman, D.A., and Peters, S.C. "Bootstrapping and Econometric Model : Some Empirical Results. "Journal of Business & Economic Statistics 2 (April 1984) : 150-158.

Efron, B. "Nonparametric Estimates of Standard Error : The Jackknife, the Bootstrap and other Methods. "Biometrika 68 (January 1981) : 589-99.

________. "Censored Data and the Bootstrap. "Journal of the American Statistical Association 76 (June 1981) : 312-319.

Efron, B. and Tibshirani, R. "Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy." Statistical Science 1 (January 1986:54-77).

Stine, A.R. "Bootstrap Prediction Intervals for Regression." Journal of the American Statistical Association 80 (December 1985): 1026-1030.

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

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.

Ranchetti, E. "Robust Model Selection in Regression." Statistic & Probability Letter 3 (1985) : 21-33.

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. การผลิตเลขสุ่มโดยการใช้โปรแกรม

เป็นการผลิตเลขสุ่มจากความสัมพันธ์ที่ซ้ำ ๆ (recurrence relation) กล่าวคือ เลขถัดไปเกิดจากการดำเนินการทางคณิตศาสตร์และตรรกศาสตร์ของตัวเลขปัจจุบัน หรือกลุ่มตัวเลขในอดีต อนุบรรพ (sequence) ของตัวเลขซึ่งผลิตได้ จึงเป็นอนุบรรพของเลขสุ่มในความหมายที่แท้จริงไม่ได้และย่อมมีคาบ (period) แต่อย่างไรก็ตาม เลขที่ผลิตในอนุบรรพเหล่านี้ อาจจะผ่านการทดสอบความเป็นสุ่มทางสถิติได้หลายอย่างและเรียกว่าเลขคล้ายสุ่ม (psedo-random number)

การผลิตเลขสุ่มโดยการใช้โปรแกรมมีข้อดีหลายประการที่สำคัญคือ วิธีนี้สามารถผลิตอนุบรรพของเลขชุดเดิมออกมาได้ซึ่งเป็นสิ่งจำเป็นอย่างยิ่งในกรณีที่ใช้การทดสอบแบบจำลองและมีความประสงค์จะทบทวนการคำนวณโปรแกรม

ข้อบกพร่องของวิธีการผลิตนี้ ก็คือ อนุบรรพของเลขสุ่มที่ผลิตออกมาเป็นอนุบรรพที่มีคาบทั้งสิ้นและการอนุมาณคุณสมบัติเชิงสถิติของเลขสุ่มเหล่านี้ทางทฤษฎีกระทำได้ยากพอสมควร

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

วิธีการแบบ Multiplicative congruentail method จะหาเลขสุ่มโดยทำการคำนวณจากสมการ

(1) $X_{i+1} = X_i, a$ (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}

2. การผลิตเลขสุ่มที่มีการแจกแจงสม่ำเสมอ

จากสมการการผลิตเลขสุ่มสามารถผลิตเลขสุ่มที่มีการแจกแจงสม่ำเสมอได้โดยตรงคือ

1) ค่า m เป็นค่าของจำนวนเต็มที่ใหญ่ที่สุด (largest integer) และเป็นเลขคี่ที่สามารถคำนวณได้จากเครื่องคอมพิวเตอร์ จาก $m = 2^b$ เมื่อ b เป็นค่าความยาว 1 word หรือจำนวน bit ใน 1 word ในการศึกษาครั้งนี้ใช้เครื่อง VAX 11-780 system 32 bit binary machine ซึ่ง bit สุดท้าย 1 bit ใช้สำหรับแสดงเครื่องหมาย ดังนั้น เลขจำนวนเต็มที่ใหญ่ที่สุดใน 1 word และเป็นเลขคี่ที่คอมพิวเตอร์ได้รับ คือ $2^{b-1}-1$ 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 เป็นเลขคู่จะพบว่า ทุก 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 จะเป็น

เลขอันดับแรกของความสัมพันธ์ระหว่างเลขคล้ายสุ่ม ดังนั้น เครื่อง VAX 11-780 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_i ค่าตัวเลขซึ่งมีการแจกแจงแบบสม่ำเสมอ คือ เศษจากการหาร $r1 = \frac{x_1}{2^b}$ ซึ่งจะมีค่าอยู่ระหว่าง 0 กับ 1 และ ค่า $0 < X_i < m$

โปรแกรมย่อยที่ใช้คือ RAND (IX, IY, YFL) ซึ่ง IX คือเลขสุ่มที่เป็นค่าเริ่มต้นที่เข้าไปในโปรแกรมย่อย IY คือ เลขสุ่มตัวถัดไปที่คำนวณได้จากเลขสุ่มเริ่มต้น YFL คือ เลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอในช่วง (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
```

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

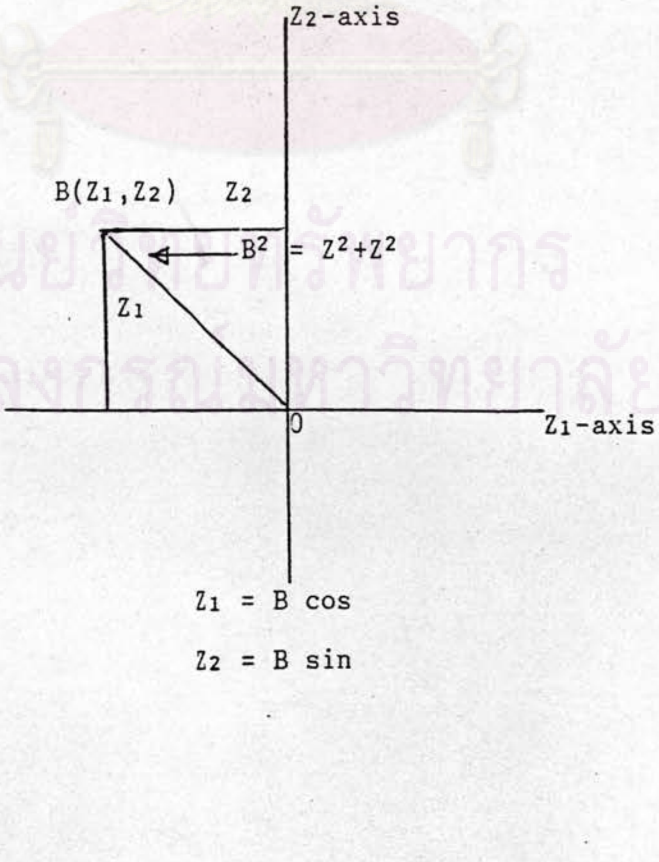
Lehmer ได้ทดลองเลือกใช้ค่า m,a และ X_o ที่จับคู่ต่าง ๆ กันเพื่อใช้ผลิตเลขคล้ายสุ่มตามสมการที่ (1) พบว่าถ้าเลือก X_o เป็นเลขคี่ และ $m = 2^r$ (เมื่อ $r>2$) และ $a = 8k \pm 3$ (เมื่อ k เป็นจำนวนเต็มบวกใด ๆ) จะได้คาบของเลขคล้ายสุ่มมากที่สุด และเท่ากับ 2^{r-2} วิธีการดังกล่าวเป็นวิธีการเลือกค่า parameter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

3. การผลิตเลขสุ่มที่มีการแจกแจงแบบปกติ

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 และค่าแปรปรวนเป็น 1 พร้อม ๆ กัน 2 ค่า ดังนี้



$Z_1 = B \cos$

$Z_2 = B \sin$

B^2 มีการแจกแจงไคสแควร์ (chi-square distribution) ด้วยระบบความเป็นอิสระ 2 ซึ่งเทียบเท่า (equivalent) กับการแจกแจงแบบเอ็กซ์โปเนนเชียล (exponential distribution) ด้วยค่าเฉลี่ย 2 ดังนั้น รัศมี B มีค่า ดังนี้

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

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

ซึ่งค่า B และ θ เป็น Mutually independent

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

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

ฟังก์ชั่นสำหรับจำลองแบบประชากรที่มีการแจกแจงเป็นแบบปกติ มีค่าเฉลี่ย μ และค่าเบี่ยงเบนมาตรฐาน σ จะเรียกใช้ FUNCTION NORMAL (DMEAN, SIGMA) ซึ่งจะได้จากค่า NORMAL = Z_1 x SIGMA + DMEAN หรือ NORMAL = Z_2 x 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)
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
      K      =  0
      RETURN
      END
```

4. การสร้างการแจกแจงแบบปกติปลอมปน

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

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

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

μ และ σ^2 เป็นค่ากำหนดค่าเฉลี่ และความแปรปรวน

p และ Ic 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. การสร้างการแจกแจงแบบลอกนอร์มอล

การแจกแจงแบบลอกนอร์มอลมีฟังก์ชันความหนาแน่นอยู่ในรูปของ

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

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

โดยมี exp (σ^2) เป็น scale parameter

และ μ เป็น shape parameter

ค่าคาดหวัง ค่าความแปรปรวน และค่าสัมประสิทธิ์ความแปรปรวนของการแจกแจงแบบลอกนอร์มอล คือ

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

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

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

ดังนั้นคำสั่งในการสร้างตัวแปรให้มีการแจกแจงแบบลอกนอร์มอล คือ

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

6. การสร้างการแจกแจงแบบแกมมา

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

การสร้างตัวแปรสุ่มให้มีการแจกแจงแบบแกมมา ใช้คุณสมบัติ 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 $\alpha = \sum_{i}^{n} \alpha_i$ ดังนั้น เมื่อ α เป็นตัวเลขค่าเต็ม หรือ $\alpha = m$ ตัวแปรจากการแจกแจงแบบแกมมา $G(m, \beta)$ สามารถผลิตได้โดยการ

รวมตัวแปรสุ่มแบบเอ็กซ์โปเนนเชียลที่เป็นอิสระ m ตัว ดังนี้

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

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

เมื่อ U_i เป็นตัวแปรสุ่มที่มาจากการแจกแจงแบบสม่ำเสมอมีพิสัยภายใน ช่วง 0 ถึง 1 ค่าคาดหวัง ค่าความแปรปรวนและค่าสัมประสิทธิ์ความแปรปรวนของการแจกแจงแบบแกมมา คือ

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

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

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

ดังนั้น คำสั่งในการสร้างตัวแปรให้มีการแจกแจงแบบแกมมา คือ

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

7. การสร้างการแจกแจงแบบดับเบิ้ลเอ็กซ์โปเนนเชียล

การแจกแจงแบบดับเบิ้ลเอ็กซ์โปเนนเชียล เป็นการแจกแจงซึ่งมีฟังก์ชันความน่าจะเป็นดังนี้

$$f(x) = \frac{1}{2\beta} e^{-\left|\frac{X-\alpha}{\beta}\right|} \qquad -\infty < X < \infty$$

$$-\infty < \alpha < \infty \ , \ \beta > 0$$

ถ้า $\alpha = 0$

$$f(x) = \frac{1}{2\beta} e^{-\left|\frac{X}{\beta}\right|} \qquad -\infty < x < \infty$$

$$\beta > 0$$

การสร้างตัวแปรสุ่มที่มีการแจกแจงแบบดับเบิ้ลเอ็กซ์โปเนนเชียล เมื่อ $\alpha = 0$ B = 5,10 ใช้วิธี Inverse Transformation ซึ่งแสดงได้ดังนี้

$$F(x) = \int_{-\infty}^{0} f(x)dx + \int_{0}^{x} f(x)dx$$

ซึ่ง

$$f(x = \begin{cases} \frac{1}{2\beta} e^{x/\beta} & \text{เมื่อ } x > 0 \\ \\ \frac{1}{2\beta} e^{x/\beta} + \frac{1}{2\beta} e^{-x/\beta} & \text{เมื่อ } x \geq 0 \end{cases}$$

พิจารณา เมื่อ $x < 0$

$$f(x) = \frac{1}{2\beta} e^{x/\beta}$$

$$F(x) = \int_{-\infty}^{0} \frac{1}{2\beta} e^{x/\beta} \, dx$$

$$= \frac{1}{2} \int_{-\infty}^{x} e^{x/\beta} \, d\left(\frac{x}{\beta}\right)$$

$$= \frac{1}{2} e^{x/\beta}$$

$$2F(X) = e^{x/\beta}$$

$$\frac{X}{\beta} = \mathrm{In}\ [2F(x)]$$

$$X = \beta\ [\mathrm{In}\ 2 + \mathrm{In}(F(X))]$$

พิจารณา เมื่อ $X \geq 0$

$$f(X) = \frac{1}{2\beta} e^{x/\beta} + \frac{1}{2\beta} e^{-x/\beta}$$

$$F(X) = \int_{-\infty}^{0} \frac{1}{2\beta} e^{x/\beta}\,dx + \int_{0}^{x} \frac{1}{2\beta} e^{-x/\beta}\,dx$$

$$= \frac{1}{2} \{ e^{\frac{x}{\beta}} \Big|_{-\infty}^{0} - e^{-\frac{x}{\beta}} \Big|_{0}^{x} \}$$

$$= \frac{1}{2} \{ e^{0} - e^{-\infty} - e^{-\frac{x}{\beta}} + e^{0} \}$$

$$= \frac{1}{2} \{ 2 - e^{-\frac{x}{\beta}} \}$$

$$e^{-\frac{x}{\beta}} = 2[1-F(x)]$$

$$\frac{-x}{\beta} = \text{In } 2 + \text{In } [1-f(x)]$$

$$X = -\beta\,[\text{In}2 + \text{In } (1-F(x))]$$

หรือ $$X = -\beta\,[\text{In}2 + \text{In}(1-YFL)]$$

ดังนั้นโปรแกรมย่อยซึ่งใช้สร้างการแจกแจงแบบดับเบิ้ลเอ็กซ์โปเนนเชียล แสดงได้ดังนี้

```
      SUBROUTINE DOUBLE (ALPHA, BETA, X)
      YFL (YFL - 0.5) 10, 10, 11
10    X = BETA * [ALOG (2.) + ALOG (YFL)]
      GOTO 15
11    Y = ALOG (2.) + ALOG (1. - YFL)
      X = -1. * BETA * y
15    RETURN
      END
```

8. การสุ่มตัวอย่างแบบใส่คืน (Sampling with replacement)

เป็นการสุ่มตัวอย่าง ที่ยอมให้มีหน่วยตัวอย่างซ้ำกันได้ นั่นก็คือแต่ละหน่วยตัวอย่างมีโอกาส (probability) ในการถูกสุ่มเท่ากัน = $\frac{1}{N}$ เมื่อ N เป็นขนาดของประชากร การวิจัยในครั้งนี้ ได้ใช้คอมพิวเตอร์เป็นเครื่องมือช่วยในการสุ่มตัวอย่างแบบใส่คืน โดยใช้ตัวเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอที่มีค่าอยู่ในช่วง [0,1] เป็นตัวเปรียบเทียบกับค่าความน่าจะเป็นสะสม (Cumulative Probability) เพื่อกำหนดหน่วยตัวอย่างตามจำนวนที่ต้องการ ซึ่งขั้นตอนการสุ่มตัวอย่างแบบใส่คืนพอจะสรุปได้ดังนี้

1. คำนวณหาค่าความน่าจะเป็นของแต่ละหน่วยตัวอย่าง = $\frac{1}{N}$
2. หาค่าความน่าจะเป็นสะสมแล้วจัดช่วง
3. สร้างตัวเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอซึ่งมีค่าอยู่ในช่วง [0,1]
4. นำตัวเลขสุ่มในข้อ 3 มาเปรียบเทียบกับค่าความน่าจะเป็นสะสม ถ้าตกอยู่ในช่วงใด หน่วยนั้น ๆ จะถูกเลือกมาเป็นตัวอย่าง
5. กระทำตามขั้นตอนในข้อ 3-4 n ครั้ง เมื่อ n คือขนาดตัวอย่างที่ต้องการ

ดังนั้นโปรแกรมย่อยที่ใช้ในการสุ่มตัวอย่างแบบใส่คืน แสดงได้ดังนี้

```
      SUBROUTINE WR(X)
      DO 30 J = 1,n
      YFL = RAN(IX)
      DO 10 I = 1,N
      IF ((YFL .GT. PP (I-1).AND.(YFL .LE. PP(I)) THEN
         X(J) = POP(I)
         GOTO 30
      END IF
10    END DO
30    END DO
```

เมื่อ n เป็นขนาดของตัวอย่าง

N เป็นขนาดของประชากร

PP เป็นค่าความน่าจะเป็นสะสม (Cumulative Probability)

X เป็นค่าของตัวอย่างที่ได้จากการสุ่มแบบใส่คืน ซึ่ง X แต่ละตัวอาจมีค่าซ้ำกันได้

สำหรับการวิจัยในครั้งนี้ X คือค่า ε^* ที่ได้จากการสุ่ม $\hat{\varepsilon}_i$; $i = 1,2,\ldots\ldots,N$ เมื่อ $\hat{\varepsilon}_i = y_i - \hat{y}_i$ แล้วนำค่า ε^* ไปใช้ในสมการ $Y^* = X\hat{\beta} + \varepsilon^*$ เพื่อคำนวณหาค่า $\hat{\beta}^* = (X'X)^{-1} X'Y^*$ และ $\hat{\bar{\beta}}^*$ ต่อไป

ภาคผนวก ข.

ภาคผนวก ข.

```
C***********************************************************************
C************************  MAIN PROGRAM  ****************************
C***********************************************************************
      DOUBLE PRECISION A,S,B.B1,BMMAD,B2,X.X1,Y1,FL
      COMMON /REGRS/A(11,11),S(12,12)/CONTA/P,IC,SKEWED/ALPHA,BETA
     *       /COEFF/B(11)/COEFF2/BMMAD(11)/COEFF4/BB(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/MSEO(3),MSEM(3),MSEB(3)
      DIMENSION B2(11),STD(3),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
      MSEM(I)  =0.0
      MSEB(I)  =0.0
      STD(I)   =0.0
      IK(I)    =0.0
    1 MSEO(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=1,6
      READ (5,17)A2(N-I+1)
   17 FORMAT(F6.4)
   16 CONTINUE
      DO 20 I = 1,2
   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)
  106 FORMAT(10X,I3,'TH ROUND',/10X,___________')
      CALL INIT
      CALL DATA
      DO 1000 J = 1,3
      DO 1000 K = 1,4
 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  = 1,4
  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
    MSEM(J)   = MSEM(J) + SD(J) ** 2
425 MSEM(J)   = MSEM(J) + SD(J)
130 CONTINUE
    DO 450 I  = 1,3
450 MSEM(I)   = MSEM(I)/JJ
    WRITE(6,475)(MSEM(I), I=1,3)
475 FORMAT('===> AVERAGE MEAN SQUARE ERROR FOR B(I) ',
   * 3(1XF10.4))
    RETURN
    END
*****************************************************************************
******************** SUBROUTINE CREATE VARIABLE ****************************
*****************************************************************************
    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/VALIAV/N,M,M2/TABLE/A2(150),SE(9)
   *       /REGRS/A(11,11),S(12,12)/SKEWED/ALPHA,BETA
    DO 90 J   = 1,4
```

```
   90 X(1,J)     = 1.0
      DO 100 I   = 2,3
      DMEAN      = XBAR1(I-1)
      SIGMA      = SG(I-1)
      DO 100 J   = 1,4
      X(I,J)     = NORMAL(DMEAN,SIGMA)
  100 CONTINUE
C******************************************************************************
C********************** CALCULATE X'X & X'Y FOR OLS ***************************
C******************************************************************************
      DO 120 I   = 1,2
      DO 120 K   = 1,2
      SIK        = 0.0
      DO 110 J   = 1,4
  110 SIK        = SIK + X(I,J) * X(K,J)
      S(I,K)     = SIK
  120 S(K,I)     = SIK
C*****************************************************************************
C*********************** CALCULATE INV. MATRIX OF X'X ************************
C*****************************************************************************
      DO 140 I   = 1,2
      DO 140 J   = 1,2
      A(I,J)     = S(I,J)
  140 A(J,I)     = S(I,J)
      DO 145 K   = 1,2
      IF (A(K,K)) 145,146,145
  146 WRITE(6,150)
```

```
150  FORMAT('A(K,K) HAS SERO IN DIAGONAL CANNOT USE THIS
    * MATRIX')
     STOP
145  CONTINUE
     CALL INVS(M,A)
     RETURN
     END
C**************************************************************************
C******************** CREATE OUTLIER DISTRIBUTION *************************
C**************************************************************************
     SUBROUTINE DATA
     REAL NORMAL
     DOUBLE PRECISION B,X,Y,A2,SW,E,T,B1
     COMMON /COEFF/B(11)/SEED/IX,IX2/SELECE/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   = 1,4
     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
     DO 106 J   = 1,4
     SUM        = 0.0
     DO 105 I   = 1,2
     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   = 1,4
130  WRITE(6,135)(X(I,J),I = 1,M2)
135  FORMAT (12(2X,F11.4)
     RETURN
     END
*****************************************************************************
************************ SUBROUTINE RANDOM ***********************************
*****************************************************************************
     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
*******************************************************************************
************************ FUNCTION NORMAL DISTRIBUTION *************************
*******************************************************************************
    FUNCTION NORMAL(DMEAN,SIGMA)
    REAL NORMAL
    COMMAN /SEED/IE,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 GAMMA 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.EQ.1.0) GOTO 10
         ALPHA = ALPHA - 1.0
         GOTO 5
   10 GAMMA1   = BETA1 * U
      RETURN
      END
C*******************************************************************************
C*************** FUNCTION DOUBLE EXPONENTIAL DISTRIBUTION **********************
C*******************************************************************************
      FUNCTION DOUBLE(ALPHA,BETA,Z)
      COMMON /SEED/IX
      YFL = RAN(IX)
      IF (YFL - 0.5) 10,10,15
   10 Z    = BETA * (ALOG(2.0) + ALOG(YFL))
      GOTO 20
   15 S    = ALOG(2.0) + ALOG(YFL)
      Z    = -1.0 * BETA * S
```

```
   20 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   = 1,3
      SIK       = 0.0
      DO 10 J   = 1,4
   10 SIK       = SIK + X(I,J) * X(M2,J)
   20 S(M2,1)   = SIK
      DO 50 I   = 1,2
   50 B2(I)     = B2(I) + A(J,I) * S(M2,J)
      RETURN
      END
C*****************************************************************************
C****************** SUBROUTINE M-ESTIMATOR (B) *******************************
C*****************************************************************************
      SUBROUTINE MEST2
      DOUBLE PRECISION A1,S1,X1,W,S2,Z,B,B1,YHAT,YRES,BMMAD,
     * X,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)
      DO  10 I = 1,3
      DO  10 j = 1,4
   10 X1(I,J) = X
      DO 20 I = 1,3
      DO 20 J = 1,4
   20 X1(I,J) = X(I,J)
      CALL BOXC2(BMMAD,FL1,SIGMA)
      CALL SEE (BMMAD,SD(2))
      SIGMA1 = SIGMA
      KCC = 10
      DO 40 I = 1,3
      DO 40 J = 1,4
   30 CONTINUE
      RETURN
      END
*****************************************************************
********** SUBROUTINE OLS1(B) FOR M-ESTIMATOR ********************
*****************************************************************
      SUBROUTINE OLS1(B2)
      DOUBLE PRECISION A1,S1,B2,X,W
      COMMON /REGRE1/A1(11,11),S1(12,120)/VARIAB/N,M,M2
     *       /DATAXY/X(12,150)/WEIGHT/W(150)
```

```
      DIMENSION B2(11)
      DO 20 I = 1,3
      DO 20 K = I,3
      SIK     = 0.0
      DO 10 J = 1,4
   10 SIK     = SIK + X(I,J) * X(K,J) * W(J)
      S1(1,K) = SIK
   20 S1(K,I) = SIK
      DO 40 I = 1,2
      DO 40 J = 1,2
      A1(J,I) = S1(I,J)
   40 A1(J,I) = S1(I,J)
      DO 45 K = 1,2
      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 = 1,2
      B2(I)   = 0.0
      DO 50 J = 1,2
   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   = 1,4
      YHAT(J)   = 0.0
      DO 20 I   = 1,2
   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   = 1,2
      A(K,K)    = -1.0/A(K,K)
      DO 5 I    = 1,2
      IF (I,K) 3,5,3
    3 A(I,K)    = -A(I,K) * A(K,K)
    5 CONTINUE
      DO 10 I   = 1,2
      DO 10 J   = 1,2
      IF 9(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    = 1,2
      IF  (J,K) 18,20,18
   18 A(K,J)     = -A(K,J) * A(K,K)
   20 CONTINUE
      DO 25 I    = 1,2
      DO 25 J    = 1,2
   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(150)/REANSF/Y1(150),FL(3),ST,FIN
     *       /VARIAB/N,M,M2/STORE/BMEST(3,11)
      DIMENSION B(11)
      DO 30 J = 1,4
      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 = 1,4
```

```
50 SLG =SLG + ALOG(Y(J))
   SLG = SLG/N
   G   = EXP(SLG)
   DO 60 J = 1,4
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.SSE1.LE.SSE3) GOTO 72
   IF (SSE3.LE.SSE1.AND.SSE1.LE.SSE2) GOTO 73
71 FM   = ST
   SSE2 = SSE1
   DO 80 I = 1,2
80 BMEST (2,I = BMEST(1,I))
   GOTO 100
72 FM = FD
   GOTO 100
73 FM = FIN
   SSE2 = SSE3
   DO 90 I = 1,2
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 = 1,2
  120 B(I) = BMEST(2,I)
      FL(1) = FM
      RETURN
      END
C**********************************************************************
C*************** SUBROUTINE TEST MULTICOLINEARITY *******************
C**********************************************************************
      SUBROUTINE SHAPWK (N,Y,ITEST)
      DOUBLE PRECITION 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 = 1,4
      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 = 1,6
   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,K)
   DOUBLE PRECISION X,T
   DIMENSION X(150)
   N1 = N-1
   DO 10 I = 1,3
   II = I+1
   DO 10 K = II,4
   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 = 1,3
      II      = I+1
      DO 10 K = II,4
      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 BCOX *******************************
C*********************************************************************
      SUBROUTINE BCOX (FL1,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  = 1,4
   10 X1(M2,I) = DLOG(Y1(I))
      GOTO 30
   15 DO 20 I  = 1,4
   20 X1(M2,I) = ((Y1(I) ** FL(K) - 1.0 / FL(K)
   30 RETURN
      END
C*****************************************************************************
C******************  SUBROUTINE TO 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 = 1,2
   15 Z(I)     = S(M2,I)
      ZZ       = S(M2,M2)
      DO 80 I = 1,2
```

```
   80 BMEST(K,I) = B(I)
      SSR      = 0.0
      DO 90 I = 1,2
   90 SSR      = SSR + B(I) * Z(I)
      SSE      = ZZ - SSR
      SSE      = SSE/FLOAT(N)
      RETURN
      END
C**************************************************************************
C***************  SUBROUTINE FOR M-ESTIMATOR  ******************************
C**************************************************************************
      SUBROUTINE MAD(B1,BMMAD,SIGMA)
      DOUBLE PRECISION E,BMMAD,B2,SUME,Z,YHAT,YRES,W,B1,SIG1,X1,SIGMA
      COMMON /DATXY1/X1(12,150)/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  = 1,4
    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 MIDIAN  = (D(N3)+E(N4))/2
18 DO 20 I = 1,4
20 SIGI(I) = DABS(E(I)-MEDIAN)
   CALL RANK(N,SIG1)
   IF (N-N1) 25,27,25
25 SIGMA   = SIGI(N2)/0.6745
   GOTO 28
27 SIGMA   = (SIG1(N3)+SIG1(N4))/(2*0.6745)
28 L       = 0
   CALL YRESID(S1,N,M,YHAT,YRES,B1)
   DO 29 I = 1,2
29 B2(I)   = B1(I)
77 DO 30 I = 1,4
   Z(I)    = DABS(YRES(I)/SIGMA)
   W(I)    = 2.0/Z(I)
   IF (W(I).GE.2.0) GOTO 555
     GOTO 30
55 W(I)    = 1.0
30 CONTINUE
   L       = L + 1
   IF (L-20) 60,60,90
60 CALL OLS1(BMBAR)
   DO 70 I = 1,2
   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 = 1,2
```

```
   80 B2(J)   = BMBAR(J)
      GOTO 777
   70 CONTINUE
   90 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 = 1,2
   10 SE      = SE + (B(I)-B2(I))**2
      SE      =SE/M
      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)
     *       /DATWY1/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)
      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 = 1,2
      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
    DO 45 I = 1,2
    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= 1,2
110 BMMAD(I)= BMEST(K,I)
    FL(3)    = FL1
    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,LF,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 =1,2
   20 BMEST(K,I)  = BMMAD(I)
      SSR     = 0.0
      DO 30 I = 1,2
   30 SSR     = SSR + BMMAD(I) * Z(I)
      SSE     = SS - SSR
      SSE     = SSE/FLOAT(N)
      RETURN
      END
```

```
C***************************************************************************
C************ SUBROUTINE FOR BOOTSTRAP METHOD ******************************
C***************************************************************************
      SUBROUTINE RANERR(B1,X)
      REAL NORMAL
      DIMENSION E(150),E1(150),YHAT(150),YRES(150),SUM1,B1(11),X(11,150),
     *          1YRESS(150),PP(150),YFL
      COMMON /ERR/E,YHAT,YRES,YRESS,E1,
     *       /VARIAB/N,M,M2,NB,/SEED/IX,/PROB/PP
      DO 30 J = 1,4
      YFL     = RAN(IX)
      DO 10 I = 1,10
      IF (YFL .GT .PP(I-1) .AND. YFL .LE. PP(I)) GOTO 5
    5 E1(J)= YRESS(I)
      GOTO 30
   10 END DO
   30 END DO
      DO 40 J = 1,4
      SUM1    = 0
      DO 50 I = 1,2
      SUM 1   = SUM1 + B1(I)*X(I,J)
   50 END DO
      X(M2,J) = SUM1 + E1(J)
   40 END DO
      RETURN
      END
```

```
C*******************************************************************
C************ SUBROUTINE FOR SAMPLING WITH REPLACEMENT **************
C*******************************************************************
      SUBROUTINE WR(X)
      DO 30 J =1,n
      YFL = RAN(IX)
      DO 10 I = 1,N
      IF ((YFL .GT. PP (I-1).AND.(YFL .LE.PP(I)) THEN
           X(J) = POP(I)
           GOTO 30
      END IF
   10 END DO
   30 END DO
```

ประวัติผู้วิจัย

นายปฐม กลั่นน้ำทิพย์ เกิดเมื่อวันที่ 7 กรกฎาคม 2503 ที่ตำบลปากจั่น อำเภอนครหลวง จังหวัดพระนครศรีอยุธยา สำเร็จการศึกษาชั้นมัธยมศึกษาตอนปลายจากโรงเรียนอยุธยาวิทยาลัย พ.ศ. 2521 สำเร็จการศึกษา "การศึกษาบัณฑิต" สาขาวิชาเอกคณิตศาสตร์ จากมหาวิทยาลัยศรีนครินทรวิโรฒ (บางแสน) พ.ศ. 2525 ปัจจุบันรับราชการในตำแหน่งนักวิชาการศึกษา 5 ระดับ 5 หัวหน้างานบริการการศึกษา คณะสาธารณสุขศาสตร์ มหาวิทยาลัยขอนแก่น