บรรณานุกรม

หนังสือ

Abraham, Farid F. and Tilier, William A. in An introduction to computer simulation in applied science Plenum Press, New York, 1972.

Rubinstein, Reuwen Y. in Simulation and the monte carlo method. New York. John Wiley & Son, 1981.

Franklin, Lehman, Richard S.and College, Marchall in Computer simulation and modelling pp. 152-162, Lawrence erlbaum associates publishers, 1977.

IBM Application Program System/360 Scientific Subroutine Package (360. A-cx-03x) Version III.

Morrison, Donald F. in Multivariate Statistical Methods. 2nd ed., McGraw-Hill, Kogakusha, 1978.

วารสาร

Afifi, A.A. and Elashoff, R.M. "Missing observations in multivariate statistics I. Review of the literature" Journal of American Standard Association 61 (1966): 596-604.

_____. "Missing obsservations in multivariate statistics II. Point estimation in simple linear regression" Journal of American Standard Association 62 (1967) : 10-29.

_____. "Missing observations in multivariate statistic III. Large sample analysis of simple linear regression" Journal of American Standard Association 64 (1969) : 337-358.

_____. "Missing observations in multivariate stantistics IV. A note on simple linear regression" Journal of American Standard Association 64 (1969) : 359-365.

Chan, Linda S. and Dunn, Olive Jean. "The treatment of missing values in discriminant analysis I. The sampling experiment" Journal of American Standard Association 67 (1972) : 473-477.

_____. "A note on the asymtotic aspect of the treatment of missing values in discriminant analysis" Journal of American Standard Association 69 (1974) : 672-673.

Chan, Linda S., Gilman, June AONG and Dunn, Olive Jean. "Alternative approaches to missing values in discriminant analysis" Journal of American Standard Association 71 (1976) : 842-844.

Jackson, Esther C. "Missing values in linear multiple discriminant analysis" Biometrics 24 (1968) : 835-843.

Scheuer, Ernest M. and Stoller, David S. "On the generation of normal random variate" Technometrics, 4 (1962): 278-281.

วิทยานิพนธ์

ยะไมพร ธรรมวัฒน์ไพศาล "วิธีการประมาณค่าที่ยาดหายไปในการวิเคราะห์การถดถอย" วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติบัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย 2522

ภาคผนวก

โปรแกรม SPSSX ทำการคำนวณและทดสอบผลของปัจจัยต่าง ๆ ที่มีต่อค่าความคลาดเคลื่อนเฉลี่ยของแต่ละวิธี

```
UNNUMBERED
DATA LIST/ N 2 P 4 RHO 6 MEAN 11-19 (5) REGR 21-29 (5)
          MREGR 31-39 (5) FACT  41-49 (5)
MANOVA MEAN REGR MREGR FACT BY N (1-5) P (1-3) RHO (1-3) /
          ANALYSIS/
          DESIGN = N P RHO N BY P N BY RHO P BY RHO
```

โปรแกรม SPSSX ทำการคำนวณและทดสอบความแตกต่างของค่าความคลาดเคลื่อนเฉลี่ยทั้ง 4 วิธี

```
UNNUMBERED
DATA LIST / METHOD 1 BLOCK 2-4 MSE 11-20 (5)
MANOVA MSE BY METHOD (1-4) BLOCK (1-106) /
          DESIGN = BLOCK METHOD
```

โปรแกรมหลักสำหรับคำนวณค่าความคลาดเคลื่อนเฉลี่ยของวิธีการประมาณค่าสูญหายทั้ง 4 วิธี

```
C
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C:::                                                                        :::
C:::  THESIS    : THE COMPARATIVE OF MISSING VALUE ESTIMATION METHODS       :::
C:::              IN MULTIVARIATE ANALYSIS                                  :::
C:::                                                                        :::
C:::  AUTHOR    : MISS PORNSIRI MUENCHAISRI  B622345                       :::
C:::                                                                        :::
C:::  DEPARTMENT: STATISTICS                                               :::
C:::                                                                        :::
C:::  FACULTY   : COMMERCE AND ACCOUNTING                                  :::
C:::                                                                        :::
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
C     VARIABLES                    DESCRIPTION
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
C     M              NUMBER OF VARIABLES
C                      M = 3 5 7 10
C     N              NUMBER OF SAMPLE
C                      N = 30 50 70 100 200
C     E              RHO : COEFFICIENT OF CORRELATION
C                      RHO = .1 .2 .3 .4 .5 .6 .7 .8 .9
C     ISEED          INITAILIZE BY 973253
C     MM             PERCENT OF MISSING VALUE IN EACH VARIABLE
C     A1(I,J)        MULTIVARIATE DATA IN VARIABLE I CASE J
C     AA(I,J)        COMPLETE DATA AFTER MISSING
C     IPLAG(I)       SIGN : NUMBER OF MISSING IN CASE I
C     PB(I,J)        SIGN : 1   DATA IS MISS
C                           0   DATA IS PRESENT
C     MISS(I)
C     ICOUNT(I)      NUMBER OF I+1 MISSING IN EACH SAMPLE
C     SM(I,J)        WORKING AREA
C     SM1(I,J)       WORKING AREA
C     SM2(I,J)       WORKING AREA
C     MIS(I,J)       THE POSITION OF MISSING VALUE IN EACH VARIABLE
C     ANS            TOTAL OF MEAN SQUARE ERROR OF EACH METHOD
C     NCOUNT         TATAL COUNT IN EACH RANK IN EACH METHOD
C     GG             WORKING AREA
C     IWANT          WORKING AREA
C     TOTAL          TOTAL OF WEIGHTED SCORE OF EACH METHOD
C     Y              THE MEAN OF EACH VARIABLE EQUAL 0
C     G              WORKING AREA
C     KNUM           NUMBER OF MISSING VALUES IN EACH SIMULATION
C
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      REAL MM
      DIMENSION Y(15),R(15,15),A1(15,300),G(15),
     *AA(15,300),IPLAG(300),PB(15,300),MISS(300),ICOUNT(15),
     *SM(15,300),SM1(15,300),SM2(15,300),MIS(15,300),ET(4),
     *ANS1(15),NCOUNT(4,4),GG(4),IWANT(4),TOTAL(15),RE(15),CV(15)
```

```
1     FORMAT(I2,F2.1,2X,F2.1,2X,I3)
2     FORMAT(I5,F5.2,I5)
3     FORMAT(2X,10F9.5)
C5     FORMAT(' RESULT ',F20.8)
C6     FORMAT(' ************ SIMULATION STAGE ************',I5)
7     FORMAT(' ANS   ',F20.8)
8     FORMAT(' FREQ.',4I8)
9     FORMAT(' TOTAL ',F20.8)
      READ(5,1) M,MM,E,N
      ISEED = 973253
      F = 0.0
      F = F + E
      DO 36 L1=1,M
      DO 36 L2=1,M
      R(L1,L2) = 1.0
      IF(L1.EQ.L2) GO TO 36
      R(L1,L2) = F
      R(L2,L1) = R(L1,L2)
36    CONTINUE
      DO 37 IZ=1,10
      Y(IZ) = 0.
      TOTAL(IZ) = 0.0
      RE(IZ) = 0.0
37    ANS1(IZ) = 0.
      TTCV = 0.0
      DO 38 IR=1,4
      DO 38 IS=1,4
38    NCOUNT(IR,IS) = 0
C
      DO 200 IP=1,100
C
C-----------------------------------------------------------------------
C            GENERATE THE MULTIVARIATE NORMAL
C-----------------------------------------------------------------------
C
      CALL MULNOR(Y,R,M,N,A1,ISEED,IDX)
      IF(IDK.EQ.0) GO TO 1000
      NUM = 0
C
C-----------------------------------------------------------------------
C           FIND THE POSITION OF MISSING VALUE IN EACH VARIABLE
C-----------------------------------------------------------------------
C
      CALL MISSNG(M,N,MM,A1,AA,IPLAG,MISS,NN,PB,ICOUNT,ISEED)
C
C-----------------------------------------------------------------------
C   CALCULATE THE SUM OF ABSOLUTE VALUE OF THE DIFFERENCE OF THE TRUE
C             VALUE AND THE VALUE ESTIMATE BY THE MEANS
C-----------------------------------------------------------------------
C
      CALL MEAN(M,N,NN,A1,AA,IPLAG,MISS,DIFF,SM,PB,AS,KNUM,TCV)
      TTCV = TTCV + TCV
      MN = N - ICOUNT(M + 1)
      NUM = NUM + 1
      G(NUM) = DIFF / KNUM
      GG(NUM) = G(NUM)
```

```
      RE(NUM) = RE(NUM) + AS
C
C.....
C   CALCULATE THE SUM OF ABSOLUTE VALUE OF THE DIFFERENCE OF THE TRUE
C             VALUE AND THE VALUE ESTIMATE BY REGRESSION METHOD
C.....
C
      CALL REGR(A1,AA,M,NN,DIFF,MISS,IPLAG,SM1,N,PB,ICOUNT,NUM,AS,IRK)
      IF(IRK.EQ.1) GO TO 1000
      NUM = NUM + 1
      G(NUM) = DIFF / KNUM
      GG(NUM) = G(NUM)
      RE(NUM) = RE(NUM) + AS / KNUM
C
C.....
C   CALCULATE THE SUM OF ABSOLUTE VALUE OF THE DIFFERENCE OF THE TRUE
C             VALUE AND THE VALUE ESTIMATE BY REGRESSION WITH
C             REPLACE THE MISSING BY ITS MEANS BEFORE
C.....
C
      CALL REGR(A1,SM,M,MN,DIFF,MISS,IPLAG,SM2,N,PB,ICOUNT,NUM,AS,IRK)
      IF(IRK.EQ.1) GO TO 1000
      NUM = NUM + 1
      G(NUM) = DIFF / KNUM
      GG(NUM) = G(NUM)
      RE(NUM) = RE(NUM) + AS / KNUM
C
C.....
C   CALCULATE THE SUM OF ABSOLUTE VALUE OF THE DIFFERENCE OF THE TRUE
C          VALUE AND THE VALUE ESTIMATE BY PRINCIPAL COMPONENT METHOD
C.....
C
C
      CALL FACT(M,A1,NN,MISS,IPLAG,PB,DIFF,SM,ICOUNT,N,AS,IRK)
      IF(IRK.EQ.1) GO TO 1000
      NUM = NUM + 1
      G(NUM) = DIFF / KNUM
      GG(NUM) = G(NUM)
      RE(NUM) = RE(NUM) + AS / KNUM
C
C.....
C   FIND THE FREQUENCY OF EACH METHOD IN EACH RANK
C        1. SORT THE SUM OF ABSOLUTE VALUE IN EACH STAGE
C        2. COUNT STEP 1 WHEN THE SUM EQUAL IN EACH RANK
C.....
C
      NUM1 = NUM - 1
      DO 40 IR=1,NUM1
      IS = IR + 1
      DO 40 IT=IS,NUM
      IF(GG(IR).LE.GG(IT)) GO TO 40
      GGG = GG(IR)
      GG(IR) = GG(IT)
      GG(IT) = GGG
40    CONTINUE
C
```

```
      DO 60 IR=1,NUM
      DO 50 IS=1,NUM
      IF(G(IR).EQ.GG(IS)) IWANT(IR) = IS
50    CONTINUE
      NCOUNT(IR,IWANT(IR)) = NCOUNT(IR,IWANT(IR)) + 1
60    CONTINUE
      CALL RPTGEN(G,NUM,ANS1)
200   CONTINUE
      DO 80 I=1,NUM
80    ET(I) = RE(I) / 1000
      ACV = TTCV / 1000
      DO 250 I=1,NUM
      DO 250 II=1,NUM
250   TOTAL(I) = TOTAL(I) + (NUM + 1 - II) * NCOUNT(I,II)
      WRITE(6,2) M,MM,N
      DO 360 L1=1,M
360   WRITE(6,3) (R(L1,L2),L2=1,M)
      DO 998 I=1,NUM
998   WRITE(6,7) ANS1(I)
      DO 300 IR=1,NUM
300   WRITE(6,8) (NCOUNT(IR,IS),IS=1,NUM)
      DO 450 J=1,NUM
450   WRITE(6,9) TOTAL(J)
      WRITE(6,500) (ET(I),I=1,NUM)
      WRITE(6,600) ACV
600   FORMAT(' AVERAGE C.V. ',F20.8)
500   FORMAT(' % ERROR ',4F15.8)
      WRITE(6,199) ISEED
199   FORMAT(' ISEED ',I20)
1000  STOP
      END
```

โปรแกรมย่อยสำหรับสร้างตัวเลขสุ่มให้มีการแจกแจงแบบปกติ $N(0 , \Sigma)$

```
C*************************************************************************
C***                                                                   ***
C***        GENERATE MULTIVARIATE NORMAL DISTRIBUTION                  ***
C***                                                                   ***
C*************************************************************************
C
      SUBROUTINE MULNOR(XBAR,R,M,N,SCORES,ISEED,IDK)
      DIMENSION XBAR(15),R(15,15),SCORES(15,300),DD(15,300)
     *,ZVECT(15),SD(15)
C
      CALL SETUP(IDK,DD,M,XBAR,SD,R)
C
C  IF IDK = 0  :SINGULAR METRIX OR SQJAR ROOT OF NEGATIVE VALUE
C     IDK = 1  :SUBROUTINE SETUP COULD FIND TRIANGULAR METRIX
C
      IF ( IDK .EQ. 0 ) GOTO 8000
      DO 1000 I=1,N
      DO 10 L=1,M
      CALL NORMAL(XX,ISEED)
      ZVECT(L) = XX
   10 CONTINUE
      DO 1200 J=1,M
      SCORES(J,I) = 0.0
      DO 1300 K=1,J
      SCORES(J,I) = SCORES(J,I) + DD(J,K) * ZVECT(K)
 1300 CONTINUE
      SCORES(J,I) = SCORES(J,I) + XBAR(J)
 1200 CONTINUE
 1000 CONTINUE
      GO TO 9000
 8000 WRITE(6,8888)
 8888 FORMAT(5X,'*****CANNOT CONTINUE...YOU MUST INPUT NEW DATA******')
 9000 RETURN
      END
```

โปรแกรมย่อยสำหรับสร้างตัวเลขสุ่มให้มีการแจกแจงสม่ำเสมอ U(0 , 1)

```
C
C
C  ***********************************************************************
C  *****                                                             *****
C  *****                    SUBROUTINE UNIFORM                       *****
C  *****                                                             *****
C  ***********************************************************************
C
C
      SUBROUTINE RANDU(IX,IY,YFL)
C
C
      IY = IX * 65539
      IF (IY) 5 , 6, 6
    5 IY = IY +2147483647 + 1
    6 YFL = IY
      YFL = YFL * .4656613E-9
      RETURN
      END
```

โปรแกรมย่อยสำหรับสร้างตัวเลขสุ่มให้มีการแจกแจงแบบปกติ N(0 , 1)

```
C
C
C  ***********************************************************************
C  *****                                                             *****
C  *****                  SUBROUTINE NORMAL                          *****
C  *****                                                             *****
C  ***********************************************************************
C
C
      SUBROUTINE NORMAL(XX,IX)
      A = 0.0
      DO 400 I=1,12
      CALL RANDU(IX,IY,Y)
      IX = IY
  400 A = A + Y
      XX = A - 6.0
      RETURN
      END
```

โปรแกรมย่อยสำหรับคำนวณ TRIANGULAR METRIX

```
C
C  ***********************************************************************
C  *****                                                             *****
C  *****                  SUBROUTINE SETUP                           *****
C  *****                                                             *****
C  ***********************************************************************
C
C                SUBROUTINE TO FIND TRIANGULAR METRIX
C
          SUBROUTINE SETUP(IDK,D,M,XBAR,SD,R)
      DIMENSION XBAR(15),SD(15),R(15,15),D(15,15)
C  CONVERT TO VARIANCE/COVARIANCE MATRIX IF NECESSERY
C     IF (R(II,1,1) .EQ. 1.0) GOTO 150
C
C  WE HAVE A VARIANCE/COVARIANCE MATRIX. COMPUTE SDS AND STORE
C  DUPLICATE MATRIX FOR DETERMINANT TEST.
C
      DO 101 I=I,M
      SD(I) = SQRT(R(I,I))
      DO 101 J=I,M
      D(I,J) = R(I,J)
  101 CONTINUE
C     GO TO 250
C
C   WE HAVE CORRELATIONS. CONVERT TO VARIANCE/COVARIANCE MATRIX
C    AND STORE DUPLICATE MATRIX
C
C 150 DO 200 I=1,M
C     DO 200 J=1,M
C     R(I,J) = R(I,J) * SD(I) *SD(J)
C     R(J,I) = R(I,J)
C     D(I,J) = R(I,J)
C     D(J,I) = R(J,I)
C 200 CONTINUE
C
C   CHECK FOR SINGULAR BY COMPUTING DETERMINANT. PIVOTAL CONDENSATION
C  ALGORITHM USED.
C
C    IF THE DETERMINANT IS LESS THAN 0.0001, A ERROR RETURN FOR
C     SINGULARLITY IS MADE
C
  250 NEXT = 2
      IWHAT = 1
  252 DO 251 I=NEXT,M
      OVERD = D(I,IWHAT) / D(IWHAT,IWHAT)
      DO 251 J=NEXT,M
      D(I,J) = D(I,J) - D(IWHAT,J) * OVERD
  251 CONTINUE
      IF (NEXT .GE. M) GO TO 260
      IWHAT = NEXT
      NEXT = NEXT  + 1
      GO TO 252
  260 DET = 1.0
```

```
      DO 261 I=1,M
  261 DET = DET* D(I,I)
      DET = ABS(DET)
      IF(DET .LT. 0.0001) GOTO 500
C
C   COMPUTE TRIANGULAR FACTORIZATION
C
  300 IF(R(1,1) .LE. 0.0) R(1,1) = 1.0
      ROOT = SQRT(R(1,1))
      DO 301 I=1,M
      D(I,1) = R(I,1) / ROOT
      DO 301 J=2,M
      D(I,J) = 0.0
  301 CONTINUE
      DO 310 I=2,M
      SUM = 0.0
      KLIMT = I-1
      DO 311 K=1,KLIMT
      SUM = SUM + D(I,K) * D(I,K)
  311 CONTINUE
      DIFF = R(I,I) - SUM
      IF(DIFF .LE. 0.0) GOTO 501
      DIFF = ABS(DIFF)
      D(I,I) = SQRT(DIFF)
      DO 312 J=2,I
      IF(I .EQ. I) GOTO 312
      SUM1 = 0.0
      KLIMT = J-1
      DO 313 K=1,KLIMT
      SUM1 = SUM1 + D(I,K) * D(J,K)
  313 CONTINUE
      D(I,J) = (R(I,J) - SUM1) / D(J,J)
  312 CONTINUE
  310 CONTINUE
C
C   NOW IT SHOW THAT DETERMINAT IS EXIST
C
      IDK = 1
      RETURN
C
C   ERROR RETURN
C
  501 WRITE(6,112)
  112 FORMAT(3X,'****** CANNOT FIND TRIANGULAR METRIX DIFF < 0.0******')
      GO TO 502
  500 WRITE(6,111)
  111 FORMAT(3X,'****** DETERMINANT IS LESS THAN 0.0001 ******',/)
  502 IDK = 0
      RETURN
      END
```

โปรแกรมย่อยสำหรับหา INVERSE

```
C  ***********************************************************************
C  ****                                                               ****
C  ****                SUBROUTINE INVERSION                           ****
C  ****                                                               ****
C  ***********************************************************************
C
      SUBROUTINE MINV(A,N,D,L,M)
      DIMENSION A(1),L(1),M(1)
C
      D = 1.0
      NK = -N
      DO 80 K=1,N
      NK = NK + N
      L(K) = K
      M(K) = K
      KK = NK + K
      BIGA = A(KK)
      DO 20 J = K,N
      IZ = N * (J-1)
      DO 20 I = K,N
      IJ = IZ + I
10    IF(ABS(BIGA)-ABS(A(IJ))) 15,20,20
15    BIGA = A(IJ)
      L(K) = I
      M(K) = J
20    CONTINUE
C
C    INTERCHANGE ROWS
C
      J = L(K)
      IF (J-K) 35,35,25
25    KI = K-N
      DO 30 I =1,N
      KI = KI + N
      HOLD = -A(KI)
      JI = KI - K + J
      A(KI) = A(JI)
30    A(JI) = HOLD
C
C     INTERCHANGE COLUMNS
C
35    I = M(K)
      IF ( I-K) 45,45,38
38    JP = N * (I-1)
      DO 40 J =1,N
      JK = NK + J
      JI = JP + J
      HOLD = - A(JK)
      A(JK) = A(JI)
40    A(JI) = HOLD
C
C    DIVIDED COLUMN BY MINUS PIVOT(VALUE OF PIVOT SYSTEM IS CONTAINED IN
C     BIGA)
```

```
C
45      IF (BIGA) 48,46,48
46      D = 0.0
        RETURN
48      DO 55 I=1,N
        IF (I-K) 50,55,50
50      IK = NK + I
        PPP = A(IK)  / (-1 * BIGA)
        A(IK) = PPP
55      CONTINUE
C
C     RECODE MATRIX
C
        DO 65 I=1,N
        IK = NK + I
        HOLD = A(IK)
        IJ = I - N
        DO 65 J =1,N
        IJ = IJ + N
        IF (I-K) 60,55,60
60      IF (J-K) 62,65,62
62      KJ = IJ -I + K
        A(IJ) = HOLD * A(KJ) +A(IJ)
65      CONTINUE
C
C     DIVIDED ROW BY PIVOT
C
        KJ = K - N
        DO 75 J=1,N
        KJ =  KJ + N
        IF(J-K) 70,75,70
70      A(KJ) = A(KJ) / BIGA
75      CONTINUE
C
C     PRODUCT OF PIVOT
C
        D = D * BIGA
C
C     REPLACE PIVOT BY RECIPOCAL
C
        A(KK) = 1.0 / BIGA
80      CONTINUE
C
C    FINAL ROW AND COLUMN INTERCHANGE
C
        K = N
100     K = (K-1)
        IF (K) 150,150,105
105     I = L(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 = A(JK)
        JI = JR + J
```

ต้นฉบับไม่มีหน้า 92

NO PAGE 92 IN ORIGINAL

โปรแกรมย่อยสำหรับเช็คตำแหน่งข้อมูลที่สูญหายไม่ให้ซ้ำกัน

```
C
C  ***********************************************************************
C  ****                                                               ****
C  ****          SUBROUTINE TO FIND EQUAL RANDOM NUMBER               ****
C  ****                                                               ****
C  ***********************************************************************
C
      SUBROUTINE DUPL(N,M,MMM,IADD,P,IX)
      DIMENSION IADD(15,300),KOUNT(15)
5     CALL SORT(N,M,MMM,IADD,IX)
      DO 200 I=1,M
      KOUNT(I) = 0
      DO 100 L1 =1,MMM
      II = 1
10    LL1 = L1 + II
      IF(LL1.GT.MMM) GO TO 200
      IF (IADD(I,L1).GT.IADD(I,LL1)) GO TO 100
      IF (IADD(I,L1).LT.IADD(I,LL1)) GO TO 100
C
      CALL RANDU(IX,IY,Y)
      IX = IY
      IYY = INT(Y*P)
      IF(IYY.GT.N.OR.IYY.LE.0) GO TO 10
      IADD(I,LL1) = IYY
      KOUNT(I) = KOUNT(I) + 1
      II = II + 1
      GO TO 10
100   CONTINUE
200   CONTINUE
      DO 400 I=1,M
      IF(KOUNT(I).GT.0) GO TO 5
400   CONTINUE
      RETURN
      END
```

โปรแกรมย่อยสำหรับ เรียงตำแหน่งสูญหายจากน้อยไปหามาก

```
C
C   ***********************************************************************
C   ****                                                               ****
C   ****          SUBROUTINE ROUTINE TO SORT RANDOM NUMBER             ****
C   ****                                                               ****
C   ***********************************************************************
C
      SUBROUTINE SORT(N,M,MMM,IADD, IX)
      DIMENSION IADD(15,300)
      MMM2 = MMM - 1
      DO 300 I=1,M
      DO 200 L1=1,MMM2
      L3 = L1 + 1
      DO 100 L2 = L3,MMM
      IF(IADD(I,L1).LE.IADD(I,L2))GO TO 100
      IP = IADD(I,L1)
      IADD(I,L1) = IADD(I,L2)
      IADD(I,L2) = IP
100   CONTINUE
200   CONTINUE
300   CONTINUE
      RETURN
      END
```

โปรแกรมย่อยสำหรับกำหนดตำแหน่งสูญหายของข้อมูล

```
C
C   ***********************************************************************
C   ****                                                               ****
C   ****      SUBROUTINE TO FIND THE POSITION OF THE MISSING VALUE     ****
C   ****                                                               ****
C   ***********************************************************************
C
      SUBROUTINE MISSNG(M,N,MM1,A1,AA,IPLAG,MISS,NN,PB,ICOUNT,IX)
      REAL MM1
      DIMENSION A1(15,300),AA(15,300),IPLAG(300),MISS(300),PB(15,300),IC
     *OUNT(15),IB(15,300)
      DO 1902 K2=1,10
1902  ICOUNT(K2) = 0
      MMM =(MM1 * N) + 1
      P = 100.
      IF (N.GT.100) P =1000.
      DO 211 I=1,M
      DO 111 J=1,MMM
150   CALL RANDU(IX,IY,Y)
      IX=IY
      IYY = INT(P * Y)
      IF(IYY.GT.N.OR.IYY.LE.0) GO TO 150
      IB(I,J) = IYY
111   CONTINUE
211   CONTINUE
      CALL DUPLIN,M,MMM,IB,P,IX)
C
      DO 350 I=1,M
      DO 350 J=1,N
      PB(I,J) = 0.
350   CONTINUE
      DO  360 I=1,M
      DO 360 J=1,MMM
      PB(I,IB(I,J)) = 1.0
360   CONTINUE
      M1 = M / 2
      DO 400 J=1,N
      JJ = 0
      DO 390 I=1,M
      IF(PB(I,J) .EQ. 1.0) JJ = JJ + 1
390   CONTINUE
      IPLAG(J) = JJ
      ICOUNT(JJ+1) = ICOUNT(JJ+1) + 1
400   CONTINUE
      NN = 0
      DO 450 J=1,N
      IF( IPLAG(J).NE.0) GO TO 450
      NN = NN + 1
      DO 420 I=1,M
      AA(I,NN)= A1(I,J)
420   CONTINUE
450   CONTINUE
      RETURN
```

โปรแกรมย่อยสำหรับคำนวณค่าความคลาดเคลื่อนเฉลี่ยโดยวิธีที่ใช้ค่าเฉลี่ย

```
C
C   ***********************************************************************
C   ****                                                               ****
C   ****                    SUBROUTINE MEAN                            ****
C   ****                                                               ****
C   ***********************************************************************
C
      SUBROUTINE MEAN(M,N,NN,A1,AA,IPLAG,MISS,DIFF,SM,PB,AS,KNUM,TCV)
      DIMENSION A1(15,300),AA(15,300),IPLAG(300),MISS(300),STD(15),
     *PB(15,300),J1(15),SUM(15),SUM1(15),AV(15),SM(15,300),CV(15)
      AS = 0.0
      KNUM = 0
      COUNT = 0
      TCV = 0.0
      DO  100 I = 1,N
      IF(IPLAG(I).NE.M) GO TO 100
      COUNT = COUNT + 1
100   CONTINUE
      DO 500 I=1,M
      SUM(I) = 0.
      SUM1(I) = 0.
      JJ = 0
      DO 400 J=1,N
      IF(IPLAG(J).EQ.M) GO TO 400
      IF(PB(I,J).EQ.1.0) GO TO 400
      SUM(I) = SUM(I) + A1(I,J)
      SUM1(I) = SUM1(I) + A1(I,J) * A1(I,J)
      JJ = JJ + 1
400   CONTINUE
      KNUM = KNUM + JJ
      AV(I) = SUM(I) / JJ
      STD(I) = SQRT(((SUM1(I)) - SUM(I) * SUM(I) / JJ) / (JJ - 1))
      CV(I) = ABS(STD(I) / AV(I))
      TCV = TCV + CV(I)
500   CONTINUE
      TCV = TCV / M
      DIFF = 0.
      DO 600 I=1,M
      J1(I) = 0
      DO 550 J=1,N
      IF(IPLAG(J).EQ.M) GO TO 550
      J1(I) = J1(I) + 1
      SM(I,J1(I)) = A1(I,J)
      IF(PB(I,J).EQ.0.) GO TO 550
      SM(I,J1(I)) = AV(I)
      PER = ABS(A1(I,J) - AV(I)) * 100 / AES(A1(I,J))
      DIFF = DIFF + (A1(I,J) - AV(I))**2
      AS = AS + PER
550   CONTINUE
600   CONTINUE
      KNUM = M * N - KNUM - M * COUNT
      AS = AS / KNUM
      RETURN
```

โปรแกรมย่อยสำหรับคำนวณค่าความคลาดเคลื่อนเฉลี่ยโดยวิธีวิเคราะห์ความถดถอยพหุเชิงเส้นและวิธีวิเคราะห์ความถดถอยพหุเชิงเส้นตัดแปลง

```
C
C  ***********************************************************************
C  ****                                                               ****
C  ****                    SUBROUTINE REGRESSION                      ****
C  ****                                                               ****
C  ***********************************************************************
C
      SUBROUTINE REGR(A1,AA,M,NN,DIFF,MISS,IPLAG,SM1,N,PB,ICOUNT,NUM,AS,
     *IRK)
      DIMENSION AA(15,300),IPLAG(300),MISS(300),MIS(15,300),
     *A1(15,300),SM1(15,300),AMS(15,300),BB(15,15),PB(15,300),ICOUNT(15)
     *,ISAVE1(15,15),AAA(15,300),RR(15,15),SM2(15,300)
C
      AS = 0.0
      K = M - 1
      NS = M
      JCOUNT = 0
      JKOUNT = 0
      NN = ICOUNT(1)
      IF(NUM.EQ.1) GO TO 10
      NN = N - ICOUNT(M + 1)
10    DO 20 I=1,NN
      DO 20 J=1,M
20    SM1(J,I) = AA(J,I)
C-----------------------------------------------------------------------
      M1 = M - 1
      DO 1000 I=1,M1
      IIII = 0
50    CALL REGRSS(SM1,M,NN,NS,AMS,BB,ISAVE1,K,RR,SM2)
      IIII = IIII + 1
      IF(IIII.GT.50) GO TO 1111
      IF(NUM.EQ.2) GO TO 600
      IF(JKOUNT.NE.0) GO TO 600
      IIP = 0
      JJJ = 0
      DO 500 J=1,N
      IF(IPLAG(J).GE.M) GO TO 500
      IF(IPLAG(J).LE.(-1) GO TO 200
      IF(IPLAG(J).NE.I) GO TO 500
      IIP = IIP + 1
      DO 100 IS=1,M
      SM1(IS,IIP) = A1(IS,J)
      IF(PB(IS,J).EQ.1.0) SM1(IS,IIP) = 0.0
100   CONTINUE
      GO TO 500
200   JJJ = JJJ + 1
      IIP = IIP + 1
      DO 300 IR=1,M
      SM1(IR,IIP) = SM2(IR,JJJ)
300   CONTINUE
500   CONTINUE
      NN = IIP
600   IIP =0
```

```
      DO 900 J=1,N
      I1 = 0
      IF(IPLAG(J).GE.M) GO TO 900
      IF(NUM.EQ.2) GO TO 700
      IF(IPLAG(J).GT.I) GO TO 900
700   IIP = IIP + 1
      IF(IPLAG(J).NE.I) GO TO 900
      DO 750 J2=1,M
      IF(PB(J2,J).EQ.0.0) GO TO 750
      I1 = I1 + 1
      MIS(I1,IIP) = J2
750   CONTINUE
      DO 850 I2=1,I
      DO 800 J2=1,M
      IF(ISAVE1(J2,M).EQ.MIS(I2,IIP)) JII = J2
800   CONTINUE
      SUM = AMS(JII,1)
      DO 830 JJ=1,K
      L = ISAVE1(JII,JJ)
830   SUM = SUM + SM1(L,IIP) * BB(JII,JJ)
      SUBTOT = ABS(SUM - SM1(ISAVE1(JII,M),IIP))
      SM1(ISAVE1(JII,M),IIP) = SUM
      IF(SUBTOT .GE. 0.1) JCOUNT = JCOUNT + 1
850   CONTINUE
900   CONTINUE
      JKOUNT = JCOUNT
      IF(JCOUNT.EQ.0) GO TO 1000
      JCOUNT = 0
      GO TO 50
1000  CONTINUE
1010  DIFF = 0.
      JJJ = 0
      DO 1600 J=1,N
      IF(IPLAG(J).EQ.M) GO TO 1600
      JJJ = JJJ + 1
      DO 1550 JJ = 1,M
      PER = ABS(SM1(JJ,JJJ) - A1(JJ,J)) * 100 /ABS(A1(JJ,J))
      AS = AS + PER
      DIFF = DIFF + (SM1(JJ,JJJ) - A1(JJ,J)) ** 2
1550  CONTINUE
1600  CONTINUE
      IRK = 0
      GO TO 2222
1111  WRITE(6,1212)
1212  FORMAT(' ITERATE MORE THAN 10 ITERATION ')
      IRK = 1
2222  RETURN
      END
```

โปรแกรมย่อยสำหรับหาส่มการความถดถอยไปประมาณค่าสูญหาย

```
C
C  ***********************************************************************
C  ****                                                               ****
C  ****                    SUBROUTINE REGRESS                         ****
C  ****                                                               ****
C  ***********************************************************************
C
      SUBROUTINE REGRSS(AA,M,NN,NS,AMS,BB,ISAVE1,K,RR,AAA)
      DIMENSION AA(15,300),ANS(15),BB(15),ISAVE(15),SB(15)
     *,RR(15,15),R(300),BB(15,15),ISAVE1(15,15),AAA(15,300),
     *AMS(15,15),XBAR(15),STD(15),D(15),T(15),RX(300),RY(300)
      ID = 0
      X = 0.0
      DO 101 J=1,NN
      DO 101 I=1,M
101   AAA(I,J) = AA(I,J)
      NNNN = NN
      CALL CORRE(NNNN,M,ID,X,XBAR,STD,RX,R,D,B,T,AA)
C
C     TEST NUMBER OF SELECTIONS
C
      DO 10 JP=1,M
10    ISAVE(JP) = JP
C
      DO 290 I=1,NS
      IF(I.EQ.1) GO TO 291
      IEE = ISAVE(1)
      MP = M - 1
      DO 11 IE=1,MP
11    ISAVE(IE) = ISAVE(IE + 1)
      ISAVE(M) = IEE
C
C   NRESI...OPTION CODE FOR TABLE OF RESIDUALS
C           0..IF IT IS NOT DESIRED
C           1..IF IT IS DESIRED
C   NDEP....DEPENDENT VARIABLE
C   K.......NUMBER OF INDEPENDENT VARIABLES INCLUDE
C   ISAVE...A VECTOR CONTAINING THE INDEPENDENT VARIABLE INCLUDE
C
291   CALL ORDER(M,R,K,ISAVE,RX,RY)
C
      CALL MINV(RX,K,DET,B,T)
C
C   TEST SINGULARITY OF THE MATRIX INVERTED
C
C     IF (DET) 112,110,112
C10   WRITE(6,14)
C     GO TO 200
112   CALL MULTR(NNNN,K,XBAR,STD,D,RX,RY,ISAVE,B,SB,T,ANS)
      DO 250 JJ=1,M
      BB(I,JJ) = B(JJ)
      ISAVE1(I,JJ) = ISAVE(JJ)
250   CONTINUE
```

```
      DO 280 JP=1,10
      AMS(I,JP) = ANS(JP)
280   CONTINUE
290   CONTINUE
      RETURN
      END
```

โปรแกรมย่อยสำหรับหาค่าสัมประสิทธิ์ความถดถอย

```
C
C  ***********************************************************************
C  ****                                                               ****
C  ****                    SUBROUTINE MULTR                           ****
C  ****                                                               ****
C  ***********************************************************************
C
      SUBROUTINE MULTR(N,K,XBAR,STD,D,RX,RY,ISAVE,B,SB,T,ANS)
      DIMENSION XBAR(15),STD(15),D(15),RX(300),RY(300),ISAVE(15),B(15)
      DIMENSION ANS(15),SB(15),T(15)
C
      MM = K + 1
C
C     BETA WEIGHTS
C
      DO 100 J=1,K
100   B(J) = 0.0
      DO 110 J=1,K
      L1 = K * (J-1)
      DO 110 I=1,K
      L = L1 + I
110   B(J) = B(J) + RY(I) * RX(L)
      RM = 0.0
      BO = 0.0
      L1 = ISAVE(MM)
C
C  COEFFICIENT OF DETERMINATION
C
      DO 120 I=1,K
      RM = RM + B(I) * RY(I)
C
C   REGRESSION COEFFICIENT
C
      L = ISAVE(I)
      B(I) = B(I) * (STD(L1) / STD(L))
C
C     INTERCEPT
C
120   BO = BO + B(I) * XBAR(L)
      BO = XBAR(L1) - BO
C
C  SUM OF SQUARES ATTRIBUTLE TO REGRESSION
C
      SSAR = RM * D(L1)
C
C  MULTIPLE CORRELATION COEFFICIENT
C
122   RM = SQRT(ABS(RM))
C
C  SUM OF SQUARES OF DEVIATIONS FROM REGRESSION
C
      SSDR = D(L1) - SSAR
C
```

```
C     VARIANCEOF ESTIMATE
C
       FN = N - K -1
       SY = SSDR / FN
C
C     STANDARD DEVIATIONS OF REGRESSION COEFFICIENTS
C
       DO 130 J=1,K
       L1 = K * (J-1) + J
       L = ISAVE(J)
128    SB(J) = SQRT(ABS((RX(L1)/D(L)) * SY))
C
C   COMPUTE  T-VALUE
C130     T(J) = 0. 0
130    T(J) = B(J) / SB(J)
C130    WRITE(6,992)D(L)
C992    FORMAT(' IN MULTR D(L) IN LINE 128 IS ',F20.3)
C
C
C    STANDARD ERROR OF ESTIMATE
C
135    SY = SQRT(ABS(SY))
C
C     F-VALUE
C
       FK = K
       SSARM = SSAR / FK
       SSDRM = SSDR / FN
       F = SSARM / SSDRM
C
       ANS(1) = BO
       ANS(2) = RM
       ANS(3) = SY
       ANS(4) = SSAR
       ANS(5) =FK
       ANS(6) = SSARM
       ANS(7) = SSDR
       ANS(8) = FN
       ANS(9) = SSDRM
       ANS(10) = F
C       WRITE(6,11111) (ANS(I),I=1,10)
C11111 FORMAT(1X,5F12.8)
       RETURN
       END
```

โปรแกรมย่อยสำหรับคำนวณเมตริกซ์ความสัมพันธ์

```
C
C  ***********************************************************************
C  ****                                                               ****
C  ****                    SUBROUTINE CORRE                           ****
C  ****                                                               ****
C  ***********************************************************************
C
      SUBROUTINE CORRE(NN1,M,ID,X,XBAR,STD,RX,R,B,C,T,AAA)
      DIMENSION XBAR(15),STD(15),RX(300),R(300),B(15),D(15),T(15)
     *,AAA(15,300)
C    INITIALIZATION
C
      N = NN1
      DO 100 J12=1,M
      B(J12) = 0.0
100   T(J12) = 0.0
      K = (M*M+M) / 2
      DO 102 I=1,K
102   R(I) = 0.0
      FN = N
      L = 0
C
127   IF (N-M) 130,130,135
130   KK = N
      GO TO 137
135   KK = M
137   DO 140 I=1,KK
C     CALL DATA(M,D)
      DO 99 J=1,M
99    D(J) = AAA(J,I)
      DO 140 J=1,M
      T(J) = T(J) + D(J)
      L = L + 1
140   RX(L) = D(J)
      PKK = KK
      IPKK = PKK
      DO 150 J=1,M
      XBAR(J) = T(J)
150   T(J) = T(J) / PKK
C
C  CALCULATE SUMS OF CROS PRODUCTS OF DEVIATIONS
C  FROM TEMPORARY MEANS FOR M OBSERVATIONS
C
      L = 0
      DO 190 I=1,KK
      JK = 0
      DO 170 J=1,M
      L = L + 1
 170  D(J) = RX(L) - T(J)
      DO 180 J=1,M
      B(J) = B(J) + D(J)
      DO 180 K=1,J
      JK = JK + 1
```

```
180   R(JK) = R(JK) + D(J) * D(K)
C
      IF (N-KK) 205,205,185
C
C   READ THE REST OF OBSERVATIONS ONE AT A TIME ,SUM THE OBSERVATIONS
C   AND CALCULATE SUMS OF CROSS-PRODUCTS OF DEVIATIONS FROM TEMPORARY
C   MEANS
C
185   KK = N - KK
      DO 200 I=1,KK
      JK = 0
C     CALL DATA(M,D)
      DO 999 J=1,M
999   D(J) = AAA(J,I+IPKK)
      DO 190 J=1,M
      XBAR(J) = XBAR(J) + D(J)
      D(J) = D(J) - T(J)
190   B(J) = B(J) + D(J)
      DO 200 J=1,M
      DO 200 K=1,J
      JK = JK + 1
200   R(JK) = R(JK) + D(J) * D(K)
      MMMM = IPKK + KK
C     WRITE(6,1000) (W(I),I=1,N)
C1000 FORMAT(' W(I)',3F15.5/)
C
C   CALCULATE MEANS
C
205   JK = 0
      DO 210 J = 1,M
      XBAR(J) = XBAR(J) / FN
C
C    ADJUST SUMS OF CROSS-PRODUCT OF DEVIATIONS FROM TEMPORARY MEANS
C
      DO 210 K=1,J
      JK = JK + 1
210   R(JK) = R(JK) - B(J) * B(K) / FN
C
C    CALCULATE CORELATION  COEFFICIENTS
C
      JK = 0
      DO 220 J=1,M
      JK = JK + J
220   STD(J) = SQRT(ABS(R(JK)))
C
C
      DO 230 J=1,M
      DO 230 K=J,M
      JK = J + (K*K-K) / 2
      L = M * (J-1) + K
      RX(L) = R(JK)
      L = M * (K-1) + J
      RX(L) = R(JK)
      IF (STD(J)*STD(K)) 225,222,225
222   R(JK) = 0.0
      GO TO 230
```

```
225   R(JK) = R(JK) / (STD(J) * STD(K))
230   CONTINUE
C
C   CALCULATE STANDARD DEVIATIONS
C
      FN = SQRT(FN-1.0)
      DO 240 J=1,M
240   STD(J) = STD(J) / FN
C
C   COPY THE DIAGONAL OF THE MATRIX OF SUM OF CROSS-PRODUCTS OF
C   DEVIATIONS FROM MEANS
C
      L = -M
      DO 250 I=1,M
      L = L + M + 1
250   B(I) = RX(L)
      RETURN
      END
```

โปรแกรมย่อยสำหรับคัดลอกเมตริกซ์ความสัมพันธ์

```
C
C
C  ********************************************************************
C  ****                                                            ****
C  ****                SUBROUTINE ORDER                            ****
C  ****                                                            ****
C  ********************************************************************
C
      SUBROUTINE ORDER(M,R,K,ISAVE,RX,RY)
      DIMENSION R(300),ISAVE(15),RX(300),RY(300)
C
      MM = 0
      NDEP = ISAVE(M)
      DO 130 J=1,K
      L2 = ISAVE(J)
      IF (NDEP-L2) 122,123,123
122   L = NDEP + (L2*L2-L2) / 2
      GO TO 125
123   L = L2 + (NDEP*NDEP-NDEP) / 2
125   RY(J) = R(L)
C
C  COPY A SUBSET MATRIX INTERCORRELATIONS AMONG INDEPENDENT VARIABLES
C
      DO 130 I=1,K
      L1 = ISAVE(I)
      IF ( L1-L2 ) 127,128,128
127   L = L1 + (L2*L2-L2) /2
      GO TO 129
128   L = L2 + (L1*L1 -L1) / 2
129   MM = MM + 1
130   RX(MM) = R(L)
C
C   PLACE THE SUBSCRIPT NUMBER OF THE DEPENDENT VARIABLES IN ISAVE(K+1)
C
C     ISAVE(K+1) = NDEP
      RETURN
      END
```

โปรแกรมย่อยสำหรับคำนวณ cumulative percentage eigen value

```
C
C   ***********************************************************************
C   ****                                                               ****
C   ****                    SUBROUTINE TRACE                           ****
C   ****                                                               ****
C   ***********************************************************************
C
      SUBROUTINE TRACE(M, R, CON, K, D)
      DIMENSION R(1),D(1)
C
      FM = M
      L = 0
      DO 100 I=1,M
      L = L + I
100   D(I) = R(L)
      K = 0
C
C TEST WHETHER I-TH EIGENVALUE IS GREATER THAN OR EQUAL TO THE CONSTANT
C
      DO 110 I=1,M
      IF(D(I)-CON) 120,105,105
105   K = K+ 1
110   D(I) = D(I) /FM
C
C  COMPUTE CUMULATIVE PERCENTAGE OF EIGENVALUES
C
120   DO 130 I=2,K
130   D(I) = D(I) + D(I-1)
      RETURN
      END
```

โปรแกรมย่อยสำหรับคำนวณค่าความคลาดเคลื่อนเฉลี่ยโดยวิธีวิเคราะห์ส่วนประกอบหลัก

```
C
C   ********************************************************************
C   ****                                                            ****
C   ****            SUBROUTINE FACTOR ANALYSIS                      ****
C   ****                                                            ****
C   ********************************************************************
C
      SUBROUTINE FACT(M,A1,NN,MISS,IPLAG,PB,DIFF,SM3,ICOUNT,N,AS,IKK)
      DIMENSION A1(15,300),IPLAG(300),PB(15,300),
     *STDZ(15,300),SS(15),S1(15),IT(15),
     *STD1(15),SM3(15,300),MISS(300),
     *EGEN(15),ICOUNT(15),AV(15),MIS(15,300)
      AS = 0.0
      CON = 1
      JCOUNT = 0
      JKOUNT = 0
      JJ = 0
      DO 20 J=1,N
      IF(IPLAG(J) .EQ.M) GO TO 20
      JJ = JJ + 1
      DO 10 I=1,M
      SM3(I,JJ) = A1(I,J)
      IF(PB(I,J).EQ.1.0) SM3(I,JJ) = 0.0
10    CONTINUE
20    CONTINUE
C.....................................................................
      M1 = M - 1
      DO 1000 I1=1,M1
      IIII = 0
50    IIP = 0
      IIII = IIII + 1
      IF(IIII.GT.50) GO TO 2000
      DO 650 I=1,M
      SS(I) = 0.0
      S1(I) = 0.0
      IT(I) = 0
      IKK = 0
      DO 600 J=1,N
      IF(IPLAG(J).EQ.M) GO TO 600
      IKK = IKK + 1
      IF(IPLAG(J).LT.I1) GO TO 610
      IF(JKOUNT.EQ.0) GO TO 611
      IF(IPLAG(J).EQ.I1) GO TO 610
611   IF(PB(I,J).EQ.1.0) GO TO 600
610   IT(I) = IT(I) + 1
      S1(I) = S1(I) + SM3(I,IKK) * SM3(I,IKK)
      SS(I) = SS(I) + SM3(I,IKK)
600   CONTINUE
      AV(I) = SS(I) / IT(I)
      STD1(I) = SQRT(((S1(I)) - SS(I) * SS(I) / IT(I)) / (IT(I) - 1))
650   CONTINUE
      DO 680 I=1,M
      JJ = 0
```

```
      DO 660 J=1,N
      IF(IPLAG(J).EQ.M) GO TO 660
      JJ = JJ + 1
      IF(IPLAG(J).LT.I1) GO TO 655
      IF(IPLAG(J).GT.I1) GO TO 651
      IF(JKOUNT.NE.0) GO TO 655
651   STDZ(I,JJ) = 0.0
      IF(PB(I,J).EQ.1.0) GO TO 660
655   STDZ(I,JJ) = (SM3(I,JJ) - AV(I)) / STD1(I)
660   CONTINUE
680   CONTINUE
C
      IC = ICOUNT( M + 1 )
      CALL FACTOR(STDZ,N,M,COM,EGEN,IC)
      IIP = 0
      IU = 0
      DO 900 J=1,N
      IF(IPLAG(J).GE.M) GO TO 900
      IIP = IIP + 1
      IF(IPLAG(J).NE.I1) GO TO 900
      I5 = 0
      DO 700 J2=1,M
      IF(PB(J2,J).EQ.0.0) GO TO 700
      I5 = I5 + 1
      MIS(I5,IIP) = J2
700   CONTINUE
      IU = IU + 1
      DO 850 I2=1,I1
      SAS = 0.0
      DO 800 I6=1,M
      SAS = SAS + EGEN(I6) * STDZ(I6,IIP)
800   CONTINUE
      JII = MIS(I2,IIP)
      FMISS = SAS * EGEN(JII)
      TMISS = FMISS * STD1(JII) + AV(JII)
      SUBTCT = ABS(SM3(MIS(I2,IIP),IIP) - TMISS)
      SM3(MIS(I2,IIP),IIP) = TMISS
      IF(SUBTCT.GE. 0.1) JCOUNT = JCOUNT + 1
850   CONTINUE
900   CONTINUE
      JKOUNT = JCOUNT
      IF(JCOUNT.EQ.0) GO TO 1000
      JCOUNT = 0
      GO TO 50
1000  CONTINUE
      DIFF = 0.0
      JJJ = 0
      DO 1600 J=1,N
      IF(IPLAG(J).EQ.M) GO TO 1600
      JJJ = JJJ + 1
      DO 1550 JJ =1,M
      PER = ABS(SM3(JJ,JJJ) - A1(JJ,J)) * 100 / ABS(A1(JJ,J))
      AS = AS + PER
      DIFF = DIFF + (SM3(JJ,JJJ) - A1(JJ,J))** 2
1550  CONTINUE
1600  CONTINUE
```

```
      IRK = 0
      GO TO 3000
 2000 WRITE(6,1999)
 1999 FORMAT(' ITERATE IN FACT MORE THAN 10 ITERATION ')
      IRK =.1
 3000 RETURN
      END
```

โปรแกรมย่อยสำหรับคำนวณค่าสัมประสิทธิ์ของคอมโปเนนท์แรกของการวิเคราะห์ส่วนประกอบหลัก

```
C
C  ***********************************************************************
C  ****                                                               ****
C  ****                  SUBROUTINE FACTOR                            ****
C  ****                                                               ****
C  ***********************************************************************
C
C
      SUBROUTINE FACTOR(STDZ,N,M,CON,EGEN, IC)
      DIMENSION STDZ(15,300),EGEN(15),V(300)
     *,STD(15),B(15),D(15),S(15),T(15),XBAR(15),R(300),TV(300)
     *,RX(300)
      ID =0
      X = 0.0
      NNN = N - IC
      CALL CORRE(NNN,M,ID,X,XBAR,STD,RX,R,D,B,T,STDZ)
      MV = 0
      CALL EIGEN(R,V,M,MV)
      CALL TRACE(M,R,CON,K,0)
      DO 30 L=1,M
      EGEN(L) = V(L)
30    CONTINUE
      RETURN
      END
```

โปรแกรมย่อยสำหรับคำนวณค่า Eigen

```
C
C  ***********************************************************************
C  ****                                                               ****
C  ****                    SUBROUTINE EIGEN                           ****
C  ****                                                               ****
C  ***********************************************************************
C
      SUBROUTINE EIGEN(A,R,M,MV)
      DIMENSION A(1),R(1)
      N = M
C
C   GENERATE IDENTITY MATRIX
C
      RANGE = 1.0 E-6
      IF(MV-1) 10,25,10
10    IQ = - N
      DO 20 J=1,N
      IQ = IQ + N
      DO 20 I=1,N
      IJ = IQ + I
      R(IJ) = 0.0
      IF(I-J) 20,15,20
15    R(IJ) = 1.0
20    CONTINUE
C
C  COMPUTE INITIAL AND FINAL NORMS(ANORM AND ANORMX)
C
25    ANORM = 0.0
      DO 35 I=1,N
      DO 35 J=1,N
      IF (I-J) 30,35,30
30    IA = I + (J*J-J) / 2
      ANORM = ANORM + A(IA) * A(IA)
35    CONTINUE
      IF (ANORM) 165,165,40
40    ANORM = 1.414 *SQRT(ANORM)
      ANORMX = ANORM * RANGE / FLOAT(N)
C
C  INITIALIZE INDICATEORS AND COMPUTE THRESHOLD, THR
C
      IND = 0
      THR = ANORM
45    THR = THR / FLOAT(N)
50    L = 1
55    M = L + 1
C
C  COMPUTE SIN AND COS
C
60    MQ = (M*M-M) / 2
      LQ = (L*L-L) / 2
      LM = L + MQ
62    IF (ABS(A(LM)) - THR) 130,65,65
65    IND = 1
```

```
      LL = L + LQ
      MM = M + MQ
      X = 0.5 * (A(LL) -A(MM))
68    Y = -A(LM) / SQRT(A(LM)*A(LM)+X*X)
      IF (X) 70,75,75
70    Y = - Y
75    SINX = Y / SQRT(2.0*(1.0+(SQRT(1.0-Y*Y))))
      SINX2 = SINX * SINX
78    COSX = SQRT(1.0-SINX2)
      COSX2 = COSX * COSX
      SINCS = SINX * COSX
C
C   ROTATE L AND M COLUMNS
C
      IQ = N * (L-1)
      IMQ = N * (M-1)
      DO 125 I=1,N
      IQ = (I*I-I) / 2
      IF (I-L) 80,115,80
80    IF(I-M) 85,115,90
85    IM = I + MQ
      GO TO 95
90    IM = M + IQ
95    IF ( I-L) 100, 105, 105
100   IL = I + LQ
      GO TO 110
105   IL = L + IQ
110   X = A(IL) * COSX - A(IM) * SINX
      A(IM) = A(IL) * SINX + A(IM) * COSX
      A(IL) = X
115   IF (MV-1) 120,125,120
120   ILR = ILQ + I
      IMR = IMQ + I
      X = R(ILR) * COSX - R(IMR) * SINX
      R(IMR) = R(ILR) * SINX + R(IMR) * COSX
      R(ILR) = X
125   CONTINUE
      X = 2.0 * A(LM) * SINCS
      Y = A(LL) * COSX2 + A(MM) * SINX2 - X
      X = A(LL) * SINX2 + A(MM) * COSX2 + X
      A(LM) = (A(LL) - A(MM)) * SINCS + A(LM) * (COSX2 - SINX2)
      A(LL) = Y
      A(MM) = X
C
C   TEST FOR  COMPLETION
C   TEST FOR M = LAST COLUMN
C
130   IF (M-N) 135,140,135
135   M = M + I
      GO TO 60
C
C  TEST FOR L = SECOND FROM LAST COLUMN
C
140   IF(L-(N-1)) 145,150,145
145   L = L + 1
      GO TO 55
```

```
150    IF( IND-1 ) 160,155,160
155    IND = 0
       GO TO 50
C
C  COMPARE THREHOLD WITH FINAL NORM
C
160    IF(THR - ANORMX) 165,165,45
C
C    SORT EIGEN VALUES AND EIGENVECTORS
C
165    IQ = - N
       DO 185 I=1,N
       IQ = IQ + N
       LL = I + (I*I-I) / 2
       JQ = N * (I-2)
       DO 185 J = I,N
       JQ = JQ + N
       MM = J + (J*J-J) / 2
       IF ( A(LL) -A(MM)) 170,185,185
170    X = A(LL)
       A(LL) = A(MM)
       A(MM) = X
       IF ( MV-1) 175,185,175
175    DO 180 K=1,N
       ILR  = IQ + K
       IMR = JQ + K
       X = R(ILR)
       R(ILR) = R(IMR)
180    R(IMR) = X
185    CONTINUE
       RETURN
       END
```

โปรแกรมย่อยสำหรับคำนวณผลรวมของค่าความคลาดเคลื่อนเฉลี่ย 1000 ครั้ง

```
C
       SUBROUTINE RPTGEN(G,NUM,ANS1)
       DIMENSION G(15),ANS1(15)
       DO 10 I=1,NUM
10     ANS1(I) = ANS1(I) + G(I)
       RETURN
       END
```

ประวัติผู้เขียน

นางสาว พรศิริ หมื่นไชยศรี เกิดเมื่อวันที่ 19 เดือนกรกฎาคม พ.ศ. 2504 ที่จังหวัดสกลนคร สำเร็จการศึกษาปริญญาตรีวิทยาศาสตรบัณฑิต (สถิติ) จากมหาวิทยาลัยเชียงใหม่ เมื่อปีการศึกษา 2525 และเข้าศึกษาต่อระดับปริญญาโท ในภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2526 โดยได้รับทุนอุดหนุนการศึกษาจากโครงการผลิตและพัฒนาอาจารย์ทบวงมหาวิทยาลัย ตามความต้องการของภาควิชาคณิตศาสตร์ คณะครุศาสตร์อุตสาหกรรมและวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตพระนครเหนือ