บรรณานุกรม

<u>ภาษาไทย</u>

ธีรพร วีระพันธุ์ "การศึกษาเปรียบเทียบวิธีการนอนพาราเมตริกสำหรับการประมาณค่าและการทดสอบสมมติฐานเกี่ยวกับพารามิเตอร์ของความถดถอยเชิงเส้นแบบง่าย" วิทยานิพนธ์ ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2529.

สมชัย ยืนนาน "การศึกษาโดยวิธีมอนติคาร์โลเปรียบเทียบอำนาจของการทดสอบการเท่ากันของความแปรปรวนระหว่างประชากรสองกลุ่ม" วิทยานิพนธ์ ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2528.

<u>ภาษาอังกฤษ</u>

Daniel, C. and Wood, F.S. <u>Fitting Equations to Data.</u> New York: Wiley-Interscience, 1980.

Douglas, C.M. and Elizabeth, A.T. <u>Introduction to Linear Regression Analysis</u>. New York : Wiley, 1982.

Draper, N.R. and Smith, H. <u>Applied Regression Analysis</u>. New York: John Wiley & Sons, 1966.

Graybill, F.A. <u>Theory and Application of The Linear Model</u>. California: Wadsworth Publishing Company, Inc, 1976.

Kempthorne, O. <u>The Design and Analysis of Experiments</u>. New York: John Wiley & Sons, 1952.

Articles

Berk, K.N. "Valiadation Regression Procedures with New Data. Technometrics 26 (November 1984) : 331-338.

Chow, G.C. "Tests of Equality between Sets of Coefficients in Two Linear Regression." Econometrica 28 (July 1960): 591-605.

Copas, J.B. "Regression, Prediction, and Shrinkage." Journal of the Royal Statistical Society B45 (1983) : 311-335.

Dempster, A.P., Schatzoff, M. and Wermuth, N. "A Simulation Study Alternatives to Least Squares." Journal of the American Statistical Association 72 (1977) : 77-106.

Geisser, S. "The Predictive Sample Reuse Method with Applications." Journal of the American Statistical Association 70 (1975): 320-328.

Gorman, J.W. and Toman, R.J. "Selection of Variables for Fitting Equations to Data." Technometrics 8 (1966): 27-51.

Hoerl, A.E. and Kennard, R.W. "Ridge Regression: Applications to Nonorthogonal Problems." Technometrics 12 (1970): 69-82.

Kennard, R.W. and Stone, L. "Computer Aided Design of Experiments." Technometrics 11 (1969) : 137-148.

Lindley, D.V. and Smith, A.F.M. "Bayes Estimates for The Linear Model." Journal of the Royal Statistical Society B33 (1972): 1-41.

Marquardt, D.W. and Snee, R.D. "Ridge Regression in Practice." The American Statistician 29 (1975): 3-19.

Snee, R.D. "Validation of Regression Models : Methods and Examples." Technometrics 19 (1977) : 415-428.

Stone, M. "Cross-validating Choice and Assessment of Statistical Predictions (with discussion)." Journal of the Royal Statistical Society B36 (1974) : 111-147.

ภาคผนวก

ภาคผนวก ก

<u>การสร้างตัวเลขสุ่ม</u> (Random Number)

ในการสร้างลักษณะการแจกแจงแบบต่าง ๆ นั้น จะต้องใช้ตัวเลขสุ่มเป็นพื้นฐานในการสร้าง สำหรับวิธีการสร้างตัวเลขสุ่มมีอยู่หลายวิธี ซึ่งในการวิจัยครั้งนี้จะใช้วิธีการสร้างเลขสุ่มตามวิธีที่ไวท์และชมิดท์ (1975 : 421) เสนอไว้ ซึ่งจะใช้โปรแกรมย่อย RANDOM ผลิตเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอในพิสัย 0 ถึง 1.0 โดยใช้คำสั่ง CALL RANDOM (IX,IY,RD) ซึ่งมีพารามิเตอร์ในวงเล็บ IX คือ เลขสุ่มตัวแรกซึ่งจะต้องเป็นจำนวนเต็มบวกที่เป็นเลขคี่ และน้อยกว่า 2147483648 ซึ่ง IX นี้จะเป็นค่าเริ่มต้นที่จะให้โปรแกรมย่อยคำนวณ IY ออกมาให้ IY จึงเป็นค่าที่เป็นเลขสุ่มจำนวนเต็มของโปรแกรมย่อยนี้ และจะใช้เป็นตัวคำนวณ IY ตัวต่อ ๆ ไป สำหรับรายละเอียดในการสร้างโปรแกรมย่อยสามารถแสดงได้ดังนี้

```
      SUBROUTINE RANDOM (IX,IY,RD)
      IY  =  IX * 65539
      IF (IY) 1,2,2
1     IY = IY + 2147483647 + 1
2     RD = IY
      RD = RD * .4656613E - 9
      IX = IY
      RETURN
      END
```


<u>การสร้างการแจกแจงแบบโลจิสติค</u>

การแจกแจงแบบโลจิสติค เป็นการแจกแจงซึ่งมีฟังก์ชั่นความน่าจะเป็น เป็นดังนี้

$$f(x) = \frac{e^{-\left(\frac{x-\alpha}{\beta}\right)}}{\beta\left[1+e^{-\left(\frac{x-\alpha}{\beta}\right)}\right]^2} \quad , \quad -\infty < x < +\infty$$

$$\alpha \; , \; \beta > 0$$

การสร้างตัวแปรสุ่มที่มีการแจกแจงแบบโลจิสติค ใช้วิธี Inverse Transformation ซึ่งแสดงได้ดังนี้

$$F(x) = \int_{-\infty}^{x} \frac{e^{-\left(\frac{x-\alpha}{\beta}\right)}}{\beta\left[1+e^{-\left(\frac{x-\alpha}{\beta}\right)}\right]^2}\, dx$$

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

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

$$= \left.\frac{1}{1+e^{-\left(\frac{x-\alpha}{\beta}\right)}}\right|_{-\infty}^{x}$$

$$= \frac{1}{1+e^{-\left(\frac{x-\alpha}{\beta}\right)}}$$

$$e^{-\left(\frac{x-\alpha}{\beta}\right)} = \frac{1-F(x)}{F(x)}$$

$$-\left(\frac{x-\alpha}{\beta}\right) = \ln\left[\frac{1-F(x)}{F(x)}\right]$$

$$x = \alpha + \beta \left[\ln(F(x)) - \ln(1-F(x)) \right]$$

หรือ $X = \alpha + \beta \left[\ln(RAN) - \ln(1-RAN) \right]$ เมื่อ RAN มีการแจกแจงแบบยูนิฟอร์มในช่วง [0,1]

ดังนั้นโปรแกรมย่อยซึ่งใช้สร้างการแจกแจงแบบโลจิสติคแสดงได้ดังนี้

```
SUBROUTINE LOGIST (ALPHA, BETA, X)
CALL RANDOM (IX,IY,RAN)
S  =  ALOG(RAN) -ALOG(1.-RAN)
X  =  ALPHA + S*BETA
RETURN
END
```

<u>การสร้างการแจกแจงแบบดับเบิลเอ็กซ์โปเนนเชียล</u>

การแจกแจงแบบดับเบิลเอ็กซ์โปเนนเชียล เป็นการแจกแจงซึ่งมีฟังก์ชั่นความน่าจะเป็นเป็นดังนี้

$$f(x) = \frac{1}{2\beta} e^{-\left|\frac{x-\alpha}{\beta}\right|} \qquad \begin{array}{l} -\infty < x < \infty \\ -\infty < \alpha < \infty \ , \ \beta > 0 \end{array}$$

ถ้า $\alpha = 0$

$$f(x) = \frac{1}{2\beta} e^{-\left|\frac{x}{\beta}\right|} \qquad \begin{array}{l} -\infty < x < \infty \\ \beta > 0 \end{array}$$

การสร้างตัวแปรสุ่มที่มีการแจกแจงแบบดับเบิลเอ็กซ์โปเนนเชียล เมื่อ $\alpha = 0$ ใช้วิธี Inverse Transformation ซึ่งแสดงได้ดังนี้

$$F(x) = \int_{-\infty}^{x} \frac{1}{2\beta} e^{-\left|\frac{x}{\beta}\right|} dx$$

$$= \frac{1}{2}\left[\int_{-\infty}^{0} e^{\frac{x}{\beta}} d\left(\frac{x}{\beta}\right) + \int_{0}^{x} e^{-\frac{x}{\beta}} d\left(\frac{x}{\beta}\right)\right]$$

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

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

$$= \frac{1}{2}\left[2 - e^{-\frac{x}{\beta}}\right]$$

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

$$-\frac{x}{\beta} = \ln 2 + \ln\left[1-F(x)\right]$$

$$x = -\beta\left[\ln 2 + \ln\left(1-F(x)\right)\right]$$

หรือ $$X = -\beta\left[\ln 2 + \ln(1-RAN)\right]$$

ดังนั้นโปรแกรมย่อยซึ่งใช้สร้างการแจกแจงแบบดับเบิ้ลเอ็กซ์โปเนนเชียล แสดงได้ดังนี้

```
SUBROUTINE DOUBLE (ALPHA,BETA,X)
CALL RANDOM (IX,IY,RAN)
Y  =  ALOG(2.)+ALOG (1.-RAN)
X  = -1*BETA*Y
RETURN
END
```

การสร้างการแจกแจงแบบปกติ

การสร้างตัวแปรสุ่มที่มีการแจกแจงแบบปกติที่มีค่าเฉลี่ย และส่วนเบี่ยงเบนมาตรฐานตามที่กำหนด จะใช้โปรแกรมย่อย NORMAL[1] ซึ่งจะพิจารณาจากสูตร

$$X = \frac{\sum_{i=1}^{k} RD_i - \frac{k}{2}}{\frac{k}{12}}$$

โดย X เป็นตัวเลขสุ่มที่มีการแจกแจงแบบปกติที่มีค่าเฉลี่ย 0 และค่าความแปรปรวน 1

RD_i เป็นตัวเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอจากโปรแกรมย่อย RANDOM

k เป็นจำนวนค่าของ RD_i ที่จะถูกนำมาใช้

โดยปกติแล้ว ตัวเลขสุ่ม X จะมีค่าเข้าใกล้เลขสุ่มที่มีการแจกแจงแบบปกติที่แท้จริงนั้น เมื่อค่าของ k เข้าใกล้ค่าอนันต์ (infinity) สำหรับโปรแกรมที่ใช้สร้างเลขสุ่มนี้จะเลือก k เป็น 12 เพื่อลดเวลาการคำนวณในเครื่องคอมพิวเตอร์ ดังนั้นจากสูตรข้างต้นจะได้สูตรใหม่ดังนี้

---

[1] System/360 Scientific Subroutine Package (360A-CM-o3X) Version III p.77-78.

$$X = \sum_{i=1}^{12} RD_i - 6.0$$

และเพื่อให้ตัวเลขสุ่มที่สร้างขึ้นมาแจกแจงเข้าใกล้การแจกแจงแบบปกติโดยที่มีค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานตามที่กำหนด ดังนั้นตัวแปรสุ่มดังกล่าวจะเป็น

$$x' = X \times S + AM$$

โดยที่ S เป็นค่าเบี่ยงเบนมาตรฐานตามที่กำหนด

AM เป็นค่าเฉลี่ยตามที่กำหนด

ดังนั้นโปรแกรมย่อย ซึ่งใช้สร้างการแจกแจงแบบปกติ แสดงได้ดังนี้

```
      SUBROUTINE NORMAL (SMEAN, SIGMA,X)
      A  =  0.
      DO 50 I = 1, 12
      CALL RANDOM (IX,IY,RAN)
      A = A+RAN
50    CONTINUE
      X = (A-6)*SIGMA+SMEAN
      RETURN
      END
```


<u>การสร้างการแจกแจงแบบปกติปลอมปน</u>

การสร้างตัวแปรสุ่มที่มีการแจกแจงแบบปกติปลอมปนที่มีค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานตามที่กำหนด จะใช้วิธีที่แรมเซย์ (Ramsay 1977) เสนอไว้ โดยพิจารณาการแจกแจงซึ่งแปลงมาจากการแจกแจงแบบปกติ ซึ่งมีฟังก์ชั่นการแปลงเป็นดังนี้

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

หมายความว่าค่า x จะมาจากการแจกแจง $N(\mu, \sigma^2)$ ด้วยความน่าจะเป็น 1-p และจากการแจกแจง $N(\mu, c^2\sigma^2)$ ด้วยความน่าจะเป็น p โดยที่

μ และ σ^2 เป็นค่ากำหนดค่าเฉลี่ยและความแปรปรวนของ ε_i

p และ c เป็นค่ากำหนดสัดส่วนการปลอมปน และสเกลแฟคเตอร์

ดังนั้นโปรแกรมย่อย ซึ่งใช้สร้างการแจกแจงแบบปกติปลอมปน แสดงได้ดังนี้

```
      SUBROUTINE SCALE  (C,P,SMEAN,SIGMA,X)
      CSIGMA =  C*SIGMA
      CALL RANDOM (IX,IY,RAN)
      IF(RAN-P)10,10,11
10    CALL NORMAL (SMEAN, CSIGMA,X)
      GOTO 15
11    CALL NORMAL (SMEAN,SIGMA,X)
15    RETURN
      END
```

ภาคผนวก ข

```
C
C     #==============================================================#
C     #          PROGRAM TO SPLITTING DATA & COMPUTE                 #
C     #   - COEFFICIENT FROM ALL DATA,ESTIMATION DATA AND            #
C     #     PREDICTION DATA                                          #
C     #   - (|X'X(EST)|/|X'X(PRE)|)**(1/P)                           #
C     #   - PROBABILITY OF TYPE 1 ERROR OF CHOW TEST                 #
C     #==============================================================#
C     #              DESCRIPTION SOME VARIABLES                      #
C     #          X(J) = INDEPENDENT VARIABLE                         #
C     #          Y(J) = DEPENDENT VARIABLE                           #
C     #          E(J) = ERROR                                        #
C     #          NP   = NUMBER OF INDEPENDENT VARIABLE               #
C     #==============================================================#
      DIMENSION AA(100,100,12),BB(100,100,12),X(100,12),XX(100,12),
     *Z(100,12),ZZ(100,12),A(12,12),D(100,100),T(12,12),XTX(12,12),
     *EST(100,12),PRE(100,12),ES(100,12),PR(100,12),DIFX(100,12),
     *E(100),Y(100),D1(100),D2(100),D3(100),D4(100),IE(100),IP(100),
     *IDE(100),IDP(100),YE(100),YP(100),DMINE(100),DMINP(100),
     *ESY(100),PRY(100),DD1(100),DD2(100),DD3(100),DD4(100),DIF(100),
     *BX(100),BXD1(100),BXD2(100),AAA(100),IJK(100),MK(100),
     *XXBAR(12),B(12),B1(12),BD1(12),BD2(12),SS(12),
     *X1(12),X2(12),X3(12),X4(12),BIT(12)
      COMMON IX
      N=100
      N1=N-1
      NP=4
      NP1=NP+1
      NP2=NP+2
      N001=0
      N005=0
      F001=3.255
      F005=2.33
      READ(5,10)SMEAN1,SIGMA1
10    FORMAT(2F3.0,F1.0)
      READ(5,20)ALPHA1,BETA1
20    FORMAT(F1.0,F7.6)
C     #==============================================================#
C     #      GENERATE FIXED VARIABLE ( (X(J)~N (50,100) )            #
C     #==============================================================#
      AIT=50.
      DO 1 I=2,NP1
      BIT(I)=1.
1     CONTINUE
      IX=973253
      DO 5 I=2,NP1
      DO 3 J=1,N
      CALL NORMAL(SMEAN1,SIGMA1,XX(J,I))
3     CONTINUE
5     CONTINUE
      TDET=0.
C     #==============================================================#
C     #           REPEAT SIMULATION 100 TIMES                        #
C     #==============================================================#
      NT=100
      DO 1500 LEK=1,NT
C     #==============================================================#
C     #      GENERATE E(J) & Y(J) : Y(J) = A+BX(J)+E(J)              #
C     #                      E(J)~L(0,100)                           #
C     #==============================================================#
      DO 7 J=1,N
      CALL LOGIST(ALPHA1,BETA1,E(J))
7     CONTINUE
      DO 11 J=1,N
      SB=0.
```

```
      DO 9 I=2,NP1
      SB=SB+(BIT(I)*XX(J,I))
9     CONTINUE
      Y(J)=AIT+SB+E(J)
11    CONTINUE
C     *==============================================================*
C     *                FIND COEFFICIENT FROM ALL DATA                *
C     *==============================================================*
      DO 13 J=1,N
      XX(J,1)=1.0
      XX(J,NP2)=Y(J)
13    CONTINUE
      CALL OLS(B,XX,N,NP1,NP2)
      DO 15 I=1,NP1
      B1(I)=B(I)
15    CONTINUE
      DO 23 J=1,N
      TBX=0.
      DO 21 I=2,NP1
      TBX=TBX+(B1(I)*XX(J,I))
21    CONTINUE
      BX(J)=TBX
23    CONTINUE
      SUMT=0.
      DO 25 J=1,N
      SUMT=SUMT+(Y(J)-(B1(1)+BX(J)))**2
25    CONTINUE
      AT=SUMT
C     *==============================================================*
C     *            SPLITTING DATA WITH DUPLEX ALGORITHM              *
C     *==============================================================*
C     * STANDARDIZE INDEPENDENT VARIABLES *
      DO 29 J=2,NP1
      SUMXX=0.
      DO 27 IK=1,N
      SUMXX=SUMXX+XX(IK,J)
27    CONTINUE
      XXBAR(J)=SUMXX/N
29    CONTINUE
      DO 35 J=2,NP1
      SUMSS=0.
      DO 31 IK=1,N
      DIFX(IK,J)=(XX(IK,J)-XXBAR(J))**2
      SUMSS=SUMSS+DIFX(IK,J)
31    CONTINUE
      SOS=SQRT(SUMSS)
      SS(J)=SOS
35    CONTINUE
      DO 39 J=2,NP1
      DO 37 IK=1,N
      X(IK,J)=(XX(IK,J)-XXBAR(J))/SS(J)
37    CONTINUE
39    CONTINUE
      DO 43 J=2,NP1
      JK=J-1
      DO 41 IK=1,N
      Z(IK,JK)=X(IK,J)
41    CONTINUE
43    CONTINUE
C     * ORTHONORMALIZE INDEPENDENT VARIABLES *
      CALL PRIME(Z,N,NP,XTX)
      CALL ORTHO(XTX,NP,T)
      DO 47 LK=1,NP
      DO 45 J=1,NP
      A(LK,J)=T(LK,J)
45    CONTINUE
47    CONTINUE
```

```
      NX=0
      DO 51 IK=1,NP
      DO 49 J=1,NP
      NX=NX+1
      AAA(NX)=A(IK,J)
 49   CONTINUE
 51   CONTINUE
      CALL MINV(AAA,NP,DDD,IJK,MK)
      NX=0
      DO 55 IK=1,NP
      DO 53 J=1,NP
      NX=NX+1
      A(IK,J)=AAA(NX)
 53   CONTINUE
 55   CONTINUE
      MM=0
      DO 61 LK=1,N
      M=1
      MM=MM+1
      DO 59 IJ=1,NP
      M=M+1
      TTX=0.
      DO 57 J=1,NP
      TX=Z(LK,J)*A(J,IJ)
      TTX=TTX+TX
 57   CONTINUE
      ZZ(MM,M)=TTX
 59   CONTINUE
 61   CONTINUE
      S=FLOAT(N)
      HS1=S/2.
      HS2=INT(HS1)
      IF(HS1.NE.HS2)GOTO 63
      NNE=HS2
      NNP=HS2
      GOTO 65
 63   NNE=HS2+1
      NNP=HS2
C     * FIND FIRST 2 POINTS IN EST DATA *
 65   AMAX=0.
      DO 77 LK=1,N1
      II=LK+1
      DO 75 J=II,N
      DO 69 K=2,NP1
      AA(LK,J,K)=(ZZ(LK,K)-ZZ(J,K))**2
      IF(K.GT.2)GOTO 67
      D(LK,J)=0.
 67   D(LK,J)=D(LK,J)+AA(LK,J,K)
 69   CONTINUE
      IF(AMAX-D(LK,J))71,71,75
 71   AMAX=D(LK,J)
      DO 73 K=2,NP1
      X1(K)=XX(LK,K)
      X2(K)=XX(J,K)
 73   CONTINUE
      Y1=XX(LK,NP2)
      Y2=XX(J,NP2)
      IDEI=LK
      IDEJ=J
 75   CONTINUE
 77   CONTINUE
C     * FIND FIRST 2 POINTS IN PRE DATA *
      BMAX=0.
      DO 89 LK=1,N1
      IF(LK.EQ.IDEI.OR.LK.EQ.IDEJ)GOTO 89
      II=LK+1
      DO 87 J=II,N
      IF (J.EQ.IDEI.OR.J.EQ.IDEJ)GOTO 87
      DO 81 K=2,NP1
```

```
      BB(LK,J,K)=(ZZ(LK,K)-ZZ(J,K))**2
      IF(K.GT.2)GOTO 79
      D(LK,J)=0.
 79   D(LK,J)=D(LK,J)+BB(LK,J,K)
 81   CONTINUE
      IF(EMAX-D(LK,J))83,83,87
 83   EMAX=D(LK,J)
      DO 85 K=2,NP1
      X3(K)=XX(LK,K)
      X4(K)=XX(J,K)
 85   CONTINUE
      Y3=XX(LK,NP2)
      Y4=XX(J,NP2)
      IDPI=LK
      IDPJ=J
 87   CONTINUE
 89   CONTINUE
      DO 91 K=2,NP1
      EST(1,K)=X1(K)
      EST(2,K)=X2(K)
      FRE(1,K)=X3(K)
      FRE(2,K)=X4(K)
 91   CONTINUE
      YE(1)=Y1
      IDE(1)=IDEI
      YE(2)=Y2
      IDE(2)=IDEJ
      YP(1)=Y3
      IDP(1)=IDPI
      YP(2)=Y4
      IDP(2)=IDPJ
      L=1
      LL=2
      NNN=N-2
      KK=N-3
 93   MM=0
      M=0
      L=L+1
      LL=LL+1
      NNN=NNN-2
      KK=KK-2
C     * FIND NEXT POINT IN EST DATA *
      DO 113 LK=1,N
      DO 95 J=1,L
      IF(LK.EQ.IDE(J))GOTO 113
 95   CONTINUE
      DO 97 J=1,L
      IF(LK.EQ.IDP(J))GOTO 113
 97   CONTINUE
      IF(NNE.EQ.NNP)GOTO 99
      IF(LL.EQ.NNE)GOTO 153
 99   M=M+1
      IE(M)=LK
      DO 103 K=2,NP1
      IF(K.NE.2)GOTO 101
      D1(M)=0.
      D2(M)=0.
 101  DD1(K)=(X1(K)-ZZ(LK,K))**2
      D1(M)=D1(M)+DD1(K)
      DD2(K)=(X2(K)-ZZ(LK,K))**2
      D2(M)=D2(M)+DD2(K)
 103  CONTINUE
      IF(D1(M)-D2(M))109,109,105
 105  DMINE(M)=D2(M)
      DO 107 K=2,NP1
      ES(M,K)=XX(LK,K)
 107  CONTINUE
      ESY(M)=XX(LK,NP2)
      GOTO 113
```

```
109   DMINE(M)=D1(M)
      DO 111 K=2,NP1
      ES(M,K)=XX(LK,K)
111   CONTINUE
      ESY(M)=XX(LK,NP2)
113   CONTINUE
      DMAXE=DMINE(1)
      DO 119 M=1,NNN
      IF(DMAXE-DMINE(M))115,115,119
115   DMAXE=DMINE(M)
      DO 117 K=2,NP1
      EST(LL,K)=ES(M,K)
117   CONTINUE
      YE(LL)=ESY(M)
      IDE(LL)=IE(M)
119   CONTINUE
C     * FIND NEXT POINT IN PRE DATA *
      DO 139 LK=1,N
      DO 121 J=1,L
      IF(LK.EQ.IDP(J))GOTO 139
121   CONTINUE
      DO 123 J=1,LL
      IF(LK.EQ.IDE(J))GOTO 139
123   CONTINUE
      IF(NNE.NE.NNP)GOTO 125
      IF(LL.EQ.NNP)GOTO 149
125   MM=MM+1
      IP(MM)=LK
      DO 129 K=2,NP1
      IF(K.NE.2)GOTO 127
      D3(MM)=0.
      D4(MM)=0.
127   DD3(K)=(X3(K)-ZZ(LK,K))**2
      D3(MM)=D3(MM)+DD3(K)
      DD4(K)=(X4(K)-ZZ(LK,K))**2
      D4(MM)=D4(MM)+DD4(K)
129   CONTINUE
      IF(D3(MM)-D4(MM))135,135,131
131   DMINP(MM)=D4(MM)
      DO 133 K=2,NP1
      PR(MM,K)=XX(LK,K)
133   CONTINUE
      PRY(MM)=XX(LK,NP2)
      GOTO 139
135   DMINP(MM)=D3(MM)
      DO 137 K=2,NP1
      PR(MM,K)=XX(LK,K)
137   CONTINUE
      PRY(MM)=XX(LK,NP2)
139   CONTINUE
      DMAXP=DMINP(1)
      DO 145 MM=1,KK
      IF(DMAXP-DMINP(MM))141,141,145
141   DMAXP=DMINP(MM)
      DO 143 K=2,NP1
      PRE(LL,K)=PR(MM,K)
143   CONTINUE
      YP(LL)=PRY(MM)
      IDP(LL)=IP(MM)
145   CONTINUE
      IF(NNE.NE.NNP)GOTO 147
      IF(LL.LT.NNP)GOTO 93
      GOTO 149
147   IF(LL.LT.NNE)GOTO 93
      GOTO 153
149   LL=NNP
      DO 151 K=2,NP1
      PRE(LL,K)=XX(LK,K)
```

```
151   CONTINUE
      YP(LL)=XX(LK,NP2)
      IDP(LL)=LK
      GOTO 157
153   LL=NNE
      DO 155 K=2,NP1
      EST(LL,K)=XX(LK,K)
155   CONTINUE
      YE(LL)=XX(LK,NP2)
      IDE(LL)=LK
C     #=========================================================#
C     #          FIND COEFFICIENT FROM EST & PRE DATA           #
C     #=========================================================#
157   DO 161 IK=2,NP1
      DO 159 J=1,NNE
      XX(J,IK)=EST(J,IK)
      XX(J,1)=1.0
      XX(J,NP2)=YE(J)
159   CONTINUE
161   CONTINUE
      CALL OLS(B,XX,NNE,NP1,NP2)
      DO 163 IK=1,NP1
      BD1(IK)=B(IK)
163   CONTINUE
      DO 167 IK=2,NP1
      DO 165 J=1,NNP
      XX(J,IK)=PRE(J,IK)
      XX(J,1)=1.0
      XX(J,NP2)=YP(J)
165   CONTINUE
167   CONTINUE
      CALL OLS(B,XX,NNP,NP1,NP2)
      DO 169 IK=1,NP1
      BD2(IK)=B(IK)
169   CONTINUE
C     #=========================================================#
C     #          FIND (|X'X(EST)|/|X'X(PRE)|)**(1/P)            #
C     #=========================================================#
      DO 173 J=2,NP1
      DO 171 IK=1,NNE
      X(IK,J)=EST(IK,J)
      X(IK,1)=1.0
171   CONTINUE
173   CONTINUE
      CALL PRIME(X,NNE,NP1,XTX)
      NX=0
      DO 179 IK=1,NP1
      DO 175 J=1,NP1
      NX=NX+1
      AAA(NX)=XTX(J,IK)
175   CONTINUE
179   CONTINUE
      CALL MINV(AAA,NP1,DDD,IJK,MK)
      DET1=DDD
      DO 183 J=2,NP1
      DO 181 IK=1,NNP
      X(IK,J)=PRE(IK,J)
      X(IK,1)=1.0
181   CONTINUE
183   CONTINUE
      CALL PRIME(X,NNP,NP1,XTX)
      NX=0
      DO 187 IK=1,NP1
      DO 185 J=1,NP1
      NX=NX+1
      AAA(NX)=XTX(J,IK)
185   CONTINUE
```

```
187   CONTINUE
      CALL MINV(AAA,NP1,DDD,IJK,MK)
      DET2=DDD
      IF(DET1.EQ.0.OR.DET2.EQ.0.)GOTO 1500
      DET=(DET1/DET2)**(1./NP1)
C     #================================================================#
C     #        TEST IN EQUALITY OF COEFFICIENT BY CHOW TEST            #
C     #================================================================#
      DO 191 J=1,NNE
      T11=0.
      DO 189 I=2,NP1
      T11=T11+(BD1(I)*EST(J,I))
189   CONTINUE
      BXD1(J)=T11
191   CONTINUE
      SUM1=0.
      DO 193 J=1,NNE
      SUM1=SUM1+(YE(J)-(BD1(1)+BXD1(J)))**2
193   CONTINUE
      A1=SUM1
      DO 197 J=1,NNP
      T22=0.
      DO 195 IJ=2,NP1
      T22=T22+(BD2(IJ)*PRE(J,IJ))
195   CONTINUE
      BXD2(J)=T22
197   CONTINUE
      SUM2=0.
      DO 199 J=1,NNP
      SUM2=SUM2+(YP(J)-(BD2(1)+BXD2(J)))**2
199   CONTINUE
      A2=SUM2
      FC=((AT-A1-A2)/NP1)/((A1+A2)/(N-(2*NP1)))
      IF(FC.GT.F001)GOTO 201
      GOTO 203
201   N001=N001+1
203   IF(FC.GT.F005)GOTO 205
      GOTO 207
205   N005=N005+1
207   TDET=TDET+DET
1500  CONTINUE
      ADET=TDET/NT
      SNT=FLOAT(NT)
      AN01=N001/SNT
      AN05=N005/SNT
      WRITE(6,209)N,NP
209   FORMAT(5X,'LOGISTIC DISTRIBUTION',5X,'N =',I3,5X,'NP =',I3//)
      WRITE(6,211)
211   FORMAT(5X,'ALPHA ESTIMATE AT .01',3X,'ALPHA ESTIMATE AT .05',
     *3X,'(IXTX(EST)I/IXTX(PRE)I)**(1/P)')
      WRITE(6,213)AN01,AN05,ADET
213   FORMAT(9X,F8.5,19X,F8.5,16X,F12.5//)
      STOP
      END
C     #===================================#
C     #          RANDOM NUMBER            #
C     #===================================#
      SUBROUTINE RANDOM(IX,IY,RD)
      IY=IX*65539
      IF(IY)10,15,15
10    IY=IY+2147483647+1
15    RD=IY
      RD=RD*.4656613E-9
      IX=IY
      RETURN
      END
```

```
C       #======================================#
C       #          NORMAL DISTRIBUTION         #
C       #======================================#
        SUBROUTINE NORMAL(SMEAN,SIGMA,X)
        COMMON IX
        A=0.
        DO 15 J=1,12
        CALL RANDOM(IX,IY,RAN)
        A=A+RAN
 15     CONTINUE
        X=(A-6.)*SIGMA+SMEAN
        RETURN
        END
C       #======================================#
C       #         LOGISTIC DISTRIBUTION        #
C       #======================================#
        SUBROUTINE LOGIST(ALPHA,BETA,X1)
        COMMON IY
 10     CALL RANDOM(IX,IY,RAN)
        IF(RAN.EQ.0)GOTO 10
        S=ALOG(RAN)-ALOG(1.-RAN)
        X1=ALPHA+S*BETA
        RETURN
        END
C       #======================================#
C       #            OLS REGRESSION            #
C       #======================================#
        SUBROUTINE OLS(B,X,N,NP1,NP2)
        DIMENSION B(12),X(100,12),A(12,12),S(12,12)
        DO 20 I=1,NP2
        DO 20 K=1,NP2
        SIK=0.0
        DO 10 J=1,N
 10     SIK=SIK+X(J,I)*X(J,K)
        S(I,K)=SIK
 20     S(K,I)=SIK
        DO 40 I=1,NP1
        DO 40 J=1,NP1
 40     A(I,J)=S(I,J)
        CALL INVS(NP1,A)
        DO 50 I=1,NP1
        B(I)=0.
        DO 50 J=1,NP1
 50     B(I)=B(I)+A(J,I)*S(NP2,J)
        RETURN
        END
C       #======================================#
C       #            INVERSE MATRIX            #
C       #======================================#
        SUBROUTINE INVS(NP1,A)
        DIMENSION A(12,12)
        DO 20 K=1,NP1
        A(K,K)=-1./A(K,K)
        DO 5 I=1,NP1
        IF(I-K)3,5,3
 3      A(I,K)=-A(I,K)*A(K,K)
 5      CONTINUE
        DO 10 I=1,NP1
        DO 10 J=1,NP1
        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,NP1
        IF(J-K)18,20,18
 18     A(K,J)=-A(K,J)*A(K,K)
 20     CONTINUE
        DO 25 I=1,NP1
```

```
      DO 25 J=1,NP1
25    A(I,J)=-A(I,J)
      RETURN
      END
C     #=======================================#
C     #              DETERMINANT              #
C     #=======================================#
      SUBROUTINE MINV(AAA,N,DDD,IJK,MK)
      DIMENSION AAA(1),IJK(1),MK(1)
      DDD = 1.0
      NK = -N
      DO 80 K=1,N
      NK = NK + N
      IJK(K) = K
      MK(K) = K
      KK = NK + K
      BIGA = AAA(KK)
      DO 20 J = K,N
      IZ = N * (J-1)
      DO 20 I = K,N
      IJ = IZ + I
10    IF( ABS(BIGA)- ABS(AAA(IJ))) 15,20,20
15    BIGA = AAA(IJ)
      IJK(K) = I
      MK(K) = J
20    CONTINUE
      J = IJK(K)
      IF (J-K) 35,35,25
25    KI = K-N
      DO 30 I =1,N
      KI = KI + N
      HOLD = -AAA(KI)
      JI = KI - K + J
      AAA(KI) = AAA(JI)
30    AAA(JI) = HOLD
35    I = MK(K)
      IF ( I-K) 45,45,38
38    JP = N * (I-1)
      DO 40 J =1,N
      JK = NK + J
      JI = JP + J
      HOLD = - AAA(JK)
      AAA(JK) = AAA(JI)
40    AAA(JI) = HOLD
45    IF (BIGA) 48,46,48
46    DDD = 0.0
      RETURN
48    DO 55 I=1,N
      IF (I-K) 50,55,50
50    IK = NK + I
      PPP = AAA(IK)  / (-1 * BIGA)
      AAA(IK) = PPP
55    CONTINUE
      DO 65 I=1,N
      IK = NK + I
      HOLD = AAA(IK)
      IJ = I - N
      DO 65 J =1,N
      IJ = IJ + N
      IF (I-K) 60,65,60
60    IF (J-K) 62,65,62
62    KJ = IJ -I + K
      AAA(IJ) = HOLD * AAA(KJ) +AAA(IJ)
65    CONTINUE
      KJ = K - N
      DO 75 J=1,N
      KJ =  KJ + N
```

```
      IF(J-K) 70,75,70
70    AAA(KJ) = AAA(KJ) / BIGA
75    CONTINUE
      DDD = DDD * BIGA
      AAA(KK) = 1.0 / BIGA
80    CONTINUE
      K = N
100   K = (K-1)
      IF (K) 150,150,105
105   I = IJK(K)
      IF (I-K) 120,120,108
108   JQ = N * (K-1)
      JR = N * (I-1)
      DO 110 J=1,N
      JK = JQ + J
      HOLD = AAA(JK)
      JI = JR + J
      AAA(JK) = -AAA(JI)
110   AAA(JI) = HOLD
120   J = MK(K)
      IF(J-K) 100,100,125
125   KI = K - N
      DO 130 I=1,N
      KI =KI + N
      HOLD = AAA(KI)
      JI = KI - K + J
      AAA(KI) = - AAA(JI)
130   AAA(JI) = HOLD
      GO TO 100
150   RETURN
      END
C     #=======================================#
C     #              TRANSPOSE                #
C     #=======================================#
      SUBROUTINE PRIME(X,N,NP,XTX)
      DIMENSION X(100,12),XTX(12,12)
      M=0
      DO 1 J=1,NP
      M=M+1
      MM=0
      II=0
 3    II=II+1
      TTX=0.
      DO 2 IK=1,N
      TX=(X(IK,J))*(X(IK,II))
      TTX=TTX+TX
 2    CONTINUE
      MM=MM+1
      XTX(M,MM)=TTX
      IF(MM.LT.NP)GOTO 3
 1    CONTINUE
      RETURN
      END
C     #=======================================#
C     #              ORTHOGONAL               #
C     #=======================================#
      SUBROUTINE ORTHO(XTX,NP,T)
      DIMENSION XTX(12,12),T(12,12)
      T(1,1)=SQRT(XTX(1,1))
      DO 10 J=2,NP
      T(1,J)=XTX(1,J)/T(1,1)
      JJ=J
      J1=J-1
20    T(JJ,J1)=0.
      JJ=JJ+1
      IF(JJ.LE.NP)GOTO 20
10    CONTINUE
```

```
      DO 30 I=2,NP
      TOT=0.
      I1=I-1
      DO 40 K=1,I1
      TT=T(K,I)**2
      TOT=TOT+TT
40    CONTINUE
      TAT=XTX(I,I)-TOT
      T(I,I)=SQRT(TAT)
      II=I
60    TOT=0.
      DO 50 K=1,I1
      TT=T(K,I)*T(K,II)
      TOT=TOT+TT
50    CONTINUE
      TAT=(XTX(I,II)-TOT)/T(I,I)
      T(I,II)=TAT
      II=II+1
      IF(II.LE.NP)GOTO 60
30    CONTINUE
      RETURN
      END
```

ประวัติผู้เขียน

น.ส. พรรณวดี โงวหลินตนา เกิดที่จังหวัดสระบุรี ได้รับปริญญาเศรษฐศาสตร์บัณฑิต จากมหาวิทยาลัยธรรมศาสตร์ เมื่อปีการศึกษา 2525 และเข้าศึกษาต่อในสาขาวิชาสถิติ ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2526