รายการอ้างอิง

ภาษาไทย

จิรายุส พู่มนตรี. การเปรียบเทียบตัวประมาณริดจ์สำหรับการวิเคราะห์ความถดถอยแบบริดจ์. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2534.

จุฬารัตน์ ตันประเสริฐ และชิติพงศ์ ตันประเสริฐ. "โครงข่ายประสาทเทียม (Artificial Neural Networks) อีกแนวทางหนึ่งของการแก้ปัญหาด้วยคอมพิวเตอร์ (ตอนที่ 1)". สาร NECTEC (กรกฎาคม-กันยายน 2538) : 19-23.

จุฬารัตน์ ตันประเสริฐ และชิติพงศ์ ตันประเสริฐ. "โครงข่ายประสาทเทียม (Artificial Neural Networks) อีกแนวทางหนึ่งของการแก้ปัญหาด้วยคอมพิวเตอร์ (ตอนที่ 2)". สาร NECTEC (ตุลาคม-ธันวาคม 2538) : 19-23.

จุฬารัตน์ ตันประเสริฐ และชิติพงศ์ ตันประเสริฐ. "โครงข่ายประสาทเทียม (Artificial Neural Networks) อีกแนวทางหนึ่งของการแก้ปัญหาด้วยคอมพิวเตอร์ (ตอนที่ 3)". สาร NECTEC (มกราคม-กุมภาพันธ์ 2539) : 13-17.

เจษฎาพร อุทธนวิบูลย์ชัย. การศึกษาเปรียบเทียบตัวประมาณริดจ์. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2533.

ดวงพร ชูรักษ์. การเปรียบเทียบการประมาณค่าในการวิเคราะห์ความถดถอยพหุคูณ โดยวิธีริดจ์ รีเกรสชัน วิเกรสชันคอมพริ้นซิเปิ้ลคอมโพเนนท์ และวิธีกำลังสองน้อยสุด ในกรณีที่เกิดพหุสัมพันธ์ระหว่างตัวแปรอิสระ. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2539.

ธันยากร ด้นชลขันธ์. การเปรียบเทียบการประมาณค่าสัมประสิทธิ์การถดถอยพหุโดยวิธีกำลังสองน้อยที่สุด วิธีริดจ์รีเกรสชัน และวิธีที่ใช้หลักการของริดจ์และสไตน์ ในกรณีที่เกิดพหุสัมพันธ์ระหว่างตัวแปรอิสระ. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2538.

ธีระพร วีระถาวร. การอนุมานเชิงสถิติขั้นกลาง : โครงสร้างและความหมาย. พิมพ์ครั้งที่ 2. กรุงเทพมหานคร : สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2536.

ภาษาต่างอังกฤษ

Chatterjee, S. and Price, B. Regression analysis by example. New York : John Wiely & Sons, 1939.

Fausett, L. Fundamentals of Neural Networks. New Jersey : Prentice-Hall, Inc., 1994.

Gorr, L. W., Nagin, D. and Szcypula, J. "Comparative study of artificial neural network and statistical models for predicting student grade point averages". International Journal of forecasting. 10 (1994) : 17-34.

Hoerl, A.E. and Kennard, R.W. "Ridge Regression : Biased Estimation for Nonorthogonal Problems". Technometrice. 12 (1970a) : 55-67.

Kuter, H.M. Applied linear statistical models. 3 rd ed. Illinois : Richard D. Iwrin, Inc., 1990.

Smith, M. Neural networks for statistic modeling. New York : Van Nostrand Reinhold, 1993.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายละเอียดของโปรแกรมที่ใช้มีดังนี้

อันดับที่	ชื่อโปรแกรม	คุณสมบัติโปรแกรม	ชื่อโปรแกรมที่เรียกใช้	หน้าที่
Main Program	-	1. รับค่าและกำหนดค่าพารามิเตอร์ในสถานการณ์ต่างๆ 2. คำนวณค่า AMSE ที่ได้จากวิธีริดจ์รีเกรสชัน	GENX, STNDRDX, CALXTX, EIGEN, NORMAL, SCALE, LOGNOR, CALY, BOXCOX, STNDRDY, WRTDT, YSD, CALK, CBTHG, CMSE	182 - 185
Subroutine				
1	GENX	สร้างเมทริกซ์ตัวแปรอิสระ X ที่สัมพันธ์กัน	NORMAL	185 - 186
2	STNDRDX	แปลงเมทริกซ์ให้อยู่ในรูปแบบมาตรฐาน		186 - 187
3	STNDRDY	แปลงเวกเตอร์ $\underset{\sim}{y}$ ให้อยู่ในรูปแบบมาตรฐาน		187 - 188
4	WRTDT	เขียนเมทริกซ์ X และเวกเตอร์ $\underset{\sim}{y}$ ที่อยู่ในรูปมาตรฐานลงแฟ้มข้อมูล		188 - 189
5	CALXTX	คำนวณหาค่า $X'X$		189 - 190
6	EIGEN	รับและส่งค่าเฉพาะและเวกเตอร์เฉพาะ	JACOBI	190 - 191
7	JACOBI	คำนวณค่าเฉพาะและเวกเตอร์เฉพาะโดยใช้วิธี JACOBI		191 - 195
8	CALY	คำนวณหาค่าเวกเตอร์ตัวแปรตาม $\underset{\sim}{y}$		195
9	BOXCOX	แปลงค่า $\underset{\sim}{y}$ โดยใช้การยกกำลังของ Box และ Cox เมื่อ $\underset{\sim}{\varepsilon}$ มีการแจกแจงออกนอร์มอล	OLS	196 - 198

ลำดับที่	ชื่อโปรแกรม	คุณสมบัติโปรแกรม	ชื่อโปรแกรมที่เรียกใช้	หน้าที่
10	OLS	คำนวณ $\hat{\beta}$ โดยใช้วิธีกำลังสองน้อยสุด	BXTY, INVRS, BMSE	198
11	CALK	คำนวณหาค่าพารามิเตอร์ k ที่เหมาะสมโดยวิธีค้นหาแบบลำดับ	MSEB	199 - 200
12	YSD	คำนวณค่าส่วนเบี่ยงเบนมาตรฐานของ y ชุดที่ 1 ถึง (n-20)		201
13	BXTY	คำนวณหาค่า $X'y$		201 - 202
14	CXTY	คำนวณหาค่า $X'y$ โดยใช้ข้อมูลชุดที่ 1 ถึง (n-20)		202
15	CBTHG	คำนวณหาค่า $\hat{\beta}_R$ โดยใช้ข้อมูลชุดที่ 1 ถึง (n-20)		202 - 203
16	BMSE	คำนวณหา MSE ในการพยากรณ์ $\hat{y}$ ที่ได้จากวิธีกำลังสองน้อยสุด		204
17	CMSE	คำนวณหาค่า MSE ในการพยากรณ์ $\hat{y}_R$ ที่ได้จากวิธีริดจ์รีเกรสชัน		204 - 205
18	INVRS	คำนวณหาเมทริกซ์ผกผัน		205 - 206
Function				
1	MSEB	คำนวณหาค่า MSE ในการประมาณ $\hat{\beta}_R$ เพื่อหาค่า k ที่เหมาะสม		200 - 201
2	NORMAL	สร้างเลขสุ่มที่มีการแจกแจงปกติ	RAND	206
3	SCALE	สร้างเลขสุ่มที่มีการแจกแจงปกติปลอมปน	NORMAL	206 - 207
4	LOGNOR	สร้างเลขสุ่มที่มีการแจกแจงลอกนอร์มอล	NORMAL	207
5	RAND	สร้างเลขสุ่มที่มีการแจกแจง U(0,1)		207

```
**********************************************************************************
* A COMPARISON ON FORCASTING METHODS BETWEEN RIDGE REGRESSION   *
* AND ARTIFICIAL NEURAL NETWORK METHODS IN MULTIPLE REGRESSION  *
* ANALYSIS WITH MULTICOLLINEARITY.                              *
**********************************************************************************
************************************************
*                    MAIN PROGRAM              *
************************************************
      CHARACTER NAME*5
      INTEGER DIST,COUNT,B,FAIL
      REAL DMEAN,SIGMA,C,P,NORMAL,LOGNOR,SCALE,KOPT,
     *    X(120,5)
      DOUBLE PRECISION MSE1,MSE2,AMSE,XTX(5,5),EGNVL(5),
     *              VECMAX(5),VECMIN(5),Y(120),XTY(5),
     *              BTHG(5),E(120),BINT(5),YL(120),
     *              YSTND(120),XSTND(120,5),MSEY,MSE,
     *              SMSE,SQMSE,BMSE,SDMSE,EGNVL2,
     *              VECMAX2,VECMIN2,BINT2
      COMMON /SEED/IX,IT
     *      /DIM/N,M
     *      /CORR/CORR1,CORR2
      IX = 13
      IT = 0
      WRITE(*,*)'KEY NO_DIST COUNT N P'
      READ(5,*)DIST,COUNT,N,M
      IF (M.EQ.3) THEN
         WRITE(*,1)' CORRELATION 1 ='
         READ(5,*)CORR1
      ELSE IF (M.EQ.5) THEN
         WRITE(*,1)' CORRELATION 1 AND 2 ='
         READ(5,*)CORR1,CORR2
```

```
      END IF
      IF (DIST.EQ.1) THEN
         WRITE(*,*)'1 KEY DMEAN SIGMA'
         READ(5,*)DMEAN,SIGMA
      ELSE IF (DIST.EQ.2) THEN
         WRITE(*,*)'2 KEY C P DMEAN SIGMA'
         READ(5,*)C,P,DMEAN,SIGMA
      ELSE IF (DIST.EQ.3) THEN
         WRITE(*,*)'3 KEY DMEAN SIGMA'
         READ(5,*)DMEAN,SIGMA
      END IF
      WRITE(*,1)' PLEASE TYPE DATA FILE NAME '
      READ(*,3)NAME
1     FORMAT(A\)
3     FORMAT(A)
      DO 5 LOOP = 1,COUNT
7        CALL GENX(X)
         CALL STNDRDX(X,XSTND)
         CALL CALXTX(XSTND,XTX)
         CALL EIGEN(XTX,EGNVL,VECMAX,VECMIN,EGNVL2,VECMAX2,VECMIN2)
         DO 10 B = 1,2
            IF (B.EQ.1) THEN
               DO 15 I = 1,M
                  BINT(I) = VECMAX(I)
                  BINT2(I) = VECMAX2(I)
15             CONTINUE
            ELSE IF (B.EQ.2) THEN
               DO 20 I = 1,M
                  BINT(I) = VECMIN(I)
                  BINT2(I) = VECMIN2(I)
20             CONTINUE
```

```
      END IF
      IF (DIST.EQ.1) THEN
        DO 25 I = 1,N
          E(I) = NORMAL(DMEAN,SIGMA)
25      CONTINUE
      ELSE IF (DIST.EQ.2) THEN
        DO 30 I = 1,N
          E(I) = SCALE(C,P,DMEAN,SIGMA)
30      CONTINUE
      ELSE IF (DIST.EQ.3) THEN
        DO 35 I = 1,N
          E(I) = LOGNOR(DMEAN,SIGMA)
35      CONTINUE
      END IF
      CALL CALY(X,BINT,E,Y)
        IF (DIST.EQ.3) THEN
          FAIL = 0
          CALL BOXCOX(Y,X,YL,FAIL)
          IF (FAIL.GT.0) THEN
            GOTO 7
          END IF
          DO 40 I = 1,N
            Y(I) = YL(I)
40        CONTINUE
        END IF
      CALL STNDRDY(Y,YSTND)
      CALL WRTDT(YSTND,XSTND,NAME,LOOP,B)
      CALL YSD(YSTND,MSEY)
      CALL CALK(EGNVL2,BINT2,MSEY,KOPT)
      CALL CXTY(XSTND,YSTND,XTY)
      CALL CBTHG(XSTND,KOPT,XTY,BTHG)
```

```
            CALL CMSE(YSTND,BTHG,XSTND,MSE)
              IF (B.EQ.1) THEN
                MSE1 = MSE
              ELSE IF (B.EQ.2) THEN
                MSE2 = MSE
              END IF
10        CONTINUE
          AMSE = (MSE1+MSE2)/2
          SMSE = SMSE+AMSE
          SQMSE = SQMSE+(AMSE**2)
          BMSE = SMSE/LOOP
          WRITE(*,*)LOOP,' AVERAGE ALL LOOP MSE = ',BMSE
          IF (LOOP.EQ.400) THEN
            SDMSE = SQRT((SQMSE-LOOP*(BMSE**2))/(LOOP-1))
            WRITE(*,*)LOOP,' SD MSE = ',SDMSE
          ENDIF
5       CONTINUE
        STOP
        END
*****************************************************
*  GENERATE X(I,J) HAVE MULTICOLLINEARITY  *
*****************************************************
        SUBROUTINE GENX(X)
        REAL Z(120,6),X(120,5),NORMAL,SUMZ(6),ZBAR(6),ZSD(6),SUMSQ(6)
        COMMON /SEED/IX,IT
      *       /DIM/N,M
      *       /CORR/CORR1,CORR2
        DO 5 J = 1,M+1
          DO 10 I = 1,N
            Z(I,J) = 0.0
            Z(I,J) = NORMAL(0.0,1.0)
```

```
10      CONTINUE
5     CONTINUE
      DO 7 J = 1,M+1
        SUMZ(J) = 0.0
        SUMSQ(J) = 0.0
        DO 9 I = 1,N
          SUMZ(J) = SUMZ(J)+Z(I,J)
          SUMSQ(J) = SUMSQ(J)+Z(I,J)**2
9       CONTINUE
        ZBAR(J) = SUMZ(J)/N
        ZSD(J) = SQRT((SUMSQ(J)-N*(ZBAR(J)**2))/(N-1))
7     CONTINUE
      DO 15 I = 1,N
        DO 20 J = 1,M
          IF (J.GE.1.AND.J.LE.3) THEN
            X(I,J) = (SQRT(1-CORR1)*Z(I,J))+(SQRT(CORR1)*Z(I,M+1))
          ELSE IF (J.GE.4) THEN
            X(I,J) = (SQRT(1-CORR2)*Z(I,J))+(SQRT(CORR2)*Z(I,M+1))
          END IF
20      CONTINUE
15    CONTINUE
      RETURN
      END
*****************************************************
*             STRANDRADIZE MATRIX X                 *
*****************************************************
      SUBROUTINE STNDRDX(X,XSTND)
      REAL X(120,5),SUMXSQ(5),XBAR(5),SUMX(5),SIGX(5)
      DOUBLE PRECISION XSTND(120,5)
      COMMON /DIM/N,M
      DO 10 J = 1,M
```

```
      SUMX(J) = 0.0
      SUMXSQ(J) = 0.0
      DO 20 I = 1,N
        SUMX(J) = SUMX(J) + X(I,J)
        SUMXSQ(J) = SUMXSQ(J) + X(I,J)**2
20    CONTINUE
10  CONTINUE
    DO 30 J = 1,M
      XBAR(J) = SUMX(J)/N
      SIGX(J) = SQRT((SUMXSQ(J)-N*(XBAR(J)**2)))
30  CONTINUE
    DO 40 J = 1,M
      DO 50 I = 1,N
        XSTND(I,J) = (X(I,J)-XBAR(J))/SIGX(J)
50    CONTINUE
40  CONTINUE
    RETURN
    END
*****************************************************
*              STRANDRADIZE MATRIX Y                *
*****************************************************
    SUBROUTINE STNDRDY(Y,YSTND)
    DOUBLE PRECISION Y(120),YSTND(120),
   *  SUMYSQ,YBAR,SUMY,SIGY
    COMMON /DIM/N,M
    SUMY = 0.0
    SUMYSQ = 0.0
    DO 10 I = 1,N
      SUMY = SUMY+Y(I)
      SUMYSQ = SUMYSQ+Y(I)**2
10  CONTINUE
```

```
      YBAR = SUMY/N
      SIGY = SQRT((SUMYSQ-N*(YBAR**2)))
      DO 20 I = 1,N
         YSTND(I) = (Y(I)-YBAR)/SIGY
20    CONTINUE
      RETURN
      END
*******************************************************
*            WRITE DATA X AND Y TO FILE               *
*******************************************************
      SUBROUTINE WRTDT(Y,X,NAME,L,B)
      DOUBLE PRECISION Y(120),X(120,5)
      INTEGER B
      CHARACTER NAME*5,NO(1000)*3,S*1,FILENAME*20
      COMMON /DIM/N,M
      OPEN(1,FILE = 'C:\FORTRAN\FIL_NO.DAT')
      READ(1,3)(NO(I),I=1,1000)
3     FORMAT(10(A,1X))
      CLOSE(1)
      IF (B.EQ.1) THEN
         S = '1'
      ELSE IF (B.EQ.2) THEN
         S = '2'
      END IF
      FILENAME = 'C:\DATA\'//NAME//NO(L)//'.T'//S
      OPEN(2,FILE = FILENAME)
      WRITE(2,6)
      WRITE(2,*)' '
      WRITE(2,7)M
      WRITE(2,8)
      WRITE(2,9)N
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
      WRITE(2,10)
      WRITE(2,11)
6     FORMAT('# TRAJAN V1.0 PATTERNS')
7     FORMAT('NO_INPUTS ',I1)
8     FORMAT('NO_OUTPUTS 1')
9     FORMAT('NO_PATTERNS ',I3)
10    FORMAT('NO_VERIFICATION 20')
11    FORMAT('ATTRIBUTES')
      DO 12 I = 1,M
         WRITE(2,13)I
13       FORMAT('X',I1)
12    CONTINUE
      WRITE(2,14)
14    FORMAT('Y')
      DO 15 I = 1,N
         WRITE(2,*)(X(I,J),J=1,M),Y(I)
15    CONTINUE
      CLOSE(2)
      RETURN
      END
*****************************************************
*              BUILDING MATRIX XTX                  *
*****************************************************
      SUBROUTINE CALXTX(X,XTX)
      DOUBLE PRECISION XTX(5,5),X(120,5),XT(5,120)
      COMMON /DIM/N,M
      DO 10 J = 1,M
         DO 10 I = 1,N
            XT(J,I) = X(I,J)
10    CONTINUE
      DO 20 I = 1,M
```

```
      DO 30 J = 1,M
        XTX(I,J) = 0.0
        DO 40 K = 1,N
          XTX(I,J) = XTX(I,J)+XT(I,K)*X(K,J)
40      CONTINUE
30    CONTINUE
20  CONTINUE
    RETURN
    END
*****************************************************
*       EIGENVALUE AND EIGENVECTOR                  *
*****************************************************
    SUBROUTINE EIGEN(A,XST,EGNVL,VECMAX,VECMIN,EGNVL2,
   *                 VECMAX2,VECMIN2)
    DOUBLE PRECISION A(5,5),X(5,5),EGNVL(5),EGNVL2(5)
   *          VECMAX(5),VECMIN(5),Q(5,5)XST(120,5),Q2(5,5),X2(5,5),
   *          VECMAX2(5),VECMIN2(5),XT(5,120)
    COMMON /DIM/N,M
    DO 5 I = 1,N
      DO 5 J = 1,M
        Q(I,J) = A(I,J)
5   CONTINUE
    CALL JACOBI(Q,1,X)
    DO 10 I = 1,M
      EGNVL(I) = Q(I,I)
      VECMAX(I) = X(I,1)
      VECMIN(I) = X(I,M)
10  CONTINUE
    DO 15 J = 1,M
      DO 15 I = 1,(N-20)
        XT(J,I) = XST(I,J)
```

```
15    CONTINUE
      DO 20 I = 1,M
         DO 30 J = 1,M
            Q2(I,J) = 0.0
            DO 40 K = 1,(N-20)
               Q2(I,J) = Q2(I,J) + XT(I,K) * XST(K,J)
40          CONTINUE
30       CONTINUE
20    CONTINUE
      CALL JACOBI(Q2,1,X2)
      DO 50 I =1,M
         EGNVL2(I) = Q2(I,I)
         VECMAX2(I) = X2(I,1)
         VECMIN2(I) = X2(I,M)
50    CONTINUE
      RETURN
      END
*****************************************************
*                    JACOBI METHOD                  *
*****************************************************
      SUBROUTINE JACOBI(Q,JVEC,V)
      INTEGER IN
      REAL EPSI
      DOUBLE PRECISION Q(5,5),V(5,5),IH(5),X(5)
      COMMON /DIM/N,M
      IF (JVEC) 10,15,10
10    DO 14 I = 1,M
         DO 14 J = 1,M
            IF (I-J) 12,11,12
11          V(I,J) = 1.0
            GOTO 14
```

```
12          V(I,J) = 0.0
14      CONTINUE
15      IN = 0
17      MI = M-1
        DO 30 I = 1,MI
          X(I) = 0.0
          MJ = I+1
          DO 30 J = MJ,M
            IF (X(I)-DABS(Q(I,J))) 20,20,30
20          X(I) = DABS(Q(I,J))
            IH(I) = J
30        CONTINUE
40        DO 70 I = 1,MI
            IF (I-1) 60,60,45
45          IF (XMAX-X(I)) 60,70,70
60          XMAX = X(I)
            IP = I
            JP = IH(I)
70        CONTINUE
          EPSI = 0.00000001
          IF (XMAX-EPSI) 1000,1000,148
148       IN = IN+1
          IF (Q(IP,IP)-Q(JP,JP)) 150,151,151
150       TANG = -2.0*Q(IP,JP)/(DABS(Q(IP,IP)-Q(JP,JP))
     *                +DSQRT((Q(IP,IP)-Q(JP,JP))**2+4.0*Q(IP,JP)**2))
          GOTO 160
151       TANG = +2.0*Q(IP,JP)/(DABS(Q(IP,IP)-Q(JP,JP))
     *                +DSQRT((Q(IP,IP)-Q(JP,JP))**2+4.0*Q(IP,JP)**2))
160       COSN = 1.0/SQRT(1.0+TANG**2)
          SINE = TANG*COSN
          QII = Q(IP,IP)
```

```
      Q(IP,IP) = COSN**2*(QII+TANG*(2.0*Q(IP,JP)+TANG*Q(JP,JP)))
      Q(JP,JP) = COSN**2*(Q(JP,JP)-TANG*(2.0*Q(IP,JP)-TANG*QII))
      Q(IP,JP) = 0.0
      IF (Q(IP,IP)-Q(JP,JP)) 152,153,153
152   TEMP = Q(IP,IP)
      Q(IP,IP) = Q(JP,JP)
      Q(JP,JP) = TEMP
      IF (SINE) 154,155,155
154   TEMP = +COSN
      GOTO 170
155   TEMP = -COSN
170   COSN =ABS(SINE)
      SINE = TEMP
153   DO 350 I = 1,MI
        IF (I-IP) 210,350,200
200     IF (I-JP) 210,350,210
210     IF (IH(I)-IP) 230,240,230
230     IF (IH(I)-JP) 350,240,350
240     K = IH(I)
250     TEMP = Q(I,K)
        Q(I,K) = 0.0
        MJ = I+1
        X(I) = 0.0
        DO 320 J = MJ,M
          IF (X(I)-DABS(Q(I,J))) 300,300,320
300       X(I) = DABS(Q(I,J))
          IH(I) = J
320     CONTINUE
        Q(I,K) = TEMP
350   CONTINUE
      X(IP) = 0.0
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
      X(JP) = 0.0
      DO 530 I = 1,M
         IF (I-IP) 370,530,420
370      TEMP = Q(I,IP)
         Q(I,IP) = COSN*TEMP+SINE*Q(I,JP)
         IF (X(I)-DABS(Q(I,IP))) 380,390,390
380      X(I) = DABS(Q(I,IP))
         IH(I) = IP
390      Q(I,JP) = -SINE*TEMP+COSN*Q(I,JP)
         IF (X(I)-DABS(Q(I,JP))) 400,530,530
400      X(I) = DABS(Q(I,JP))
         IH(I) = JP
         GOTO 530
420      IF (I-JP) 430,530,480
430      TEMP = Q(IP,I)
         Q(IP,I) = COSN*TEMP+SINE*Q(I,JP)
         IF (X(IP)-DABS(Q(IP,I))) 440,450,450
440      X(IP) = DABS(Q(IP,I))
         IH(IP) = I
450      Q(I,JP) = -SINE*TEMP+COSN*Q(I,JP)
         IF (X(I)-DABS(Q(I,JP))) 400,530,530
480      TEMP = Q(IP,I)
         Q(IP,I) = COSN*TEMP+SINE*Q(JP,I)
         IF (X(IP)-DABS(Q(IP,I))) 490,500,500
490      X(IP) = DABS(Q(IP,I))
         IH(IP) = I
500      Q(JP,I) = -SINE*TEMP+COSN*Q(JP,I)
         IF (X(JP)-DABS(Q(JP,I))) 510,530,530
510      X(JP) = DABS(Q(JP,I))
         IH(JP) = I
530   CONTINUE
```

```
      IF (JVEC) 540,40,540
540   DO 550 I=1,M
        TEMP = V(I,IP)
        V(I,IP) = COSN*TEMP+SINE*V(I,JP)
        V(I,JP) = -SINE*TEMP+COSN*V(I,JP)
550   CONTINUE
      GOTO 40
1000  RETURN
      END
*************************************************
*              BUILDING MATRIX Y                *
*************************************************
      SUBROUTINE CALY(X,B,E,Y)
      REAL X(120,5)
      DOUBLE PRECISION B(5),TY(120),Y(120),E(120)
      COMMON /DIM/N,M
      DO 10 I = 1,N
        TY(I) = 0.0
        DO 20 K = 1,M
          TY(I) = TY(I)+X(I,K)*B(K)
20      CONTINUE
        Y(I) = TY(I)+E(I)
10    CONTINUE
      RETURN
      END
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
**************************************************
*       TRANFROM Y BY BOXCOX METHOD        *
**************************************************
      SUBROUTINE BOXCOX(Y,X,YL,F)
      INTEGER F
      REAL LAMB,LAMMIN,LAMMAX,LAMOPT,X(120,5),XT(5,120)
      DOUBLE PRECISION Y(120),YL(120),YSTR(120),XTX(5,5),
     *                 SUM,G,SIG,S(3)
      CHARACTER STATUS*10
      COMMON /DIM/N,M
      SUM = 0.0
      DO 5 I = 1,N
        IF (Y(I).LE.0) THEN
          WRITE(*,*) 'CAN NOT FIND GEOMETRIC MEAN'
          F = F+1
          RETURN
        END IF
5     CONTINUE
      DO 7 J = 1,M
        DO 7 I = 1,N
          XT(J,I) = X(I,J)
7     CONTINUE
      DO 9 I = 1,M
        DO 9 J = 1,M
          XTX(I,J) = 0.0
          DO 9 K = 1,N
            XTX(I,J) = XTX(I,J)+XT(I,K)*X(K,J)
9     CONTINUE
      DO 10 I = 1,N
        SUM = SUM+DLOG(Y(I))
10    CONTINUE
```

```
      G = DEXP(SUM/N)
      DO 15 I = 1,N
         YSTR(I) = Y(I)/G
15    CONTINUE
      LAMMIN = -15.0
      LAMMAX = 16.0
      E = 0.01
20    LAMOPT = (LAMMIN+LAMMAX)/2
      DO 25 J = 1,3
         S(J) = 0.0
         LAMB = LAMOPT+(J-2)*E
         DO 30 I = 1,N
            YL(I) = 0.0
            IF (LAMB.NE.0) THEN
               YL(I) = ((YSTR(I)**LAMB)-1)/LAMB
            ELSE
               YL(I) = DLOG(YSTR(I))
            END IF
30       CONTINUE
         CALL OLS(YL,XTX,SIG,X)
         S(J) = SIG
25    CONTINUE
      IF (S(2).LE.S(1).AND.S(2).LE.S(3)) THEN
         STATUS = 'OPTIMUM'
      ELSE IF (S(1).GT.S(3)) THEN
         LAMMIN = LAMOPT
      ELSE IF (S(1).LT.S(3)) THEN
         LAMMAX = LAMOPT
      END IF
      IF ((LAMMAX-LAMMIN).GT.E.AND.(STATUS.NE.'OPTIMUM')) GOTO 20
      DO 35 I = 1,N
```

```
      IF (LAMOPT.NE.0) THEN
         YL(I) = ((YSTR(I)**LAMOPT)-1)/LAMOPT
      ELSE
         YL(I) = DLOG(YSTR(I))
      END IF
35    CONTINUE
      STATUS = 'RETURN'
      RETURN
      END
*******************************************************
* ORDINARY LEAST SQUARE METHOD FIND SIGMA *
*******************************************************
      SUBROUTINE OLS(Y,XTX,SIG,X)
      REAL X(120,5)
      DOUBLE PRECISION XTX(5,5),Y(120),SIG,XINVL(5,5),
     *                 XTYL(5),BTHGL(5),AMSE
      COMMON /DIM/N,M
      CALL BXTY(X,Y,XTYL)
      CALL INVRS(XTX,XINVL)
      DO 5 I =1,M
         BTHGL(I) = 0.0
         DO 10 J = 1,M
            BTHGL(I) = BTHGL(I)+XINVL(I,J)*XTYL(J)
10       CONTINUE
5     CONTINUE
      CALL BMSE(Y,BTHGL,X,AMSE)
      SIG = AMSE
      RETURN
      END
```

```
**************************************************
*      CALCULATE K FOR RIDGE ESTIMATOR           *
**************************************************
      SUBROUTINE CALK(EGN,B,MSEY,KOPT)
      REAL D,KINT,K01,KOPT,KBF01,K001,KSTP,KAF01,
     *     MKINT,MSEB,MK01,MKBF01,MK001,MKAF01
      DOUBLE PRECISION EGN(5),B(5),MSEY
      COMMON /DIM/N,M
      KOPT = 0.0
      D = 0.01
      KINT = 0.0
5     K01 = KINT + D
      MKINT = MSEB(KINT,EGN,B,MSEY)
      MK01 = MSEB(K01,EGN,B,MSEY)
      IF (MKINT-MK01) 10,15,15
10    K01 = KINT
      KOPT = K01
      GOTO 17
15    KINT =K01
      GOTO 5
17    D = 0.001
      IF (KOPT.EQ.0) GOTO 40
      KBF01 = KOPT - 0.01
20    K001 = KBF01 + D
      IF (K001.GE.KOPT) GOTO 40
      MKBF01 = MSEB(KBF01,EGN,B,MSEY)
      MK001 = MSEB(K001,EGN,B,MSEY)
      IF (MKBF01-MK001) 25,30,30
25    KOPT = KBF01
      GOTO 60
30    KBF01 = K001
```

```
      GOTO 20
40    KSTP = KOPT + 0.01
      K001 = KOPT
43    KAF01 = K001 + D
      IF (KAF01.GE.KSTP) GOTO 60
      MKAF01 = MSEB(KAF01,EGN,B,MSEY)
      MK001 = MSEB(K001,EGN,B,MSEY)
      IF (MKAF01-MK001) 45,45,50
45    K001 = KAF01
      GOTO 43
50    KOPT = K001
60    RETURN
      END
**************************************************
*          FUNCTION MSE OF BETAHAG               *
**************************************************
      REAL FUNCTION MSEB(K,EGN,B,MSEY)
      REAL K,BI
      DOUBLE PRECISION EGN(5),B(5),SUM,COV,BIAS,MSEY
      COMMON /DIM/N,M
      SUM = 0.0
      BI = 0.0
      DO 10 I = 1,M
        SUM = SUM + (EGN(I))/((EGN(I)+K)**2)
10    CONTINUE
      COV = MSEY*SUM
      DO 20 I = 1,M
        BI = BI+(B(I)**2)/((EGN(I)+K)**2)
20    CONTINUE
      BIAS = (K**2)*BI
      MSEB = COV+BIAS
```

```
      RETURN
      END
*****************************************************
*  SUBROUTINE MSE OF DEPENDENT VARIABLE   *
*****************************************************
      SUBROUTINE YSD(Y,MSEY)
      DOUBLE PRECISION Y(120),SUMSQ,SUMY,MSEY,YBAR
      COMMON /DIM/N,M
      SUMSQ = 0.0
      SUMY = 0.0
      DO 10 I = 1,(N-20)
         SUMY = SUMY+Y(I)
         SUMSQ =SUMSQ+(Y(I)**2)
10    CONTINUE
      YBAR = SUMY/(N-20)
      MSEY = (SUMSQ-(N-20)*(YBAR**2))/(N-21)
      RETURN
      END
*****************************************************
*              BUILDING MATRIX XTY                  *
*****************************************************
      SUBROUTINE BXTY(X,Y,XTY)
      REAL X(120,5)
      DOUBLE PRECISION Y(120),XTY(5),XT(5,120)
      COMMON /DIM/N,M
      DO 10 J = 1,M
         DO 10 I = 1,N
            XT(J,I) = X(I,J)
10    CONTINUE
      DO 20 I = 1,M
         XTY(I) = 0.0
```

```
      DO 20 K = 1,N
        XTY(I) = XTY(I)+XT(I,K)*Y(K)
20    CONTINUE
      RETURN
      END
*****************************************************
*          BUILDING MATRIX XTY 1 TO N-20            *
*****************************************************
      SUBROUTINE CXTY(X,Y,XTY)
      DOUBLE PRECISION Y(120),XTY(5),X(120,5),XT(5,120)
      COMMON /DIM/N,M
      DO 10 J = 1,M
        DO 10 I = 1,(N-20)
          XT(J,I) = X(I,J)
10    CONTINUE
      DO 20 I = 1,M
        XTY(I) = 0.0
        DO 20 K = 1,(N-20)
          XTY(I) = XTY(I)+XT(I,K)*Y(K)
20    CONTINUE
      RETURN
      END
*****************************************************
*              SUBROUTINE BETAHAG                   *
*****************************************************
      SUBROUTINE CBTHG(X,KOPT,XTY,B)
      REAL KOPT
      DOUBLE PRECISION XTX(5,5),XT(5,120),X(120,5),XTY(5),B(5),
     *                 XK(5,5),XINV(5,5)
      COMMON /DIM/N,M
      DO 10 J = 1,M
```

```
      DO 10 I = 1,(N-20)
         XT(J,I) = X(I,J)
10    CONTINUE
      DO 20 I = 1,M
         DO 30 J = 1,M
            XTX(I,J) = 0.0
            DO 40 K = 1,(N-20)
               XTX(I,J) = XTX(I,J)+XT(I,K)*X(K,J)
40          CONTINUE
30       CONTINUE
20    CONTINUE
      DO 50 I = 1,M
         DO 50 J = 1,M
            XK(I,J) = 0.0
            IF (I-J) 60,70,60
60          XK(I,J) = XTX(I,J)
            GOTO 50
70          XK(I,J) = XTX(I,J)+KOPT
50    CONTINUE
      CALL INVRS(XK,XINV)
      DO 80 I =1,M
         B(I) = 0.0
         DO 85 K = 1,M
            B(I) = B(I)+XINV(I,K)*XTY(K)
85       CONTINUE
80    CONTINUE
      RETURN
      END
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
*****************************************************
*          MEAN SQURE ERROR (YHAG)                  *
*****************************************************
      SUBROUTINE BMSE(Y,B,X,MSE)
      REAL X(120,5)
      DOUBLE PRECISION Y(120),B(5),YH(120),ESQ,MSE
      COMMON /DIM/N,M
      ESQ = 0.0
      DO 10 I = 1,N
         YH(I) = 0.0
         DO 10 J = 1,M
            YH(I) = YH(I)+X(I,J)*B(J)
10    CONTINUE
      DO 20 I = 1,N
         ESQ = ESQ+(Y(I)-YH(I))**2
20    CONTINUE
      MSE = ESQ/N
      RETURN
      END
*****************************************************
*    MEAN SQURE ERROR (YHAG)  N-19 TO N             *
*****************************************************
      SUBROUTINE CMSE(Y,B,X,MSE)
      DOUBLE PRECISION Y(120),B(5),YH(120),ESQ,X(120,5),MSE
      COMMON /DIM/N,M
      ESQ = 0.0
      DO 10 I = (N-19),N
         YH(I) = 0.0
         DO 10 J = 1,M
            YH(I) = YH(I)+X(I,J)*B(J)
10    CONTINUE
```

```
      DO 20 I = (N-19),N
         ESQ = ESQ+(Y(I)-YH(I))**2
20    CONTINUE
      MSE = ESQ/20
      RETURN
      END
*******************************************************
*       BUILDING  INVERSE MATRIX (XTX+KI)       *
*******************************************************
      SUBROUTINE INVRS(X,XINV)
      DOUBLE PRECISION X(5,5),XINV(5,5),A(5,5)
      COMMON /DIM/N,M
      DO 10 I = 1,M
         DO 10 J = 1,M
            A(I,J) = X(I,J)
10    CONTINUE
      DO 20 K = 1,M
         A(K,K) = -1.0/A(K,K)
         DO 30 I = 1,M
            IF ((I-K).NE.0) A(I,K) = -A(I,K)*A(K,K)
30       CONTINUE
         DO 40 I = 1,M
            DO 40 J = 1,M
               IF (((I-K)*(J-K)).NE.0) A(I,J) = A(I,J)-A(I,K)*A(K,J)
40       CONTINUE
         DO 50 J = 1,M
            IF ((J-K).NE.0) A(K,J) = -A(K,J)*A(K,K)
50       CONTINUE
20    CONTINUE
      DO 60  I = 1,M
         DO 60 J = 1,M
```

```
            XINV(I,J) = -A(I,J)
60      CONTINUE
        RETURN
        END
*****************************************************
*         ERROR DISTRIBUTION IS NORMAL              *
*****************************************************
        FUNCTION NORMAL(DMEAN,SIGMA)
        REAL RAND,NORMAL
        COMMON /SEED/IX,IT
        PI = 3.142857143
        IF (IT.EQ.1) GOTO 10
        RU1 = RAND(IX)
        RU2 = RAND(IX)
        U1 = SQRT(-2*ALOG(RU1))*COS(2*PI*RU2)
        U2 = SQRT(-2*ALOG(RU1))*SIN(2*PI*RU2)
        NORMAL = DMEAN+SIGMA*U1
        IT = 1
        RETURN
10      NORMAL = DMEAN+SIGMA*U2
        IT = 0
        RETURN
        END
*************************************************************
* ERROR DISTRIBUTION IS SCALE-CONTAMINATE NORMAL  *
*************************************************************
        FUNCTION SCALE(C,P,DMEAN,SIG1)
        REAL NORMAL,SIG2,Q,SCALE
        COMMON /SEED/IX,IT
        SIG2 = C*SIG1
        Q = RAND(IX)
```

```
      IF ((Q-P).LE.0) THEN
        SCALE = NORMAL(DMEAN,SIG2)
      ELSE
        SCALE = NORMAL(DMEAN,SIG1)
      ENDIF
      RETURN
      END
*****************************************************
*      ERROR DISTRIBUTION IS LOGNORMAL              *
*****************************************************
      FUNCTION LOGNOR(DMEAN,SIGMA)
      REAL LOGNOR,NORMAL,E
      E = NORMAL(DMEAN,SIGMA)
      LOGNOR = EXP(E)
      RETURN
      END
*****************************************************
*                RANDOM  UNIFORM(0,1)               *
*****************************************************
      FUNCTION RAND(IX)
      IX = (IX*16807)
      IF (IX.LT.0) IX = (IX+2147483647)+1
      RAND = IX
      RAND = RAND/2147483647.0
      RETURN
      END
*****************************************************
```

ประวัติผู้วิจัย

นางสาวพัชรี คุณะสารพันธ์ เกิดวันที่ 14 มิถุนายน พ.ศ. 2516 ที่ อ. พนัสนิคม จ. ชลบุรี สำเร็จการศึกษาปริญญาตรีวิทยาศาสตรบัณฑิต สาขาวิชาสถิติ (เกียรตินิยมอันดับ 2) จากมหาวิทยาลัยมหาสารคาม ในปีการศึกษา 2537 และเข้าศึกษาต่อในหลักสูตรสถิติศาสตรมหาบัณฑิต ที่จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2538

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย