รายการอ้างอิง

ภาษาไทย

จิรายุส พู่มนตรี. 2534. การเปรียบเทียบตัวประมาณริดจ์สำหรับการวิเคราะห์การถดถอยแบบริดจ์. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย.

เจษฎาพร ยุทธนวิบูลน์ชัย. 2532. การศึกษาเปรียบเทียบตัวประมาณริดน์. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย.

ธันยากร ต้นชลชันธ์. 2539. การเปรียบเทียบการประมาณค่าสัมประสิทธิ์การถดถอยพหุคูณโดยวิธีกำลังสองน้อยสุด วิธีริดจ์รีเกรสชั่น และวิธีที่ใช้หลักการของริดน์และสไตล์ในกรณีที่เกิดพหุสัมพันธ์ระหว่างตัวแปรอิสระ. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย.

วิชิต หล่อจีระชุณห์กุล. 2539. เทคนิคการพยากรณ์. 1000 เล่ม. พิมพ์ครั้งที่ 1. กรุงเทพ : โครงการส่งเสริมเอกสารวิชาการ สถาบันบัณฑิตพัฒนบริหารศาสตร์.

สมพล จารุธนศักดิ์กูร. 2539. การเปรียบเทียบวิธีประมาณค่าพารามิเตอร์ในการวิเคราะห์ความถดถอยพหุคูณด้วยวิธีกำลังสองน้อยสุด วิธีริดจ์รีเกรสชั่นที่ใช้ข้อสนเทศโดยหลักเกณฑ์และวิธีลิว ดิเจียนทั่วไป เมื่อเกิดพหุสัมพันธ์ระหว่างตัวแปรอิสระ. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย.

ภาษาต่างประเทศ

Hoerl, and Kennard , Robert W. 1970. Ridge Regression : Applications to Nonorthogonal Problems. Technometrics 12: 69-82.

Hoerl, R., Schuenmeyer, J., and Hoerl, A. 1986. A Simulation of Biased Estimation and Subset Selection Regression Tecnique. Technometrics 28: 369-380.

Leo Breiman. 1995. Better Subset Regression Using the Nonnegative Garrote. Technometrics 37: 373-384.

Leo Breiman. 1995. Heuristics of Instability and Stabilization in Model Selection. The Annals of Statistics 24: 2350-2383.

Wichern, D.W. and Churchill, A.G. 1978. A Comparison of Ridge Estimators. Technometrics 20: 304.

ภาคผนวก

ภาคผนวก

มีรายละเอียดต่าง ๆ ดังต่อไปนี้

1. ตารางแสดงรายละเอียดของโปรแกรมที่ใช้ในการวิจัย

2. โปรแกรมหลัก 1 สำหรับการแจกแจงแบบปกติและปกติปลอมปน

3. โปรแกรมหลัก 2 สำหรับการแจกแจงแบบไวบูลล์และลอกนอร์มอล

4. โปรแกรมย่อยต่าง ๆ สำหรับใช้ในการวิจัย

รายละเอียดของโปรแกรมที่ใช้ในการวิจัยมีดังนี้

ชื่อโปรแกรม	คุณสมบัติของโปรแกรม	ชื่อโปรแกรมที่เรียกใช้	หน้าที่
MAIN 1	- อ่านค่าพารามิเตอร์ที่กำหนด - คำนวณค่า $\hat{\beta}_i$ ของทั้ง 3 วิธี - คำนวณค่า MSE ของทั้ง 3 วิธี - คำนวณค่า AMSE ของทั้ง 3 วิธี หมายเหตุเมื่อความคลาดเคลื่อนมีการแจกแจงแบบปกติและปกติปลอมปน	SIMX,STAND, EIGEN,ROPT, NORMAL,SCNOR, INMA,SIMY,BOLS ,BRID,BGAR, BMSE	190-196
MAIN 2	- อ่านค่าพารามิเตอร์ที่กำหนด - คำนวณค่า $\hat{\beta}_i$ ของทั้ง 3 วิธี - คำนวณค่า MSE ของทั้ง 3 วิธี - คำนวณค่า AMSE ของทั้ง 3 วิธี หมายเหตุ เมื่อความคลาดเคลื่อนมีการแจกแจงแบบไวบูลล์และลอกนอร์มอล	SIMX,STAND, EIGEN,ROPT, LOGNOR,WEIBUL ,INMA,SIMY, BOLS,BRID,BGAR ,BMSE	196-202
SIMX	สร้างเมทริกซ์ของตัวแปรอิสระ X ให้มีความสัมพันธ์ตามที่กำหนดและหาค่าสหสัมพันธ์ระหว่างตัวแปรอิสระ	NORMAL,CORRE	202-203
CORRE	หาค่าสหสัมพันธ์ระหว่างตัวแปรอิสระ		203-204
STAND	แปลงเมทริกซ์ของตัวแปรอิสระ X ให้อยู่ในรูปมาตรฐาน		204-205
EIGEN	หาค่าเฉพาะ(eigenvalues) และเวกเตอร์เฉพาะ(eigenvectors) โดยวิธีทำซ้ำของวิธี Jacobi	EIGE	205-206

ชื่อโปรแกรม	คุณสมบัติของโปรแกรม	ชื่อโปรแกรมที่เรียกใช้	หน้าที่
EIGE	หาเวกเตอร์เฉพาะที่ได้จากค่าเฉพาะที่มากที่สุด		206-208
INMA	คำนวณหาเมทริกซ์ผกผัน		208-209
ROPT	คำนวณหาค่า k ที่เหมาะสมเพื่อใช้ในสมการถดถอยริดจ์	INMA	209-211
SIMY	คำนวณหาค่าตัวแปรตาม y		211
BOLS	คำนวณ $\hat{\beta}$ จากวิธีกำลังสองน้อยสุด		211-212
BRID	คำนวณ $\hat{\beta}$ จากวิธีสมการการถดถอยริดจ์โดยใช้วิธีของบลีแมน	INMA	212-213
BGAR	คำนวณ $\hat{\beta}$ จากวิธีสมการการถดถอยเชิงเส้นของการ์ล็อต	INMA	213-214
BMSE	หาค่า MSE ของตัวประมาณ β ของแต่ละวิธี	INMA	214
RANDU	สร้างเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอ		214
NORMAL	สร้างเลขสุ่มที่มีการแจกแจงแบบปกติ	RANDU	215
SCNOR	สร้างเลขสุ่มที่มีการแจกแจงแบบปกติปลอมปน	RANDU,NORMAL	215
WEIBUL	สร้างเลขสุ่มที่มีการแจกแจงแบบไวบูลล์	RANDU	216
LOGNOR	สร้างเลขสุ่มที่มีการแจกแจงแบบลอกนอร์มอล	NORMAL	216

โปรแกรมหลัก 1 สำหรับการแจกแจงแบบปกติและปกติปลอมปน

```
C****  MAIN 1 PROGRAM FOR NORMAL & SCALE DISTRIBUTION ******
      INTEGER M,N,COUNT,IX
      REAL SCNOR,NORMAL,AM,SD,C,P
      COMMON /SEED/IX,K1/NO/M1,N/CORR/COR1,COR2,COR3,COR4
      REAL IN(6,6),XTX(6,6),Y(100),XTY(10)
     *,MAXEIG(6),MINEIG(6),MSE,SUM(6),TMEAN(6)
     *,OBETA(6),RBETA(6),GBETA(6)
     *,X(100,6),ERR(100),SXX(100,6),RSING(6,6)
      DOUBLE PRECISION NOMSE(500),NRMSE(500),NGMSE(500)
     *,SNOMSE,SNRMSE,SNGMSE
     *,ANOMSE,ANRMSE,ANGMSE
     *,SQNOMSE,SQNRMSE,SQNGMSE
     *,VNOMSE,VNRMSE,VNGMSE
     *,SDNOMSE,SDNRMSE,SDNGMSE
      DOUBLE PRECISION PNOR,PNOG,PNRO,PNRG,PNGO,PNGR
      DOUBLE PRECISION COMSE(500),CRMSE(500),CGMSE(500)
     *,SCOMSE,SCRMSE,SCGMSE
     *,ACOMSE,ACRMSE,ACGMSE
     *,SQCOMSE,SQCRMSE,SQCGMSE
     *,VCOMSE,VCRMSE,VCGMSE
     *,SDCOMSE,SDCRMSE,SDCGMSE
      DOUBLE PRECISION PCOR,PCOG,PCRO,PCRG,PCGO,PCGR
      OPEN(6,FILE = 'D:\FORTRAN\EIR141.OUT')
C************************ SET VALUE ************************
      M = 3.0
      AM = 1.0
      SD = 0.05
      C = 3.0
      P = 0.05
      COUNT= 500
      M1 = M+1
C************************ SET  SAMPLE SIZE *****************
      WRITE(6,3) AM,SD
 3    FORMAT(2X,'NORMAL DISTRIBUTION ',
     *2X,'MEAN = ',F5.2,2X,'SD = ',F5.3)
      WRITE(6,4) C,P,AM,SD
 4    FORMAT(2X,'SCALE-NORMAL DISTRIBUTION ',
     * 2X,'SCALE = ',F5.2,2X,'PERCENT = ',F5.2,
     * 2X,'MEAN = ',F5.2,2X,'SD = ',F5.2)
```

```
      DO 5 ISA = 1,4
      GOTO(10,15,20,25),ISA
 10   N = 10
      GOTO 35
 15   N = 30
      GOTO 35
 20   N = 50
      GOTO 35
 25   N = 100
c******************* SET  CORRELATION ************************
 35   DO 40 ICO = 1,6
      GOTO(45,50,55,60,65,70),ICO
 45   COR1 = 0.1
      COR2 = 0.01
      COR3 = 0.001
      COR4 = 0.0001
      GOTO 75
 50   COR1 = 0.3
      COR2 = 0.09
      COR3 = 0.027
      COR4 = 0.0081
      GOTO 75
 55   COR1 = 0.5
      COR2 = 0.25
      COR3 = 0.125
      COR4 = 0.0625
      GOTO 75
 60    COR1 = 0.7
       COR2 = 0.49
       COR3 = 0.343
       COR4 = 0.2401
       GOTO 75
 65   COR1 = 0.9
      COR2 = 0.81
      COR3 = 0.729
      COR4 = 0.6561
      GOTO 75
 70   COR1 = 0.99
      COR2 = 0.9801
      COR3 = 0.9703
```

```
      COR4 = 0.9650
 75   K1 = 0.0
      IX = 2917142
C********************SET VALUE EQUAL ZERO ****************
      SNOMSE = 0.0
      SNRMSE = 0.0
      SNGMSE = 0.0
      SQNOMSE = 0.0
      SQNRMSE = 0.0
      SQNGMSE = 0.0

      SCOMSE = 0.0
      SCRMSE = 0.0
      SCGMSE = 0.0
      SQCOMSE = 0.0
      SQCRMSE = 0.0
      SQCGMSE = 0.0
C*************************************************************
      WRITE(6,80) N ,M,COUNT,COR1
 80    FORMAT(/,2X,'SAMPLE SIZES = ',I3,2X,'VARIABLES=',I2
     *,/,2X,'COUNT OF SIMULATION =',I3
     *,2X,'BEGIN CORRELATION =',F4.2)
      CALL SIMX(X,TMEAN,SUM)
      CALL STAND(X,TMEAN,XTX,SXX,SUM)
      CALL EIGEN(XTX,MAXEIG,MINEIG)
      CALL ROPT(XTX,RSING)
C******************* USED CALL  INVERSE  MATRIX ****************
      DO 85 J=1,M1
      DO 85 K=1,M1
      IN(K,J) = XTX(K,J)
      IN(J,K) = XTX(J,K)
 85   CONTINUE
      DO 90 K=1,M1
      IF (IN(K,K)) 90,95,90
 95   WRITE(6,105)
105   FORMAT('IN(K,K) HAS ZERO ON DIAGONAL')
       STOP
 90   CONTINUE
      CALL INMA(IN)
C*************************************************************
```

```
      DO 110 I = 1,2
      IF (I.EQ.1) THEN
      DO 115 K = 1,COUNT
      DO 120 J = 1,N
      ERR(J) = NORMAL(AM,SD)
120 CONTINUE
      CALL SIMY(Y,SXX,ERR,MAXEIG)
C******************************** OLS ********************
      CALL BOLS(IN,SXX,XTY,Y,OBETA)
      CALL BMSE(OBETA,MAXEIG,MSE)
      NOMSE(K) = MSE
C******************************** RID ********************
      CALL BRID(RSING,XTX,XTY,RBETA)
      CALL BMSE(RBETA,MAXEIG,MSE)
      NRMSE(K) = MSE
C******************************** GARROTE *****************
      CALL BGAR(OBETA,SXX,Y,GBETA)
      CALL BMSE(GBETA,MAXEIG,MSE)
      NGMSE(K) = MSE
C********************** SUM MSE  ******************
      SNOMSE = SNOMSE + NOMSE(K)
      SNRMSE = SNRMSE + NRMSE(K)
      SNGMSE = SNGMSE + NGMSE(K)
C********************** AVERAGE MSE ***************
      ANOMSE = SNOMSE/FLOAT(COUNT)
      ANRMSE = SNRMSE/FLOAT(COUNT)
      ANGMSE = SNGMSE/FLOAT(COUNT)
C************************ SQUARE MSE ****************
      SQNOMSE = SQNOMSE + NOMSE(K)**2
      SQNRMSE = SQNRMSE + NRMSE(K)**2
      SQNGMSE = SQNGMSE + NGMSE(K)**2
C************************ VARIANCE MSE *************
      VNOMSE = (SQNOMSE-(COUNT*(ANOMSE**2)))/FLOAT(COUNT-1)
      VNRMSE = (SQNRMSE-(COUNT*(ANRMSE**2)))/FLOAT(COUNT-1)
      VNGMSE = (SQNGMSE-(COUNT*(ANGMSE**2)))/FLOAT(COUNT-1)
C********************** STANDARD DIVATION MSE**************
      SDNOMSE = DSQRT(VNOMSE)
      SDNRMSE = DSQRT(VNRMSE)
      SDNGMSE = DSQRT(VNGMSE)
C*********************************************************
```

```
      PNOR = ((ANRMSE-ANOMSE)/ANOMSE)*100
      PNOG = ((ANGMSE-ANOMSE)/ANOMSE)*100
      PNRO = ((ANOMSE-ANRMSE)/ANRMSE)*100
      PNRG = ((ANGMSE-ANRMSE)/ANRMSE)*100
      PNGO = ((ANOMSE-ANGMSE)/ANGMSE)*100
      PNGR = ((ANRMSE-ANGMSE)/ANGMSE)*100
C*********************************************************
115 CONTINUE
      WRITE(6,125) ANOMSE,ANRMSE,ANGMSE
125 FORMAT(2X,'AV OLS= ',F20.10,2X,'AV RID= ',F20.10
     *,2X,'AV GAR= ',F20.10 )
      WRITE(6,130) SDNOMSE,SDNRMSE,SDNGMSE
 130 FORMAT(2X,'SD OLS= ',F20.10,2X,'SD RID= ',F20.10
     *,2X,'SD GAR= ',F20.10)
      WRITE(6,135) PNOR,PNOG
 135 FORMAT(2X,'MOLS&RID= ',F20.10,2X,'MOLS&GAR=',F20.10)
      WRITE(6,140) PNRO,PNRG
 140 FORMAT(2X,'MRID&OLS= ',F20.10,2X,'MRID&GAR=',F20.10)
      WRITE(6,145) PNGO,PNGR
 145 FORMAT(2X,'MGAR&OLS= ',F20.10,2X,'MGAR&RID=',F20.10)
      ELSE
      DO 150 K = 1,COUNT
      DO 155 J = 1,N
      ERR(J) = SCNOR(C,P,AM,SD)
155  CONTINUE
      CALL SIMY(Y,SXX,ERR,MAXEIG)
C******************************** OLS ********************
      CALL BOLS(IN,SXX,XTY,Y,OBETA)
      CALL BMSE(OBETA,MAXEIG,MSE)
      COMSE(K) = MSE
C******************************** RID ********************
      CALL BRID(RSING,XTX,XTY,RBETA)
      CALL BMSE(RBETA,MAXEIG,MSE)
      CRMSE(K) = MSE
C******************************** GARROTE ****************
      CALL BGAR(OBETA,SXX,Y,GBETA)
      CALL BMSE(GBETA,MAXEIG,MSE)
      CGMSE(K) = MSE
C********************** SUM MSE  ***************
      SCOMSE = SCOMSE + COMSE(K)
```

```
      SCRMSE = SCRMSE + CRMSE(K)
      SCGMSE = SCGMSE + CGMSE(K)
C********************** AVERAGE MSE ***************
      ACOMSE = SCOMSE/FLOAT(COUNT)
      ACRMSE = SCRMSE/FLOAT(COUNT)
      ACGMSE = SCGMSE/FLOAT(COUNT)
C********************** SQUARE MSE **************
      SQCOMSE = SQCOMSE + COMSE(K)**2
      SQCRMSE = SQCRMSE + CRMSE(K)**2
      SQCGMSE = SQCGMSE + CGMSE(K)**2
C********************** VARIANCE MSE *************
      VCOMSE = (SQCOMSE-(COUNT*(ACOMSE**2)))/FLOAT(COUNT-1)
      VCRMSE = (SQCRMSE-(COUNT*(ACRMSE**2)))/FLOAT(COUNT-1)
      VCGMSE = (SQCGMSE-(COUNT*(ACGMSE**2)))/FLOAT(COUNT-1)
C********************** STANDARD DIVATION MSE**************
      SDCOMSE = DSQRT(VCOMSE)
      SDCRMSE = DSQRT(VCRMSE)
      SDCGMSE = DSQRT(VCGMSE)
C*********************************************************
      PCOR = ((ACRMSE-ACOMSE)/ACOMSE)*100
      PCOG = ((ACGMSE-ACOMSE)/ACOMSE)*100
      PCRO = ((ACOMSE-ACRMSE)/ACRMSE)*100
      PCRG = ((ACGMSE-ACRMSE)/ACRMSE)*100
      PCGO = ((ACOMSE-ACGMSE)/ACGMSE)*100
      PCGR = ((ACRMSE-ACGMSE)/ACGMSE)*100
C*********************************************************
 150 CONTINUE
      WRITE(6,160) ACOMSE,ACRMSE,ACGMSE
 160 FORMAT(2X,'AV OLS= ',F20.10,2X,'AV RID= ',F20.10
     *,2X,'AV GAR= ',F20.10 )
      WRITE(6,165) SDCOMSE,SDCRMSE,SDCGMSE
 165 FORMAT(2X,'SD OLS= ',F20.10,2X,'SD RID= ',F20.10
     * ,2X,'SD GAR= ',F20.10)
      WRITE(6,170) PCOR,PCOG
 170 FORMAT(2X,'MOLS&RID= ',F20.10,2X,'MOLS&GAR=',F20.10)
      WRITE(6,175) PCRO,PCRG
 175 FORMAT(2X,'MRID&OLS= ',F20.10,2X,'MRID&GAR=',F20.10)
      WRITE(6,180) PCGO,PCGR
 180 FORMAT(2X,'MGAR&OLS= ',F20.10,2X,'MGAR&RID=',F20.10)
      ENDIF
```

```
  110 CONTINUE
  40  CONTINUE
   5  CONTINUE
      STOP
      END
```

โปรแกรมหลัก 2 สำหรับการแจกแจงแบบไวบูลล์และลอกนอร์มอล

```
C********** MAIN 2 PROGRAM FOR LOG-NORMAL & WEIBULL ***********
     INTEGER M,N,COUNT,IX
     REAL LOGNOR,WEIBUL,AL,BE
     COMMON /SEED/IX,K1/NO/M1,N/CORR/COR1,COR2,COR3,COR4
     REAL IN(6,6),XTX(6,6),Y(100),XTY(10)
    *,MAXEIG(6),MINEIG(6),MSE,SUM(6),TMEAN(6)
    *,OBETA(6),RBETA(6),GBETA(6)
    *,X(100,6),ERR(100),SXX(100,6),RSING(6,6)
     DOUBLE PRECISION LOMSE(500),LRMSE(500),LGMSE(500)
*,SLOMSE,SLRMSE,SLGMSE
    *,ALOMSE,ALRMSE,ALGMSE
    *,SQLOMSE,SQLRMSE,SQLGMSE
    *,VLOMSE,VLRMSE,VLGMSE
    *,SDLOMSE,SDLRMSE,SDLGMSE
    *,PLOR,PLOG,PLRO,PLRG,PLGO,PLGR
     DOUBLE PRECISION WOMSE(500),WRMSE(500),WGMSE(500)
    *,SWOMSE,SWRMSE,SWGMSE
    *,AWOMSE,AWRMSE,AWGMSE
    *,SQWOMSE,SQWRMSE,SQWGMSE
    *,VWOMSE,VWRMSE,VWGMSE
    *,SDWOMSE,SDWRMSE,SDWGMSE
    *,PWOR,PWOG,PWRO,PWRG,PWGO,PWGR
     OPEN(6,FILE = 'D:\FORTRAN\EIR142.OUT')
C************************** SET VALUES ******************************
     M = 3.0
     AM = 0.0
     SD = SQRT(0.05)
     AL = 1.0
     BE = 5.0
     COUNT= 500
     M1 = M+1
C********************** SET SAMPLE SIZE ***************88
     WRITE(6,3) AM,SD
```

```
 3   FORMAT(2X,' LOG-NORMAL DISTRIBUTION ',2X,'MEAN = ',F5.2
    *,2X,'SD = ',F5.2)
     WRITE(6,4) AL,BE
 4   FORMAT(2X,'WEIBUL DISTRIBUTION ',2X,'SCALE = ',F8.3,
    * 2X,'SHAPE=',F8.3)
     DO 5 ISA=1,4
     GOTO(10,15,20,25),ISA
 10  N=10
     GOTO 35
 15  N=30
     GOTO 35
 20  N=50
     GOTO 35
 25  N=100
c********************** SET  CORRELATION ************************
 35  DO 40 ICO = 1,6
     GOTO(45,50,55,60,65,70),ICO
 45  COR1 = 0.1
     COR2 = 0.01
     COR3 = 0.001
     COR4 = 0.0001
     GOTO 75
 50  COR1 = 0.3
     COR2 = 0.09
     COR3 = 0.027
     COR4 = 0.0081
     GOTO 75
 55  COR1 = 0.5
     COR2 = 0.25
     COR3 = 0.125
     COR4 = 0.0625
     GOTO 75
 60  COR1 = 0.7
     COR2 = 0.49
     COR3 = 0.343
     COR4 = 0.2401
     GOTO 75
 65  COR1 = 0.9
     COR2 = 0.81
     COR3 = 0.729
```

```
      COR4 = 0.6561
      GOTO 75
   70 COR1 = 0.99
      COR2 = 0.9801
      COR3 = 0.9703
      COR4 = 0.9650
      GOTO 75
   75 K1 = 0.0
      IX = 2917142
C*********************SET VALUE EQUAL ZERO ****************
      SLOMSE = 0.0
      SLRMSE = 0.0
      SLGMSE = 0.0
      SQLOMSE = 0.0
      SQLRMSE = 0.0
      SQLGMSE = 0.0
      SWOMSE = 0.0
      SWRMSE = 0.0
      SWGMSE = 0.0
      SQWOMSE = 0.0
      SQWRMSE = 0.0
      SQWGMSE = 0.0
C*************************************************************
       WRITE(6,80) N ,M,COUNT,COR1
   80  FORMAT(/,2X,'SAMPLE SIZES = ',I3,2X,'VARIABLES=',I2
     *,/,2X,'COUNT OF SIMULATION =',I3
     *,2X,'BEGIN CORRELATION =',F4.2)
      CALL SIMX(X,TMEAN,SUM)
      CALL STAND(X,TMEAN,XTX,SXX,SUM)
      CALL EIGEN(XTX,MAXEIG,MINEIG)
      CALL ROPT(XTX,RSING)
C********************** USED CALL  INVERSE  MATRIX **********************
      DO 85 J=1,M1
      DO 85 K=1,M1
      IN(K,J) = XTX(K,J)
      IN(J,K) = XTX(J,K)
   85  CONTINUE
      DO 90 K=1,M1
      IF (IN(K,K)) 90,95,90
   95  WRITE(6,105)
```

```
105 FORMAT('IN(K,K) HAS ZERO ON DIAGONAL')
    STOP
 90  CONTINUE
       CALL INMA(IN)
C*********************************************************
    DO 110 I = 1,2
    IF (I.EQ.1) THEN
    DO 115 K = 1,COUNT
    DO 120 J = 1,N
    ERR(J) = LOGNOR(AM,SD)
120 CONTINUE
    CALL SIMY(Y,SXX,ERR,MAXEIG)
C******************************** OLS ********************
    CALL BOLS(IN,SXX,XTY,Y,OBETA)
    CALL BMSE(OBETA,MAXEIG,MSE)
    LOMSE(K) = MSE
C******************************** RID ********************
    CALL BRID(RSING,XTX,XTY,RBETA)
    CALL BMSE(RBETA,MAXEIG,MSE)
    LRMSE(K) = MSE
C******************************** GARROTE ****************
    CALL BGAR(OBETA,SXX,Y,GBETA)
    CALL BMSE(GBETA,MAXEIG,MSE)
    LGMSE(K) = MSE
C********************** SUM MSE  ***************
    SLOMSE = SLOMSE + LOMSE(K)
    SLRMSE = SLRMSE + LRMSE(K)
    SLGMSE = SLGMSE + LGMSE(K)
C********************** AVERAGE MSE **************
    ALOMSE = SLOMSE/FLOAT(COUNT)
    ALRMSE = SLRMSE/FLOAT(COUNT)
    ALGMSE = SLGMSE/FLOAT(COUNT)
C************************ SQUARE MSE *************
    SQLOMSE = SQLOMSE + LOMSE(K)**2
    SQLRMSE = SQLRMSE + LRMSE(K)**2
    SQLGMSE = SQLGMSE + LGMSE(K)**2
C************************ VARIANCE MSE ************
    VLOMSE = (SQLOMSE-(COUNT*(ALOMSE**2)))/FLOAT(COUNT-1)
    VLRMSE = (SQLRMSE-(COUNT*(ALRMSE**2)))/FLOAT(COUNT-1)
    VLGMSE = (SQLGMSE-(COUNT*(ALGMSE**2)))/FLOAT(COUNT-1)
```

```
C********************** STANDARD DIVATION MSE**************
      SDLOMSE = DSQRT(VLOMSE)
      SDLRMSE = DSQRT(VLRMSE)
      SDLGMSE = DSQRT(VLGMSE)
C********************** COMPARISON ON 3 METHOD ************8
      PLOR = ((ALRMSE-ALOMSE)/ALOMSE)*100
      PLOG = ((ALGMSE-ALOMSE)/ALOMSE)*100
      PLRO = ((ALOMSE-ALRMSE)/ALRMSE)*100
      PLRG = ((ALGMSE-ALRMSE)/ALRMSE)*100
      PLGO = ((ALOMSE-ALGMSE)/ALGMSE)*100
      PLGR = ((ALRMSE-ALGMSE)/ALGMSE)*100
C*********************************************************
 115 CONTINUE
       WRITE(6,125) ALOMSE,ALRMSE,ALGMSE
 125  FORMAT(2X,'AV OLS= ',F20.10,2X,'AV RID= ',F20.10
     * ,2X,'AV GAR= ',F20.10 )
       WRITE(6,130) SQLOMSE,SQLRMSE,SQLGMSE
 130  FORMAT(/,2X,'SQUARE MSE FOR OLS     = ',F30.25
     * ,/,2X,'SQUARE MSE FOR RIDGE   = ',F30.25
      WRITE(6,135) PLOR,PLOG
 135 FORMAT(2X,'MOLS&RID= ',F20.10,2X,'MOLS&GAR= ',F20.10)
      WRITE(6,140) PLRO,PLRG
 140 FORMAT(2X,'MRID&OLS= ',F20.10,2X,'MRID&GAR= ',F20.10)
      WRITE(6,145) PLGO,PLGR
 145 FORMAT(2X,'MGAR&OLS= ',F20.10,2X,'MGAR&RID= ',F20.10)
       ELSE
       DO 150 K = 1,COUNT
      DO 155 J=1,N
      ERR(J) = WEIBUL(AL,BE)
155   CONTINUE
     CALL SIMY(Y,SXX,ERR,MAXEIG)
C*************************** OLS ********************
     CALL BOLS(IN,SXX,XTY,Y,OBETA)
     CALL BMSE(OBETA,MAXEIG,MSE)
     WOMSE(K) = MSE
C*************************** RID ******************
     CALL BRID(RSING,XTX,XTY,RBETA)
     CALL BMSE(RBETA,MAXEIG,MSE)
     WRMSE(K) = MSE
C********************** GARROTE ****************
```

```
      CALL BGAR(OBETA,SXX,Y,GBETA)
      CALL BMSE(GBETA,MAXEIG,MSE)
      WGMSE(K) = MSE
C********************* SUM MSE  ***************
      SWOMSE = SWOMSE + WOMSE(K)
      SWRMSE = SWRMSE + WRMSE(K)
      SWGMSE = SWGMSE + WGMSE(K)
C********************* AVERAGE MSE ***************
      AWOMSE = SWOMSE/FLOAT(COUNT)
      AWRMSE = SWRMSE/FLOAT(COUNT)
      AWGMSE = SWGMSE/FLOAT(COUNT)
C********************* SQUARE MSE **************
      SQWOMSE = SQWOMSE + WOMSE(K)**2
      SQWRMSE = SQWRMSE + WRMSE(K)**2
      SQWGMSE = SQWGMSE + WGMSE(K)**2
C********************* VARIANCE MSE *************
      VWOMSE = (SQWOMSE-(COUNT*(AWOMSE**2)))/FLOAT(COUNT-1)
      VWRMSE = (SQWRMSE-(COUNT*(AWRMSE**2)))/FLOAT(COUNT-1)
      VWGMSE = (SQWGMSE-(COUNT*(AWGMSE**2)))/FLOAT(COUNT-1)
C********************* STANDARD DIVATION MSE**************
      SDWOMSE = DSQRT(VWOMSE)
      SDWRMSE = DSQRT(VWRMSE)
      SDWGMSE = DSQRT(VWGMSE)
C********************  COMPARISON ON 3 METHOD ************
      PWOR = ((AWRMSE-AWOMSE)/AWOMSE)*100
      PWOG = ((AWGMSE-AWOMSE)/AWOMSE)*100
      PWRO = ((AWOMSE-AWRMSE)/AWRMSE)*100
      PWRG = ((AWGMSE-AWRMSE)/AWRMSE)*100
      PWGO = ((AWOMSE-AWGMSE)/AWGMSE)*100
      PWGR = ((AWRMSE-AWGMSE)/AWGMSE)*100
C*********************************************************
 150 CONTINUE
      WRITE(6,160) AWOMSE,AWRMSE,AWGMSE
 160 FORMAT(2X,'AV OLS= ',F20.10,2X,'AV RID= ',F20.10
     * ,2X,'AV GAR= ',F20.10 )
      WRITE(6,165) SDWOMSE,SDWRMSE,SDWGMSE
 165  FORMAT(2X,'SD OLS= ',F20.10,2X,'SD RID= ',F20.10
     * ,2X,'SD GAR= ',F20.10)
      WRITE(6,170) PWOR,PWOG
 170   FORMAT(2X,'MOLS&RID= ',F20.10,2X,'MOLS&GAR= ',F20.10)
```

```
      WRITE(6,175) PWRO,PWRG
 175  FORMAT(2X,'MRID&OLS= ',F20.10,2X,'MRID&GAR= ',F20.10)
      WRITE(6,180) PWGO,PWGR
 180  FORMAT(2X,'MGAR&OLS= ',F20.10,2X,'MGAR&RID= ',F20.10)
     ENDIF
 110 CONTINUE
 40   CONTINUE
  5  CONTINUE
     STOP
```

C**********

โปรแกรมย่อยสำหรับสร้างตัวแปรอิสระให้มีความสัมพันธ์ตามที่กำหนด

```
C*****  SUBROUTINE  FOR  SIMULATE INDEPENDENT VARIABLES ******
     SUBROUTINE SIMX(X,TMEAN,SUM)
     REAL NORMAL
     COMMON /SEED/IX,K1/NO/M1,N/CORR/COR1,COR2,COR3,COR4
     REAL SUM(6),TMEAN(6),X(100,6),Z(100,7)
     DO 10 J = 2,M1+1
     DO 10 I = 1,N
     Z(I,J) = 0.0
     Z(I,J) = NORMAL(0.0,1.0)
 10  CONTINUE
C**************** 3 INDEPENDENT  VARIABLES ********************
     DO 15 J=1,M1
     IF (J.EQ.1) THEN
     DO 20 I = 1,N
     X(I,J) = 1.0
 20  CONTINUE
     ENDIF
     IF (J.GE.2.AND.J.LE.4) THEN
     DO 25 I=1,N
     X(I,J) = SQRT(1-COR1)*Z(I,J)+SQRT(COR2)*Z(I,5)
 25 CONTINUE
     ENDIF
C******************** 5 INDEPENDENT VARIABLES*********************
     IF (J.GT.4) THEN
     DO 30 I=1,N
     X(I,J) = SQRT(1-COR3)*Z(I,J)+SQRT(COR4)*Z(I,7)
```

```
 30 CONTINUE
    ENDIF
 15  CONTINUE

C**********CALL CORRELATION*****************************
    CALL CORRE(X,TMEAN,SUM)
    RETURN
    END
```

C**********

โปรแกรมย่อยสำหรับคำนวณความสัมพันธ์ของตัวแปรอิสระ

```
C********** SUBROUTINE FOR COMPUTE CORRELATION **********
    SUBROUTINE CORRE(X,TMEAN,SUM)
    COMMON /NO/M1,N
    REAL TMEAN(6),SUM(6),X(100,6)
    DOUBLE PRECISION XATXA(100,6),XA(100,6),SSA(6),CORX(6,6)
    DO 30 J=2,M1
    SUM(J) = 0
    DO 35 I=1,N
    SUM(J)=SUM(J)+X(I,J)
 35  CONTINUE
 30  CONTINUE
    DO 40 J=2,M1
    TMEAN(J)=SUM(J)/FLOAT(N)
 40  CONTINUE
    DO 45 J=2,M1
    DO 45 I=1,N
    XA(I,J)=X(I,J)-TMEAN(J)
 45  CONTINUE
    DO 50 J=2,M1
    SSA(J)=0
    DO 55 I=1,N
    SSA(J)=SSA(J)+XA(I,J)*XA(I,J)
 55  CONTINUE
 50  CONTINUE
    DO 65 J=2,M1
```

```
      DO 65 K=2,M1
      XATXA(J,K)= 0.0
      DO 70 I=1,N
      XATXA(J,K) = XATXA(J,K)+XA(I,J)*XA(I,K)
 70   CONTINUE
      XATXA(K,J)=XATXA(J,K)
 65   CONTINUE
      DO 75 J=2,M1
      DO 75 K=2,M1
      CORX(J,K) = XATXA(J,K)/SQRT(SSA(J)*SSA(K))
 75   CONTINUE
      RETURN
      END

C*********************************************************************
```

โปรแกรมย่อยสำหรับปรับตัวแปรอิสระให้อยู่ในรูปมาตรฐาน

```
C******** SUBROUTINE FOR BUILD STANDARDIZE X ***************
      SUBROUTINE STAND(X,TMEAN,XTX,SXX,SUM)
      COMMON /NO/M1,N
      REAL XTX(6,6),TMEAN(6),SUM(6),X(100,6),SXX(100,6)
      DOUBLE PRECISION SQX(6),SDX(6),SUMX,XBAR
      SUMX = 0.0
      DO 5 J = 2,M1
      SUMX = SUMX + SUM(J)
 5    CONTINUE
      XBAR = SUMX/FLOAT(N*(M1-1))
      DO 10 J = 2,M1
      SQX(J) = 0.0
      DO 10 I = 1,N
      SQX(J) = SQX(J) + X(I,J)*X(I,J)
 10   CONTINUE
      DO 20 J = 2,M1
      SDX(J) = DSQRT((SQX(J)-(N*TMEAN(J)**2))/FLOAT(N-1))
 20 CONTINUE
      DO 25 J = 1,M1
      IF(J.EQ.1) THEN
      DO 30 I = 1,N
      SXX(I,J) = 1.0
 30 CONTINUE
      ELSE
```

```
      DO 35 I = 1,N
      SXX(I,J) = (X(I,J)-TMEAN(J))/SDX(J)
 35 CONTINUE
      ENDIF
 25 CONTINUE
      DO 40 J=1,M1
      DO 40 K=1,M1
      XTX(J,K)= 0.0
      DO 45 I=1,N
      XTX(J,K) = XTX(J,K)+SXX(I,J)*SXX(I,K)
 45 CONTINUE
      XTX(K,J)=XTX(J,K)
 40  CONTINUE
      RETURN
      END
C*************************************************************************
```

โปรแกรมย่อยสำหรับค่าเฉพาะที่มากที่สุดและเวกเตอร์ของค่าเฉพาะที่มากที่สุด

```
C********** SUBROUTINE FIND MAX EIGEN VALUE & VECTOR ***********
      SUBROUTINE EIGEN(XTX,MAXEIG,MINEIG)
      COMMON /NO/M1,N
      REAL MAXEIG(6),MINEIG(6),XTX(6,6),LAMDA(6),EIG(6,6)
      REAL MAX,MIN
      CALL EIGE(XTX,LAMDA,EIG)
C***************** FIND MAX EIGENVALUE *****************
      MAX = LAMDA(1)
      DO 5 I = 2,M1
      IF(LAMDA(I).GT.MAX) MAX = LAMDA(I)
  5   CONTINUE
      DO 10 L = 1,M1
      IF (LAMDA(L).EQ.MAX) THEN
      DO 15 I = 1,M1
      MAXEIG(I) = EIG(I,1)
 15   CONTINUE
       ENDIF
 10   CONTINUE
C***************** FIND MIN EIGENVALUE *****************
      MIN = LAMDA(1)
      DO 20 I = 2,M1
      IF(LAMDA(I).LT.MIN) MIN = LAMDA(I)
 20   CONTINUE
```

```
      DO 25 L = 1,M1
      IF (LAMDA(L).EQ.MIN) THEN
      DO 30 I = 1,M1
      MINEIG(I) = EIG(I,3)
 30   CONTINUE
     ENDIF
 25  CONTINUE
      RETURN
      END

C*********************************************************************
```

โปรแกรมย่อยสำหรับค่าเฉพาะและเวกเตอร์ของค่าเฉพาะ

```
C************  SUBROUTINE FIND EIGEN VALUE & VECTOR  **********
      SUBROUTINE EIGE(XTX,LAMDA,EIG)
      REAL XTX(6,6),LAMDA(6),EIG(6,6)
      DOUBLE PRECISION AX(6,6),EI(6),ATA(6,6)
      REAL MAX,MIN
      COMMON /NO/M1,N
      NB = 1.0
      NK = 5.0
      ERR = 0.000001
      DO 5 I=1,M1
      DO 5 J=1,M1
      ATA(I,J) = XTX(I,J)
 5    CONTINUE
      DO 10 I=1,M1
      DO 10 J=1,M1
      IF (I.EQ.J) THEN
      AX(I,J) = 1.0
      ELSE
      AX(I,J) = 0.0
      ENDIF
 10   CONTINUE
      DO 15 NB = 1,NK
      MD = M1-1
      DO 20 I=1,MD
      MI = I+1
      DO 20 J=MI,M1
      XA = 0.0
      IF (DABS(ATA(I,J)).GT.ABS(XA)) THEN
```

```
      I1 = I
      J1 = J
      XA = ATA(I,J)
      ENDIF
      S1 = 0.0
      S2 = 0.0
      S3 = 0.0
      S5 = 0.0
      S8 = 0.0
      IF (XA.NE.0) THEN
       S1 = ATA(I1,I1)-ATA(J1,J1)
       S2 = ABS(S1)/2.0
       S3 = DSQRT(S2**2+ATA(I1,J1)**2)
       S4 = SQRT(2.0)*SQRT(1+S2/S3)
       S5 = 1/S4
       S6 = ATA(I1,J1)
       S7 = 2*S3*S5
       S8 = S6/S7
        IF(S1.LT.0) THEN
        S8 = (-1)*S8
      ENDIF
      ENDIF
      DO  25 K = 1,M1
       IF ((K.NE.I1).AND.(K.NE.J1)) THEN
       T1 = ATA(I1,K)*S5 + ATA(J1,K)*S8
       T2 = ATA(J1,K)*S5 - ATA(I1,K)*S8
       ATA(I1,K) = T1
       ATA(J1,K) = T2
       ENDIF
25    CONTINUE
       T1 = ATA(I1,I1)*S5**2 + 2*ATA(I1,J1)*S5*S8 + ATA(J1,J1)*S8**2
       T2 = ATA(I1,I1)*S8**2 - 2*ATA(I1,J1)*S5*S8 + ATA(J1,J1)*S5**2
      ATA(I1,I1) = T1
      ATA(J1,J1) = T2
      ATA(I1,J1) = 0.0
      ATA(J1,I1) = 0.0
      DO 30 L = 1,M1
      ATA(L,I1) = ATA(I1,L)
      ATA(L,J1) = ATA(J1,L)
      T1 = AX(L,I1)*S5 + AX(L,J1)*S8
```

```
      T2 = AX(L,J1)*S5 - AX(L,I1)*S8
      AX(L,I1) = T1
      AX(L,J1) = T2
   30 CONTINUE
      IF((NB.GE.NK).OR.(ABS(XA).LE.ERR)) GOTO 35
   20 CONTINUE
      GOTO 15
   15  CONTINUE
   35  MAX = 0.0
      MIN = 9E9
      DO 40 I = 1,M1
      EI(I) = ATA(I,I)
       IF(EI(I).GE.MAX) THEN
       MAX = EI(I)
       PMAX = I
       ENDIF
       IF(EI(I).LE.MIN) THEN
       MIN = EI(I)
       PMIN = I
       ENDIF
   10   CONTINUE
       DO 45 I = 1,M1
      LAMDA(I) = ATA(I,I)
   45  CONTINUE
      DO 50 J=1,M1
      IF ((J.EQ.PMAX).OR.(J.EQ.PMIN)) THEN
      DO 55 I = 1,M1
      EIG(I,1) = AX(I,PMAX)
   55  CONTINUE
      DO 60 I = 1,M1
      EIG(I,3) = AX(I,PMIN)
   60  CONTINUE
      ENDIF
   50 CONTINUE
      RETURN
      END
```

C**********

โปรแกรมย่อยสำหรับคำนวณเมทริกซ์ผกผัน

C********* SUBROUTINE FOR COMPUTE INVERSE MATRIX ******

```
      SUBROUTINE INMA(IN)
      COMMON /NO/M1,N
      REAL  IN(6,6)
      DO 5  K =1,M1
      IN(K,K)= -1.0/IN(K,K)
      DO 10 I=1,M1
       IF (I-K) 15,10,15
 15  IN(I,K) = -IN(I,K)*IN(K,K)
 10  CONTINUE
      DO 30 I=1,M1
      DO 30 J=1,M1
      IF ((I-K)*(J-K)) 35,30,35
 35  IN(I,J) = IN(I,J)-IN(I,K)*IN(K,J)
 30  CONTINUE
      DO 5 J=1,M1
      IF (J-K) 40,5,40
 40  IN(K,J) = -IN(K,J)*IN(K,K)
 5    CONTINUE
      DO 45 I=1,M1
      DO 45 J=1,M1
      IN(I,J)=-IN(I,J)
 45  CONTINUE
       RETURN
       END
```

C**********

โปรแกรมย่อยสำหรับหาค่า k สำหรับใช้ในสมการการถดถอยริดจ์

```
C***********      SUBROUTINE FIND  K FOR RIDGE METHOD **************
      SUBROUTINE ROPT(XTX,RSING)
      REAL XTX(6,6),RSING(6,6),IN(6,6),A,OPT
      DOUBLE PRECISION  SING(6,6),LAM(6,6)
     *,MULXTX(6,6),TRACE(500)
      COMMON /NO/M1,N
      DO 5  A = 0.001,30,0.001
      DO 10 I = 1,M1
      DO 10 J = 1,M1
      IF (I.EQ.J) THEN
      SING(I,J) = A
      ELSE
      SING(I,J) = 0.0
```

```
      ENDIF
10    CONTINUE
      DO 15 I = 1,M1
      DO 15 J = 1,M1
      LAM(I,J) = XTX(I,J)+SING(I,J)
15    CONTINUE
      DO 20 J=1,M1
      DO 20 K=1,M1
      IN(K,J) = LAM(K,J)
      IN(J,K) = LAM(J,K)
20    CONTINUE
      DO 25 K=1,M1
      IF (IN(K,K)) 25,30,25
30    WRITE(6,35)
35    FORMAT('IN(K,K) HAS ZERO ON DIAGONAL')
      STOP
25    CONTINUE
     CALL INMA(IN)
      DO 40 I = 1,M1
      DO 40 J = 1,M1
      MULXTX(I,J) = 0.0
      DO 45 K = 1,M1
      MULXTX(I,J) = MULXTX(I,J) + XTX(K,I)*IN(K,J)
45    CONTINUE
40    CONTINUE
      TRACE(A) = 0.0
      DO 50 I = 1,M1
      DO 50 J = 1,M1
      IF ((I-J).EQ.0) THEN
      TRACE(A) = TRACE(A) + MULXTX(I,J)
      ENDIF
50    CONTINUE
      IF (ABS(TRACE(A)-(M1-1)).LE.0.001) THEN
      OPT = A
      GOTO 55
      ENDIF
5     CONTINUE
55    DO 60 I = 1,M1
      DO 60 J = 1,M1
      IF (I.EQ.J) THEN
```

```
      RSING(I,J) = OPT
      ELSE
      RSING(I,J) = 0.0
      ENDIF
   60 CONTINUE
      RETURN
      END

C*********************************************************************
```

โปรแกรมย่อยสำหรับสร้างตัวแปรตาม

```
C******** SUBROUTINE FOR BUILD DEPENDENT VARIABLE *********
      SUBROUTINE SIMY(Y,SXX,ERR,MAXEIG)
      COMMON /NO/M1,N
      REAL Y(100),MAXEIG(6),ERR(100),SXX(100,6)
      DOUBLE PRECISION TOY(100)
      DO 5 I = 1,N
      TOY(I) = 0.0
      DO 10 J = 1,M1
      TOY(I) = TOY(I) + SXX(I,J)*MAXEIG(J)
   10 CONTINUE
    5 CONTINUE
      DO 15 J = 1,N
      Y(J) = TOY(J) + ERR(J)
   15 CONTINUE
      RETURN
      END

C*********************************************************************
```

โปรแกรมย่อยสำหรับหาตัวประมาณสัมประสิทธิ์จากวิธีกำลังสองน้อยสุด

```
C******SUBROUTINE FOR BUILD BETA FROM OLS METHODS ****************
      SUBROUTINE BOLS(IN,SXX,XTY,Y,OBETA)
      COMMON /NO/M1,N
      REAL IN(6,6),Y(100),OBETA(6)
     *,XTY(10),SXX(100,6)
      DOUBLE PRECISION SXXT(6,100)
      DO 5 J = 1,M1
      DO 5 I = 1,N
      SXXT(J,I) = SXX(I,J)
    5 CONTINUE
      DO 10 J = 1,M1
```

```
      XTY(J) = 0.0
      DO 15 I = 1,N
      XTY(J) = XTY(J)+ SXXT(J,I)*Y(I)
15    CONTINUE
10    CONTINUE
      DO 20 J = 1,M1
      OBETA(J) = 0.0
      DO 25 I = 1,M1
      OBETA(J) = OBETA(J) + IN(J,I)*XTY(I)
25    CONTINUE
20    CONTINUE
      RETURN
      END

C*********************************************************************
```

โปรแกรมย่อยสำหรับหาตัวประมาณสัมประสิทธิ์จากสมการถดถอยริดจ์

```
C****** SUBROUTINE FOR BUILD BETA FROM RIDGE METHOD **************
      SUBROUTINE BRID(RSING,XTX,XTY,RBETA)
      REAL XTX(6,6),RSING(6,6),XTY(10),RBETA(6),IN(6,6)
      DOUBLE PRECISION  RLAM(6,6)
      COMMON /NO/M1,N
      DO 5 I=1,M1
      DO 5 J=1,M1
      RLAM(I,J) = XTX(I,J) + RSING(I,J)
 5    CONTINUE
      DO 10 J=1,M1
      DO 10 K=1,M1
      IN(K,J) = RLAM(K,J)
      IN(J,K) = RLAM(J,K)
10    CONTINUE
      DO 15 K=1,M1
      IF (IN(K,K)) 15,20,15
 20   WRITE(6,25)
 25   FORMAT('IN(K,K) HAS ZERO ON DIAGONAL')
      STOP
 15   CONTINUE
      CALL INMA(IN)
      DO 30 I = 1,M1
      RBETA(I) = 0.0
      DO 35 J = 1,M1
```

```
      RBETA(I) = RBETA(I) + IN(I,J)*XTY(J)
   35 CONTINUE
   30 CONTINUE
      RETURN
      END

C**********************************************************************
```

โปรแกรมย่อยสำหรับหาตัวประมาณสัมประสิทธิ์จากสมการถดถอยเชิงเส้นการ์ล็อต

```
C****  SUBROUTINE FOR  BUILD BETA FROM GARROTE METHOD *******
      SUBROUTINE BGAR(OBETA,SXX,Y,GBETA)
      COMMON /NO/M1,N
      REAL GBETA(6),OBETA(6),SXX(100,6),Y(100),IN(6,6)
      DOUBLE PRECISION AJX(100,6),AJXTAJX(6,6),AJXT(6,100)
     *,AJXTY(6),C(6)
      DO 5 I = 1,M1
      DO 5 J = 1,N
      AJX(J,I) = OBETA(I)*SXX(J,I)
    5  CONTINUE
      DO 10 I = 1,M1
      DO 10 J = 1,M1
      AJXTAJX(I,J) = 0.0
      DO 15 K = 1,N
      AJXTAJX(I,J) = AJXTAJX(I,J) + AJX(K,I)*AJX(K,J)
   15 CONTINUE
   10 CONTINUE
      DO 20 J=1,M1
      DO 20 K=1,M1
      IN(K,J) = AJXTAJX(K,J)
      IN(J,K) = AJXTAJX(J,K)
   20  CONTINUE
      DO 25 K=1,M1
      IF (IN(K,K)) 25,30,25
   25 WRITE(6,35)
   35 FORMAT('IN(K,K) HAS ZERO ON DIAGONAL')
      STOP
   25 CONTINUE
      CALL INMA(IN)
      DO 40 J = 1,M1
      DO 40 I = 1,N
      AJXT(J,I) = AJX(I,J)
```

```
 40 CONTINUE
    DO 45 J = 1,M1
    AJXTY(J) = 0.0
    DO 50 I = 1,N
    AJXTY(J) = AJXTY(J) + AJXT(J,I)*Y(I)
 50 CONTINUE
 45 CONTINUE
    DO 55 J = 1,M1
    C(J) = 0.0
    DO 60 I = 1,M1
    C(J) = C(J) + IN(J,I)*AJXTY(I)
 60 CONTINUE
 55 CONTINUE
    DO 65 I = 1,M1
    GBETA(I) = C(I)*OBETA(I)
65  CONTINUE
    RETURN
    END
```

C**********

โปรแกรมย่อยสำหรับหาค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง

```
C*************** SUBROUTINE FIND MSE ************************
    SUBROUTINE BMSE(BETA,MAXEIG,MSE)
    COMMON /NO/M1,N
    REAL MAXEIG(6),BETA(6),MSE
    MSE = 0.0
    DO 5 I = 1,M1
    MSE = MSE + (BETA(I)-MAXEIG(I))**2
 5  CONTINUE
    RETURN
    END
```

C**********

โปรแกรมย่อยสร้างเลขสุ่มให้มีการแจกแจงสม่ำเสมอ

```
C************ FUNCTION  FOR  SIMULATE FROM UNIFORM***********
    FUNCTION RANDU(IX)
    IX = IX*16807
    IF (IX.LT.0) IX=IX+2147483647+1
    RANDU = IX
    RANDU = RANDU/2147483647
```

```
      RETURN
      END

C**************************************************************************
```

โปรแกรมย่อยสร้างเลขสุ่มให้มีการแจกแจงแบบปกติ

```
C********* FUNCTION FOR  SIMULATE FROM NORMAL***************
      REAL FUNCTION NORMAL(AM,SD)
      COMMON /SEED/IX,K1
      REAL PI,U1,U2,NOR1,NOR2,AM,SD
      PI=3.14159265
      IF (K1.EQ.1) GOTO 5
        U1 = RANDU(IX)
        U2 = RANDU(IX)
        NOR1 = SQRT(-2*ALOG(U1))*COS(2*PI*U2)
        NOR2 = SQRT(-2*ALOG(U1))*SIN(2*PI*U2)
      NORMAL = NOR1*SD+AM
      K1=1
      RETURN
5     NORMAL = NOR2*SD+AM
      K1=0
      RETURN
      END


C**************************************************************************
```

โปรแกรมย่อยสร้างเลขสุ่มให้มีการแจกแจงแบบปกติปลอมปน

```
C********* FUNCTION FOR  SIMULATE FROM  SCALE  NORMAL*********
      REAL FUNCTION SCNOR(C,P,AM,SD)
      REAL NORMAL,ADSD,C,P,AM,SD,R1
      COMMON /SEED/IX,K1
      ADSD = C*SD
      R1 = RANDU(IX)
      IF(R1.LE.P) THEN
      SCNOR = NORMAL(AM,ADSD)
      ELSE
      SCNOR = NORMAL(AM,SD)
      ENDIF
      RETURN
      END
```

```
C*************************************************************************
```
โปรแกรมย่อยสร้างเลขสุ่มให้มีการแจกแจงแบบลอกนอร์มอล
```
C********* FUNCTION FOR  SIMULATE FROM  LOGNORMAL***********
      REAL FUNCTION LOGNOR(AM,SD)
      REAL NORMAL
      COMMON /SEED/IX,K1
      LOGNOR = EXP(NORMAL(AM,SD))
      RETURN
      END

C*************************************************************************
```
โปรแกรมย่อยสร้างเลขสุ่มให้มีการแจกแจงแบบไวบูลล์
```
C********* FUNCTION FOR  SIMULATE FROM  WEIBULL  *************
      REAL FUNCTION WEIBUL(AL,BE)
      COMMON /SEED/IX,K1
      REAL R2,AL,BE
      R2 = RANDU(IX)
      WEIBUL = AL*(-ALOG(1.0-R2))**(1.0/BE)
      RETURN
      END
```

ประวัติผู้วิจัย

นางสาวอัชฌา อระวีพร เกิดวันที่ 7 มีนาคม พ.ศ. 2518 สำเร็จการศึกษาปริญญาวิทยาศาสตรบัณฑิต(วท.บ.) สาขาสถิติประยุกต์ ภาควิชาสถิติประยุกต์ คณะวิทยาศาสตร์ประยุกต์ จากสถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ในปีการศึกษา 2538 และเข้าศึกษาต่อในหลักสูตรสถิติศาสตรมหาบัณฑิต ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์-มหาวิทยาลัย ในปีการศึกษา 2539