รายการอ้างอิง

ภาษาไทย

[1] ควงพร ชูรักษ์. "การเปรียบเทียบการประมาณค่าในการวิเคราะห์ความถดถอยพหุ โดยวิธีริดจ์รีเกรสชัน รีเกรสชันพรินซิเปิ้ลคอมโพเนนท์และวิธีกำลังสองน้อยที่สุด ในกรณีที่เกิดพหุสัมพันธ์ระหว่างตัวแปรอิสระ" วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย., 2529.

[2] เจษฎาพร ยุทธนวิบูลย์ชัย. "การศึกษาเปรียบเทียบตัวประมาณริดจ์" วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย., 2533.

[3] จิราพุส พู่มนตรี. "การเปรียบเทียบตัวประมาณริดจ์สำหรับการวิเคราะห์การถดถอยแบบริดจ์." วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย., 2534.

ภาษาต่างประเทศ

[4] Authur E. Hoerl and Robert W. Kennard. "Ridge Regression : Biased Estimation for Nonorthogonal Problems." *Technometrics.*, February 1970., page 55-67.

[5] Authur E. Hoerl and Robert W. Kennard. "Ridge Regression : Applications to Nonorthogonal Problems." *Technometrics.*, February 1970., page 69-82.

[6] Liu Kejian. "A New Class of Biased Estimate in Linear Regression." *Commun. Statist.-Theory Meth.*, 22(2) 1993., page 393-402.

[7] Robert H. Crouse and Chun Jin. "Unbiased Ridge Estimation with Prior Information and Ridge Trace." *Commun. Statist.-Theory Meth.*, 24(9) 1995., page 2341-2354.

[8] Fikri Akdeniz and Selahattin Kaciranlar. "On the Almost Unbiased Generalized Liu Estimator and Unbiased Estimation of the Bias and MSE." *Commun. Statist.-Theory Meth.*, 24(7) 1995., page 1789-1797.

[9] Dean W. Wichern and Gilbert A.Churchill. "A Comparison of Ridge Estimators." *Technometrics.*, August 1978., page 301-311

[10] Gary C. Mcdonald and Diane I. Galarneau. "A Monte Carlo Evaluation of Some Ridge-Type Estimators." *J. Amer. Statist. Assoc.*, June 1975., page 407-416

[11] Shan S. Kuo. "Numerical methods and Computers." Massachusetts : Addison-Wesley,Inc., 1965., page 185-206.

ภาคผนวก

รายละเอียดของโปรแกรมที่ใช้ในการวิจัยมีดังนี้

ชื่อโปรแกรม	คุณสมบัติของโปรแกรม	ชื่อโปรแกรมที่เรียกใช้	หน้าที่
โปรแกรมหลัก			
MAIN	- อ่านค่าพารามิเตอร์ที่กำหนด	GENX,STDDZE,RP	180-186
	- คำนวน $\hat{\underset{\sim}{\beta}}$, $\hat{\underset{\sim}{\beta}}_R(kI,\underset{\sim}{J})$ และ $\hat{\underset{\sim}{\beta}}_L(D)$	CALXTX,EIGEN,LIU	
	- คำนวนและเปรียบเทียบค่า AMSE ทั้ง 3 วิธี	BUILDY,MSE	
		<u>หมายเหตุ</u> เมื่อความคลาดเคลื่อนมีการแจกแจงแบบลอกนอร์มอลจะเรียกโปรแกรม BOXCOX	
SUBROUTINE			
GENX	สร้างเมทริกซ์ของตัวแปรอิสระ X	NORMAL	186-187
RANDOM	สร้างเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอ		187
STDDZE	แปลงเมทริกซ์ X ให้อยู่ในรูปมาตรฐาน		190
BUILDY	คำนวนหาค่าตัวแปรตาม $\underset{\sim}{y}$		192
BOXCOX	แปลงค่าของ $\underset{\sim}{y}$ ตาม λ ที่เหมาะสม	OLS	187-189
JACOBI	คำนวนหาค่าเฉพาะและเวกเตอร์เฉพาะด้วยวิธี Jacobi		193-196
EIGEN	รับ-ส่งค่าเฉพาะและเวกเตอร์เฉพาะ	JACOBI	193
OLS	คำนวนหาค่า $\hat{\underset{\sim}{\beta}}$ และ S^2		197-198
RP	คำนวนหาค่า $\hat{\underset{\sim}{\beta}}_R(kI,\underset{\sim}{J})$	OLS,FINDK	198-199
LIU	คำนวนหาค่า $\hat{\underset{\sim}{\beta}}_L(D)$	OLS,EIGEN,DOPT	200
DOPT	คำนวนหาเมทริกซ์ D ที่เหมาะสม		201
CALXTX	คำนวนหาค่า $X'X$		201
INVRS	คำนวนหาเมทริกซ์ผกผัน		202
MSE	คำนวนหาค่า MSE ของทั้ง 3 วิธี		203

ชื่อโปรแกรม	คุณสมบัติของโปรแกรม	ชื่อโปรแกรมที่เรียกใช้	หน้าที่
FUNCTION			
NORMAL	สร้างเลขสุ่มที่มีการแจกแจงแบบปกติ	RANDOM	191
SCAL	สร้างเลขสุ่มที่มีการแจกแจงแบบปกติปลอมปน	RANDOM,NORMAL	191
LOGNOR	สร้างเลขสุ่มที่มีการแจกแจงแบบลอกนอล	NORMAL	192
FINDK	คำนวนหาค่า k ที่เหมาะสม	INVRS	199-200
SD	คำนวนหาค่าส่วนเบี่ยงเบนมาตรฐาน		202

```
C*************************************************************************
C*                                                                       *
C*          A COMPARISON OF PARAMETERS ESTIMATING METHODS IN             *
C*        MULTIPLE REGRESSION ANALYSIS BY LEAST SQUARE METHOD,           *
C*          RIDGE REGRESSION WITH PRIOR INFORMATION METHOD               *
C*              AND GENERALIZED LIU KEJIAN METHOD                        *
C*     WHEN EXISTING MULTICOLLINEARITY AMONG INDEPENDENT VARIABLES.      *
C*                                                                       *
C*************************************************************************
      INTEGER COUNT,BINT,DERROR,ROUND
      REAL    NORMAL,X(100,6),XT(6,100),ERROR(100),D(6,6),LOGNOR,SCAL,
     *        XSTAR(100,6),KOPT,SUMK,KBAR
      DOUBLE PRECISION XTX(6,6),VECMAX(6),VECMIN(6),EIGVAL(6),
     *                 BINIT(6),Y(100),BOLS(6),BRP(6),BLIU(6),
     *                 MSEOLS,MSERP,MSELIU,TMSLS1,TMSRP1,TMSLI1,
     *                 TMSLS2,TMSRP2,TMSLI2,AMSLS1,AMSRP1,AMSLI1,
     *                 AMSLS2,AMSRP2,AMSLI2,PD1,PD2,
     *                 DRP1,DRP2,DLK1,DLK2,DLS1,DLS2,TMSLS,TMSRP,TMSLK,
     *                 TMSLSS,TMSRPS,TMSLKS,AMSELS,AMSERP,AMSELK,
     *                 SDLS,SDRP,SDLK,SD,MSLS1,MSRP1,MSLK1,MSLS2,MSRP2,MSLK2,
     *                 MMSLS,MMSRP,MMSLK,YSTAR(100),BINITS(6)
      CHARACTER DISTRI*20
      COMMON /SEED/IX,KK/DIM/M
     *       /PARA/DMEAN,SIGMA
     *       /CORREL/CORR1,CORR2
      IX = 13
      KK = 0
      READ(5,*) COUNT,M,N
      IF (N.EQ.4) THEN
        READ(5,*) CORR1
      ELSE IF (N.EQ.6) THEN
```

```
      READ(5,*) CORR1,CORR2
    END IF
    READ(5,*) DERROR
    IF (DERROR.EQ.1) THEN
      READ(5,*) DMEAN,SIGMA
      DISTRI = 'NORMAL'
    ELSE IF (DERROR.EQ.2) THEN
      READ(5,*) C,P,DMEAN,SIGMA
      DISTRI = 'SCALE-CONTAMINATE'
    ELSE IF (DERROR.EQ.3) THEN
      READ(5,*) DMEAN,SIGMA
      DISTRI = 'LOGNORMAL'
    END IF
    DO 5 ROUND=1,COUNT
      CALL GENX(X,N)
      CALL STDDZE(X,XSTAR,N)
      CALL CALXTX(XSTAR,XTX,N)
      CALL EIGEN(XTX,N,EIGVAL,VECMAX,VECMIN)
      DO 11 BINT=1,2
        IF (BINT.EQ.1) THEN
          DO 20 I=1,N
            BINT(I) = VECMAX(I)
20        CONTINUE
        ELSE IF (BINT.EQ.2) THEN
          DO 21 I=1,N
            BINIT(I) = VECMIN(I)
21        CONTINUE
        END IF
        IF (DERROR.EQ.1) THEN
          DO 25 I=1,M
            ERROR(I) = NORMAL(DMEAN,SIGMA)
```

```
25        CONTINUE
        ELSE IF (DERROR.EQ.2) THEN
          DO 100 I=1,M
            ERROR(I) = SCAL(C,P,DMEAN,SIGMA)
100       CONTINUE
        ELSE IF (DERROR.EQ.3) THEN
          DO 101 I=1,M
            ERROR(I) = LOGNOR(DMEAN,SIGMA)
101       CONTINUE
        END IF
        CALL BUILDY(XSTAR,BINIT,ERROR,Y,N)
        IF (DERROR.EQ.3) THEN
          CALL BOXCOX(XSTAR,Y,N)
        END IF
        CALL RP(XSTAR,Y,BRP,KOPT,N)
        CALL LIU(XSTAR,Y,BOLS,BLIU,N)
        CALL MSE(BINIT,BOLS,BRP,BLIU,MSEOLS,MSERP,MSELIU,N)
        SUMK = SUMK+KOPT
        IF (BINT.EQ.1) THEN
          TMSLS1 = TMSLS1+MSEOLS
          TMSRP1 = TMSRP1+MSERP
          TMSLI1 = TMSLI1+MSELIU
          MSLS1 = MSEOLS
          MSRP1 = MSERP
          MSLK1 = MSELIU
        ELSE IF (BINT.EQ.2) THEN
          TMSLS2 = TMSLS2+MSEOLS
          TMSRP2 = TMSRP2+MSERP
          TMSLI2 = TMSLI2+MSELIU
          MSLS2 = MSEOLS
          MSRP2 = MSERP
```

```
            MSLK2 = MSELIU
          END IF
   11   CONTINUE
        MMSLS = (MSLS1+MSLS2)/2
        MMSRP = (MSRP1+MSRP2)/2
        MMSLK = (MSLK1+MSLK2)/2
        TMSLS = TMSLS+MMSLS
        TMSRP = TMSRP+MMSRP
        TMSLK = TMSLK+MMSLK
        TMSLSS = TMSLSS+MMSLS**2
        TMSRPS = TMSRPS+MMSRP**2
        TMSLKS = TMSLKS+MMSLK**2
    5 CONTINUE
      KBAR = SUMK/(COUNT*2)
      AMSELS = TMSLS/COUNT
      AMSERP = TMSRP/COUNT
      AMSELK = TMSLK/COUNT
      SDLS = SD(TMSLSS,AMSELS,COUNT)
      SDRP = SD(TMSRPS,AMSERP,COUNT)
      SDLK = SD(TMSLKS,AMSELK,COUNT)
      AMSLS1 = TMSLS1/COUNT
      AMSRP1 = TMSRP1/COUNT
      AMSLI1 = TMSLI1/COUNT
      AMSLS2 = TMSLS2/COUNT
      AMSRP2 = TMSRP2/COUNT
      AMSLI2 = TMSLI2/COUNT
      PD1 = ((AMSELS-AMSERP)/AMSELS)*100
      PD2 = ((AMSELS-AMSELK)/AMSELS)*100
      DRP1 = ((AMSELS-AMSERP)/AMSERP)*100
      DRP2 = ((AMSELK-AMSERP)/AMSERP)*100
      DLK1 = ((AMSELS-AMSELK)/AMSELK)*100
```

```
      DLK2 = ((AMSERP-AMSELK)/AMSELK)*100
      DLS1 = ((AMSERP-AMSELS)/AMSELS)*100
      DLS2 = ((AMSELK-AMSELS)/AMSELS)*100
      WRITE(6,400)
  400 FORMAT(////,22X,'SUMMARY OF RESULT')
      WRITE(6,405) COUNT
  405 FORMAT(/,2X,'ROUND OF SIMULATION                    :',I4)
      WRITE(6,410) M
  410 FORMAT(2X,'SAMPLE SIZE                              :',I4)
      WRITE(6,415) N
  415 FORMAT(2X,'NUMBER OF INDEPENDENT VARIABLES          :',I4)
      IF (DERROR.EQ.1.OR.DERROR.EQ.3) THEN
        WRITE(6,417) DISTRI
  417   FORMAT(2X,'DISTRIBUTION OF ERROR IS ',A18)
        WRITE(6,418) DMEAN,SIGMA
  418   FORMAT(21X,'DMEAN =',F4.1,5X,'SIGMA = ',F5.2)
      ELSE IF (DERROR.EQ.2) THEN
        WRITE(6,417) DISTRI
        WRITE(6,419) C,P,DMEAN,SIGMA
  419   FORMAT(12X,'C =',F4.1,3X,'P =',F5.2,3X,'DMEAN =',F4.1,3X,
     *'SIGMA =',F5.2)
      END IF
      WRITE(6,420) CORR1
  420 FORMAT(2X,'MULTICOLLINEARITY AMONG X1,X2 AND X3     :',F5.2)
      IF (N.EQ.6) THEN
        WRITE(6,430) CORR2
  430   FORMAT(2X,'MULTICOLLINEARITY BETWEEN X4 AND X5    :',F5.2)
      END IF
      DO 435 I=1,2
        IF (I.EQ.1) THEN
          WRITE(6,440)
```

```
440     FORMAT(/,2X,'TRUE BETA IS EIGENVECTOR CORRESPONDING TO LARGEST ',
     *  'EIGENVALUE')
        WRITE(6,441)
441     FORMAT(2X,'AVERAGE MEAN SQUARE ERROR')
        WRITE(6,442) AMSLS1
442     FORMAT(2X,'LEAST SQUARE METHOD                    :',F9.4)
        WRITE(6,443) AMSRP1
443     FORMAT(2X,'RIDGE REGRESSION WITH PRIOR INFORMATION METHOD :',F9.4)
        WRITE(6,444) AMSLI1
444     FORMAT(2X,'LIU KEJIAN "S METHOD                   :',F9.4)
      ELSE IF (I.EQ.2) THEN
        WRITE(6,445)
445     FORMAT(/,2X,'TRUE BETA IS EIGENVECTOR CORRESPONDING TO SMALLEST ',
     *  'EIGENVALUE')
        WRITE(6,442) AMSLS2
        WRITE(6,443) AMSRP2
        WRITE(6,444) AMSLI2
      END IF
435 CONTINUE
    WRITE(6,500) AMSELS
500 FORMAT(/,2X,'AVERAGE MEAN SQUARE ERROR LS        =',F10.4)
    WRITE(6,501) AMSERP
501 FORMAT(2X,'AVERAGE MEAN SQUARE ERROR RP          =',F10.4)
    WRITE(6,502) AMSELK
502 FORMAT(2X,'AVERAGE MEAN SQUARE ERROR LK          =',F10.4)
    WRITE(6,503) SDLS
503 FORMAT(2X,'STANDARD DEVIATION OF MSE(LS)         =',F10.4)
    WRITE(6,504) SDRP
504 FORMAT(2X,'STANDARD DEVIATION OF MSE(RP)         =',F10.4)
    WRITE(6,505) SDLK
505 FORMAT(2X,'STANDARD DEVIATION OF MSE(LK)         =',F10.4)
```

```
      WRITE(6,800) PD1
  800 FORMAT(/,2X,'AMSE(LS) HAVE MORE AMSE THAN AMSE(RP)    =',F10.4,' %')
      WRITE(6,810) PD2
  810 FORMAT(2X,'AMSE(LS) HAVE MORE AMSE THAN AMSE(LK)    =',F10.4,' %',/)
      IF ((AMSERP.LT.AMSELS).AND.(AMSERP.LT.AMSELK)) THEN
        WRITE(6,815) DRP1
  815   FORMAT(2X,'AMSE(RP) HAVE LESS AMSE THAN AMSE(LS) =',F10.4,' %')
        WRITE(6,820) DRP2
  820   FORMAT(2X,'AMSE(RP) HAVE LESS AMSE THAN AMSE(LK) =',F10.4,' %')
      ELSE IF ((AMSELK.LT.AMSELS).AND.(AMSELK.LT.AMSERP)) THEN
        WRITE(6,825) DLK1
  825   FORMAT(2X,'AMSE(LK) HAVE LESS AMSE THAN AMSE(LS) =',F10.4,' %')
        WRITE(6,830) DLK2
  830   FORMAT(2X,'AMSE(LK) HAVE LESS AMSE THAN AMSE(RP) =',F10.4,' %')
      ELSE
        WRITE(6,835) DLS1
  835   FORMAT(2X,'AMSE(LS) HAVE LESS AMSE THAN AMSE(RP) =',F10.4,' %')
        WRITE(6,840) DLS2
  840   FORMAT(2X,'AMSE(LS) HAVE LESS AMSE THAN AMSE(LK) =',F10.4,' %')
      END IF
      WRITE(6,845) KBAR
  845 FORMAT(2X,'AVERAGE OF VALUE K                =',F10.5)
      STOP
      END
C*************************************************
      SUBROUTINE GENX(X,N)
      REAL X(100,6),Z(100,7),NORMAL
      COMMON /SEED/IX,KK
     *      /DIM/M
     *      /CORREL/CORR1,CORR2
```

```
      DO 10 I=1,M
        Z(I,N+1) = NORMAL(0.0,1.0)
        DO 20 J=1,N
          IF (J.EQ.1) THEN
            X(I,J) = 1.0
          ELSE IF (J.GE.2.AND.J.LE.4) THEN
            Z(I,J) = NORMAL(0.0,1.0)
            X(I,J) = SQRT(1-CORR1)*Z(I,J)+SQRT(CORR1)*Z(I,N+1)
          ELSE IF (J.GE.5) THEN
            Z(I,J) = NORMAL(0.0,1.0)
            X(I,J) = SQRT(1-CORR2)*Z(I,J)+SQRT(CORR2)*Z(I,N+1)
          END IF
   20   CONTINUE
   10 CONTINUE
      RETURN
      END
C*************************************************
      SUBROUTINE RANDOM(IX,IY,FLY)
      IY = IX*16807
      IF (IY.LT.0) IY = IY+2147483647+1
      FLY = IY
      FLY = FLY/2147483647
      IX = IY
      RETURN
      END
C*************************************************
      SUBROUTINE BOXCOX(X,Y,N)
      DOUBLE PRECISION Y(100),YTRANS(100),YSTAR(100),SUM,GEOMN,
     *                 XTX(6,6),XTY(6),SIG2,BOLS(6),S(3)
      REAL X(100,6),LAMMIN,LAMOPT,LAMMAX,LAMB
      INTEGER FAIL
```

```
      CHARACTER STATUS*10
      COMMON /DIM/M
      SUM = 0.0
      DO 5 I=1,M
        IF (Y(I).LE.0) THEN
          FAIL = FAIL+1
          WRITE(6,6)
    6     FORMAT(' CAN NOT FIND GEOMETRIC MEAN')
          WRITE(6,7) FAIL
    7     FORMAT(' FAIL =',I3)
          RETURN
        END IF
    5 CONTINUE
      DO 10 I=1,M
        SUM = SUM+DLOG(Y(I))
   10 CONTINUE
      GEOMN = DEXP(SUM/M)
      DO 20 I=1,M
        YSTAR(I) = Y(I)/GEOMN
   20 CONTINUE
      LAMMIN = -15.00
      LAMMAX = 16.00
      E = 0.01
   15 LAMOPT = (LAMMIN+LAMMAX)/2
      DO 40 I=1,3
        LAMB = LAMOPT+(I-2)*E
        DO 30 J=1,M
          IF (LAMB.NE.0) THEN
            YTRANS(J) = ((YSTAR(J)**LAMB)-1)/LAMB
          ELSE
            YTRANS(J) = DLOG(YSTAR(J))
```

```
        END IF
 30     CONTINUE
        CALL OLS(X,YTRANS,XTX,XTY,SIG2,BOLS,N)
        S(I) = SIG2
 40   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 15
      DO 50 I=1,M
        IF (LAMOPT.NE.0) THEN
          Y(I) = ((YSTAR(I)**LAMOPT)-1)/LAMOPT
        ELSE
          Y(I) = DLOG(YSTAR(I))
        END IF
 50   CONTINUE
      STATUS = 'RETURN'
      RETURN
      END
C*********************************************
```

```
      SUBROUTINE STDDZE(X,XSTAR,N)
      REAL X(100,6),XSTAR(100,6)
      DOUBLE PRECISION BINIT(6),Y(100),SUMX(6),SUMXSQ(6),SUMY,SUMYSQ,
     *                 XBAR(6),YBAR,SYY,SXX(6),BINITS(6),YSTAR(100)
      COMMON /DIM/M
      DO 10 J=2,N
        SUMX(J) = 0.0
        SUMXSQ(J) = 0.0
      DO 10 I=1,M
        SUMX(J) = SUMX(J)+X(I,J)
        SUMXSQ(J) = SUMXSQ(J)+X(I,J)**2
   10 CONTINUE
      DO 15 J=2,N
        XBAR(J) = SUMX(J)/M
        SXX(J) = SYY(SUMXSQ(J),XBAR(J),M)
   15 CONTINUE
      DO 20 J=1,N
      DO 20 I=1,M
        IF (J.EQ.1) THEN
          XSTAR(I,J) = X(I,J)
        ELSE
          XSTAR(I,J) = ((X(I,J)-XBAR))/SXX(J))
        END IF
   20 CONTINUE
      RETURN
      END
C*****************************************************
```

```
      REAL FUNCTION NORMAL(DMEAN,SIGMA)
      COMMON /SEED/IX,KK
      PI = 3.1415926
      IF (KK.EQ.1) GOTO 10
      CALL RANDOM(IX,IY,FLY)
      R1 = FLY
      CALL RANDOM(IX,IY,FLY)
      R2 = FLY
      Z1 = SQRT(-2*ALOG(R1))*COS(2*PI*R2)
      Z2 = SQRT(-2*ALOG(R1))*SIN(2*PI*R2)
      NORMAL = Z1*SIGMA+DMEAN
      KK = 1
      RETURN
   10 NORMAL = Z2*SIGMA+DMEAN
      KK = 0
      RETURN
      END
C*************************************************
      REAL FUNCTION SCAL(C,P,DMEAN,SIGMA)
      REAL NORMAL,C,ADJSIG
      COMMON /SEED/IX,KK
      ADJSIG = C*SIGMA
      CALL RANDOM(IX,IY,FLY)
      IF (FLY.LE.P) THEN
        SCAL = NORMAL(DMEAN,ADJSIG)
      ELSE
        SCAL = NORMAL(DMEAN,SIGMA)
      END IF
      RETURN
      END
C*************************************************
```

```
      REAL FUNCTION LOGNOR(DMEAN,SIGMA)
      REAL NORM,NORMAL
      COMMON /SEED/IX,KK
      NORM = NORMAL(DMEAN,SIGMA)
      LOGNOR = EXP(NORM)
      RETURN
      END
C*************************************************
      SUBROUTINE BUILDY(X,BINIT,ERROR,Y,N)
      REAL X(100,6),ERROR(100)
      DOUBLE PRECISION BINIT(6),TY(100),Y(100)
      COMMON /DIM/M
      DO 10 I=1,M
        TY(I) = 0.0
        DO 15 K=1,N
          TY(I) = TY(I)+X(I,K)*BINIT(K)
 15     CONTINUE
        Y(I) = TY(I)+ERROR(I)
 10   CONTINUE
      RETURN
      END
C*************************************************
```

```
      SUBROUTINE EIGEN(A,N,EIGVAL,VECMAX,VECMIN)
      DOUBLE PRECISION A(6,6),X(6,6),EIGVAL(6),VECMAX(6),VECMIN(6)
      CALL JACOBI(N,A,1,NR,X)
      DO 1 I=1,N
        EIGVAL(I) = A(I,I)
        VECMAX(I) = X(I,2)
        VECMIN(I) = X(I,N)
    1 CONTINUE
      RETURN
      END
C*************************************************
      SUBROUTINE JACOBI(N,Q,JVEC,M,V)
      DOUBLE PRECISION Q(6,6),V(6,6),IH(6),X(6)
      IF (JVEC) 10,15,10
   10 DO 14 I=1,N
      DO 14 J=1,N
        IF (I-J) 12,11,12
   11     V(I,J) = 1.0
          GOTO 14
   12     V(I,J) = 0.0
   14 CONTINUE
   15 M = 0
   17 MI = N-1
      DO 30 I=1,MI
        X(I) = 0.0
        MJ = I+1
        DO 30 J=MJ,N
          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 M = M+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,N
            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
      X(JP) = 0.0
      DO 530 I=1,N
        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,N
      TEMP = V(I,IP)
      V(I,IP) = COSN*TEMP+SINE*V(I,JP)
550   V(I,JP) = -SINE*TEMP+COSN*V(I,JP)
    GOTO 40
1000 RETURN
    END
C*****************************************************
```

```
      SUBROUTINE OLS(X,Y,XTX,XTY,SIG2,BOLS,N)
      REAL X(100,6),XT(6,100)
      DOUBLE PRECISION XTX(6,6),XTXINV(6,6),XTY(6),BOLS(6),Y(100)
      DOUBLE PRECISION YTY,BTXTY,SIG2
      COMMON /DIM/M
      DO 10 I=1,M
      DO 10 J=1,N
        XT(J,I) = X(I,J)
   10 CONTINUE
      DO 15 I=1,N
      DO 15 J=1,N
        XTX(I,J) = 0.0
        DO 15 K=1,M
          XTX(I,J) = XTX(I,J)+XT(I,K)*X(K,J)
   15 CONTINUE
      DO 25 I=1,N
        XTY(I) = 0.0
        DO 25 K=1,M
          XTY(I) = XTY(I)+XT(I,K)*Y(K)
   25 CONTINUE
      CALL INVRS(XTX,XTXINV,N)
      DO 35 I=1,N
        BOLS(I) = 0.0
        DO 35 K=1,N
          BOLS(I) = BOLS(I)+XTXINV(I,K)*XTY(K)
   35 CONTINUE
      YTY = 0.0
      DO 40 I=1,M
        YTY = YTY+Y(I)**2
   40 CONTINUE
     BTXTY = 0.0
```

```
      DO 45 I=1,N
        BTXTY = BTXTY+BOLS(I)*XTY(I)
   45 CONTINUE
      SIG2 = (YTY-BTXTY)/(M-N)
      RETURN
      END
C*************************************************
      SUBROUTINE RP(XSTAR,BRP,KOPT,N)
      REAL X(100,6),XT(6,100),XSTAR(100,6)
      REAL KOPT,TBOLS,JBAR,SIG2,TRXTXI,TRACE,FINDK
      DOUBLE PRECISION BRP(6),XTX(6,6),XTY(6),BOLS(6),Y(100)
      DOUBLE PRECISION XTXINV(6,6),XTXRP(6,6),XTYRP(6),YSTAR(100)
      COMMON /DIM/M
      CALL OLS(XSTAR,Y,XTX,XTY,SIG2,BOLS,N)
      TBOLS = 0.0
      DO 25 I=1,N
        TBOLS = TBOLS+BOLS(I)
   25 CONTINUE
      JBAR = TBOLS/N
      KOPT = FINDK(N,SIG2,BOLS,JBAR,XTX)
      DO 10 I=1,N
      DO 10 J=1,N
        IF (I.EQ.J) THEN
          XTXRP(I,J) = XTX(I,J)+KOPT
        ELSE
          XTXRP(I,J) = XTX(I,J)
        END IF
   10 CONTINUE
      DO 30 I=1,N
        XTYRP(I) = XTY(I)+KOPT*JBAR
   30 CONTINUE
```

```
      CALL INVRS(XTXRP,XTXINV,N)
      DO 40 I=1,N
        BRP(I) = 0.0
        DO 40 K=1,N
          BRP(I) = BRP(I)+XTXINV(I,K)*XTYRP(K)
   40 CONTINUE
      RETURN
      END
C*****************************************************
      REAL FUNCTION FINDK(N,SIG2,BOLS,JBAR,XTX)
      REAL JBAR,TRACE,TBJ
      DOUBLE PRECISION BOLS(6),XTX(6,6),XTXINV(6,6),SIG2
      INTEGER P
      CALL INVRS(XTX,XTXINV,N)
      TRACE = 0.0
      DO 5 I=1,N
      DO 5 J=1,N
        IF (I.EQ.J) THEN
          TRACE = TRACE+XTXINV(I,J)
        END IF
    5 CONTINUE
      TBJ = 0.0
      DO 10 I=1,N
        TBJ = TBJ+((BOLS(I)-JBAR)**2)
   10 CONTINUE
      P = N
      IF (TBJ.GT.(SIG2*TRACE)) THEN
        FINDK = (P*SIG2)/(TBJ-(SIG2*TRACE))
      ELSE
        FINDK = (P*SIG2)/TBJ
      END IF
```

```
      RETURN
      END
C*****************************************************
      SUBROUTINE LIU(XSTAR,Y,BOLS,BLIU,N)
      REAL X(100,6),XT(6,100),D(6,6),XSTAR(100,6)
      DOUBLE PRECISION XTX(6,6),XTY(6),BOLS(6),BLIU(6),XTXLIU(6,6),
     *                 XTYLIU(6),SIG2,XTXINV(6,6),Y(100),EIGVAL(6),
     *                 VECMAX(6),VECMIN(6),YSTAR(100)
      COMMON /DIM/M
      CALL OLS(XSTAR,Y,XTX,XTY,SIG2,BOLS,N)
      DO 15 I=1,N
      DO 15 J=1,N
        IF (I.EQ.J) THEN
          XTXLIU(I,J) = XTX(I,J)+1
        ELSE
          XTXLIU(I,J) = XTX(I,J)
        END IF
   15 CONTINUE
      CALL EIGEN(XTX,N,EIGVAL,VECMAX,VECMIN)
      CALL DOPT(BOLS,SIG2,EIGVAL,D,N)
      DO 30 I=1,N
        XTYLIU(I) = XTY(I)+D(I,I)*BOLS(I)
   30 CONTINUE
      CALL INVRS(XTXLIU,XTXINV,N)
      DO 40 I=1,N
        BLIU(I) = 0.0
        DO 40 K=1,N
          BLIU(I) = BLIU(I)+XTXINV(I,K)*XTYLIU(K)
   40 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DOPT(BOLS,SIG2,EIGVAL,D,N)
      REAL D(6,6)
      DOUBLE PRECISION BOLS(6),EIGVAL(6),P1(6),P2(6),SIG2
      COMMON /DIM/M
      DO 10 I=1,N
        P1(I) = EIGVAL(I)*((BOLS(I)**2)-SIG2)
        P2(I) = (EIGVAL(I)*(BOLS(I)**2))+SIG2
        D(I,I) = P1(I)/P2(I)
   10 CONTINUE
      RETURN
      END
C*****************************************************
      SUBROUTINE CALXTX(X,XTX,N)
      REAL X(100,6),XT(6,100)
      DOUBLE PRECISION XTX(6,6)
      COMMON /DIM/M
      DO 10 I=1,M
      DO 10 J=1,N
        XT(J,I) = X(I,J)
   10 CONTINUE
      DO 15 I=1,N
      DO 15 J=1,N
        XTX(I,J) = 0.0
        DO 15 K=1,M
          XTX(I,J) = XTX(I,J)+XT(I,K)*X(K,J)
   15 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INVRS(X,XINV,N)
      DOUBLE PRECISION X(6,6),XINV(6,6),A(6,6)
      DO 1 I=1,N
      DO 1 J=1,N
        A(I,J) = X(I,J)
    1 CONTINUE
      DO 20 K=1,N
        A(K,K) = -1.0/A(K,K)
        DO 5 I=1,N
          IF ((I-K).NE.0) A(I,K) = -A(I,K)*A(K,K)
    5   CONTINUE
        DO 10 I=1,N
        DO 10 J=1,N
          IF (((I-K)*(J-K)).NE.0) A(I,J) = A(I,J)-A(I,K)*A(K,J)
   10   CONTINUE
        DO 15 J=1,N
          IF ((J-K).NE.0) A(K,J) = -A(K,J)*A(K,K)
   15   CONTINUE
   20 CONTINUE
      DO 25 I=1,N
      DO 25 J=1,N
        XINV(I,J) = -A(I,J)
   25 CONTINUE
      RETURN
      END
C*******************************************************
      DOUBLE PRECISION FUNCTION SD(YSQR,YBAR,COUNT)
      DOUBLE PRECISION YSQR,YBAR
      INTEGER COUNT
      SD = DSQRT((YSQR-(COUNT*(YBAR**2)))/(COUNT-1))
      RETURN
      END
```

```
      DOUBLE PRECISION FUNCITION SYY(YSQR,YBAR,COUNT)
      DOUBLE PRECISION YSQR,YBAR
      INTEGER COUNT
      SYY = DSQRT(YSQR-(COUNT*(YBAR**2)))
      RETURN
      END
C*************************************************
      SUBROUTINE MSE(BINITS,BOLS,BRP,BLIU,MSEOLS,MSERP,MSELIU,N)
      DOUBLE PRECISION BINITS(6),BOLS(6),BRP(6),BLIU(6)
      DOUBLE PRECISION MSEOLS,MSERP,MSELIU
      COMMON /DIM/M
      MSEOLS = 0.0
      MSERP  = 0.0
      MSELIU = 0.0
      DO 10 I=1,N
        MSEOLS = MSEOLS+(BINITS(I)-BOLS(I))**2
        MSERP  = MSERP +(BINITS(I)-BRP(I))**2
        MSELIU = MSELIU+(BINITS(I)-BLIU(I))**2
   10 CONTINUE
      RETURN
      END
```

ประวัติผู้วิจัย

นายสมพล จารุธนศักดิ์กูร เกิดวันที่ 30 กรกฎาคม พ.ศ. 2516 สำเร็จการศึกษาระดับปริญญาตรี จากสาขาสถิติ คณะวิทยาศาสตร์ มหาวิทยาลัยบูรพา ในปีการศึกษา 2536 และได้เข้าศึกษาต่อที่ภาควิชาสถิติ คณะพาณิชย์ศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปี พ.ศ. 2537