

## รายการอ้างอิง

### ภาษาไทย

- Anderson, T.W. 1971. *Introduction to multivariate statistical analysis*. New York : John Wiley.
- Bleasley, J.D. and Springer, S.G. 1977. The percentage points of the normal distribution. *Applied Statistics* 26:118-121.
- Cooper, B.E. 1968a. The integral of student's  $t$  - distribution. *Applied Statistics* 17:189-190.
- Hochberg, Y., Tamhane, A.C. 1987. *Multiple comparison procedure*. New York : John Wiley.
- James, S. 1991. Approximate multinormal probabilities applied to correlated multiple endpoints in clinical trial. *Statistics in Medicine* 10:1123-1135.
- Lehmacher, W., Wassmer, G., Reitmeir, P. 1991. Procedure for two-sample comparisons with multiple endpoints controlling the experimentwise error rate. *Biometrics* 47:511-521.
- Morrison, D.F. 1990. *Multivariate statistical methods*. New York : McGraw-Hill.
- O'Brien, P.C. 1984. Procedure for comparing samples with multiple endpoints. *Biometrics* 40:1079-1087.
- Popcock, S.J., Geller, N.L., Tsiatis, A.A. 1987. The analysis of multiple endpoints. *Biometrics* 43:487-498.
- Ramsey, P.H. 1980, winter. Exact type I error rates for robustness of student's  $t$  test with unequal variances. *Journal of Education Statistics* 5:337-349.
- Rubinstein, R.Y. 1981. *Simulation and the Monte Carlo Method*. New York : John Wiley.
- Troendle, J.F. 1995. A stepwise resampling method of multiple endpoints testing. *Journal of the American Statistical Association* 90:370-378.
- Westfall, P.H., Young, S.S. 1989. P value adjustment for multiple tests in multivariate binomial models. *Journal of the American Statistical Association* 84:780-786.
- Wright, S.P. 1992. Adjusted P-values for simultaneous inference. *Biometrics* 48:1005-1013.



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## ภาคผนวก ก

ตารางที่ 4.30 แสดงเปอร์เซ็นต์ส่วนต่างสัมพัทธ์ระหว่าง วิธี OLS และ GLS ( R (O,G) ) ซึ่งคำนวณได้ดังนี้

$$R (O,G) = \frac{|POLS - PGLS|}{\text{Max} (POLS, PGLS)} \times 100$$

เมื่อ POLS หมายถึง ค่าจากการทดสอบของ วิธี OLS

และ PGLS หมายถึง ค่าจากการทดสอบของ วิธี GLS

สัญลักษณ์ที่ใช้ในตารางที่ 4.30 มีดังนี้

กรณีที่ 1 หมายถึง กรณีสหสัมพันธ์เท่ากัน ณ ระดับนัยสำคัญ  $\alpha = 0.05$

กรณีที่ 2 หมายถึง กรณีสหสัมพันธ์ไม่เท่ากัน ณ ระดับนัยสำคัญ  $\alpha = 0.05$

กรณีที่ 3 หมายถึง กรณีสหสัมพันธ์เท่ากัน ณ ระดับนัยสำคัญ  $\alpha = 0.01$

กรณีที่ 4 หมายถึง กรณีสหสัมพันธ์ไม่เท่ากัน ณ ระดับนัยสำคัญ  $\alpha = 0.01$

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 4.30 แสดงเปอร์เซ็นต์ความต่างสัมพัทธ์ระหว่าง วิธี OLS และ GLS

K	n	$\rho$	กรณีที่ 1	กรณีที่ 2	กรณีที่ 3	กรณีที่ 4
3	10	0.0	-	-	-	-
		0.1	-	-	-	-
		0.2	-	-	-	-
		0.3	-	-	-	-
		0.4	-	-	-	-
		0.5	-	-	-	-
		0.6	-	1.43	-	-
		0.7	1.08	-	-	-
		0.8	-	-	-	-
		0.9	1.27	12.70	1.30	8.70
	30	0.0	0.21	0.51	-	-
		0.1	0.43	4.14	1.55	1.27
		0.2	0.45	3.05	0.00	1.28
		0.3	0.48	1.89	1.27	6.94
		0.4	0.36	1.29	2.30	1.61
		0.5	0.26	4.08	1.08	6.67
		0.6	1.63	0.76	0.83	4.17
		0.7	0.28	0.00	0.44	4.55
		0.8	0.29	4.31	1.39	7.14
		0.9	0.30	0.00	0.55	0.00
	50	0.0	0.00	0.36	-	-
		0.1	0.00	0.00	0.86	3.13
		0.2	0.20	0.44	0.43	5.31
		0.3	0.21	0.96	0.46	1.89
		0.4	0.21	2.60	0.25	5.00
		0.5	0.00	2.09	0.00	2.50
		0.6	0.22	5.26	0.57	1.64
		0.7	0.00	1.32	1.18	4.92
		0.8	0.23	2.74	0.61	0.00
		0.9	0.48	3.70	1.28	9.43

หมายเหตุ “-” หมายถึง กรณีนั้นไม่สามารถหาค่า R (O,G) ได้ เนื่องจาก วิธี OLS หรือ GLS ไม่สามารถควบคุมความน่าจะเป็นของความคลาดเคลื่อนประเภทที่ 1 ได้

ตารางที่ 4.30 ( ต่อ )

K	n	p	กรณีที่ 1	กรณีที่ 2	กรณีที่ 3	กรณีที่ 4
5	10	0.0	-	-	-	-
		0.1	-	-	-	-
		0.2	-	-	-	-
		0.3	-	-	-	-
		0.4	-	-	-	-
		0.5	-	-	-	-
		0.6	-	-	-	-
		0.7	2.58	-	-	-
		0.8	-	-	-	-
		0.9	4.44	-	-	-
	30	0.0	0.20	0.47	-	-
		0.1	0.00	0.26	-	-
		0.2	0.84	2.43	2.57	5.20
		0.3	0.68	4.36	1.68	7.92
		0.4	1.38	3.72	1.52	7.18
		0.5	0.75	3.50	0.34	8.81
		0.6	0.26	4.40	2.58	14.97
		0.7	1.97	4.74	2.39	11.76
		0.8	0.87	10.78	3.32	4.12
		0.9	0.00	3.14	1.49	7.14
	50	0.0	0.00	1.26	-	-
		0.1	0.20	0.65	0.00	1.27
		0.2	0.20	0.67	0.21	1.87
		0.3	0.61	1.39	0.00	2.16
		0.4	0.00	3.87	0.67	3.39
		0.5	0.42	2.52	0.48	1.92
		0.6	0.22	0.85	0.78	8.58
		0.7	0.45	2.36	0.58	13.79
		0.8	0.46	3.85	0.29	12.90
		0.9	0.00	2.03	0.65	9.29

ตารางที่ 4.30 ( ต่อ )

K	n	p	กรณีที่ 1	กรณีที่ 2	กรณีที่ 3	กรณีที่ 4
7	10	0.0	-	-	-	-
		0.1	-	-	-	-
		0.2	-	-	-	-
		0.3	-	-	-	-
		0.4	-	-	-	-
		0.5	-	-	-	-
		0.6	-	-	-	-
		0.7	-	-	-	-
		0.8	-	-	-	-
		0.9	-	-	-	-
	30	0.0	-	-	-	-
		0.1	0.80	0.20	-	-
		0.2	1.03	0.21	0.91	0.69
		0.3	1.72	1.49	1.01	0.00
		0.4	1.38	0.88	1.81	3.49
		0.5	2.16	0.72	0.94	3.86
		0.6	0.52	1.71	1.47	7.72
		0.7	0.55	2.75	3.95	13.88
		0.8	1.47	6.15	0.45	13.40
		0.9	1.52	8.45	0.52	4.49
	50	0.0	0.00	0.00	-	-
		0.1	0.00	0.00	-	-
		0.2	0.00	0.20	0.20	0.21
		0.3	0.20	0.20	0.42	0.00
		0.4	0.00	0.40	1.12	1.52
		0.5	0.21	0.00	0.00	2.53
		0.6	0.43	1.69	0.26	7.32
		0.7	0.67	4.54	1.06	9.07
		0.8	0.23	1.48	0.30	3.90
		0.9	0.71	3.50	0.63	1.63

## ภาคผนวก ข

## รายละเอียดของโปรแกรมที่ใช้ในการวิจัยมีดังนี้

ชื่อ โปรแกรม	คุณสมบัติของ โปรแกรม	ชื่อ โปรแกรมที่เรียกใช้	หน้าที่
<b>โปรแกรมหลัก</b>			
MAIN	<ul style="list-style-type: none"> <li>- อ่านค่าพารามิเตอร์ที่กำหนด</li> <li>- กำหนดค่าเริ่มต้นให้กับตัวแปรที่ใช้ นับจำนวน</li> <li>- ทดสอบสมมติฐานเกี่ยวกับความแตกต่างของค่าเฉลี่ยระหว่าง 2 ประชากรบนพื้นฐานของตัวแปรตามพหุคูณ ด้วย วิธี BON JAM OLS GLS และ WFY</li> <li>- คำนวณค่าความน่าจะเป็นของความเคลื่อนไหวประเภทที่ 1 และค่าอำนาจการทดสอบ</li> <li>- แสดงผล</li> </ul>	GENY,CALPV,ASSORT, BON, JAM, OLS, GLS, WFY, POWER, CALPH, CALINP	147-151
<b>SUBROUTINE</b>			
GENY	สร้างเมทริกซ์ตัวแปรตาม Y1 , Y2	NORMAL	151-152
RANDOM	สร้างเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอ		153-154
CALPV	คำนวณค่า P- value สำหรับวิธี BON JAM และ WFY	PROBST , SS , AVRG	154
ASSORT	เรียงลำดับค่า P- value จากน้อยไปมาก		156-157
BON	ทดสอบสมมติฐานด้วยวิธี BON		157-158
JAM	ทดสอบสมมติฐานด้วยวิธี JAM	PPND , PHI	158-159
POWER	คำนวณค่า $CPI(J)$ ; $J = 1, \dots, K$ สำหรับวิธี BON JAM และ WFY ( $I = 1, 2, 5$ )	FAC	161-162
CALPH	คำนวณค่า $\hat{p}$ สำหรับวิธี OLS และ GLS	SKK , SS , AVRG	163
OLS	ทดสอบสมมติฐานด้วยวิธี OLS	FAC , INDEX , INSERT	165-169

ชื่อโปรแกรม	คุณสมบัติของโปรแกรม	ชื่อโปรแกรมที่เรียกใช้	หน้าที่
CALINP	คำนวณค่า $\hat{\rho}^{-1}$ สำหรับวิธี GLS		164-165
GLS	ทดสอบสมมุติฐานด้วยวิธี GLS	FAC , INDEX , INSERT	170-174
INDEX	คำนวณดัชนีของสมมุติฐานที่เป็นไปได้ทั้งหมด ในแต่ละขั้นของการทดสอบ สำหรับวิธี OLS และ GLS	FAC	174-179
INSERT	คำนวณดัชนีของสมมุติฐานที่ต้องทดสอบในแต่ละขั้นของการทดสอบ สำหรับวิธี OLS และ GLS		179-180
WFY	ทดสอบสมมุติฐานด้วยวิธี WFY	SAMP , CALPV, MINP	180-181
SAMP	สุ่มตัวอย่างแบบแทนที่สำหรับวิธี WFY	SRD	181-182
SRD	สร้างเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอสำหรับการสุ่มซ้ำของวิธี WFY		182-183
FUNCTION			
NORMAL	สร้างเลขสุ่มที่มีการแจกแจงแบบปกติ		153
SS	คำนวณค่า SUM OF SQUARE		154-155
AVRG	คำนวณค่าเฉลี่ย		155
PROBST	คำนวณค่าฟังก์ชันการแจกแจงสะสมของการแจกแจง $t ( F(t) )$		155-156
PPND	คำนวณค่า $\Phi^{-1}(a)$ (ค่า $z$ ที่ทำให้ฟังก์ชันการแจกแจงสะสมของ $z$ หรือ $\Phi(z)$ มีค่าเท่ากับ $a$ )		159-161
PHI	คำนวณค่า $\phi(a)$ ฟังก์ชันความหนาแน่นของการแจกแจงปกติ		161
FAC	คำนวณค่า $FAC(a,b)$		162
SKK	คำนวณค่า $\hat{\sigma}_{kk}$		163-164
MINP	คำนวณค่าต่ำสุดของ P - value		183



```

C*****
C*      A COMPARISON ON TESTING PROCEDURES FOR DIFFERENCE      *
C*      BETWEEN TWO POPULATION MEANS BASED ON MULTIPLE        *
C*      DEPENDENT VARIABLES : BON , JAM , OLS , GLS AND WFY     *
C*****

```

C MAIN PROGRAM

```

      DOUBLE PRECISION T(7),PH(7,7),INP(7,7)
      REAL Y1(50,7),Y2(50,7),M1(7),M2(7),P(7,7),PV(7),PO1(7)
*      PO2(7),PO3(7),PO4(7),PO5(7)
      INTEGER CP1(7),CP2(7),CP3(7),CP4(7),CP5(7),IND(7),FAC
      CHARACTER ECORR,TYPEI
      COMMON /SEBD/LX,LL
*      /SSEED/LSX
*      /DVAR/K
*      /CRIT/Z
*      /ALPHA/AL
*      /CORR/P,C
*      /CHAR/ECORR,TYPEI
*      /NUM/N1,N2
*      /MEAN/M1,M2

      OPEN(1,FILE='IP')
      OPEN(2,FILE='OP')
      LX = 16807
      LSX = 17
      LL = 0
      NROUND = 500
1      READ (1,*,END=30) K,N1,N2,C,AL,TYPEI,ECORR
      WRITE (2,*) '##INPUT DATA##'
      WRITE (2,*) 'K,N1,N2,C,AL=',K,N1,N2,C,AL
      WRITE (2,*) 'TYPEI=',TYPEI,'ECORR=',ECORR
      IF (AL.EQ.0.01) THEN
          Z = 2.32635

```

```

ELSE
    Z = 1.64485
END IF
DO 5 I = 1,K
DO 5 J = 1,K
    IF (I.EQ.J) THEN
        P(I,J) = 1.0
    ELSE
        IF (ECORR.EQ.'Y') THEN
            P(I,J) = C
        ELSE
            P(I,J) = C**ABS(I-J)
        END IF
    END IF
END IF
5 CONTINUE
DO 10 I = 1,K
    M2(I) = 0.0
    IF (TYPE1.EQ.'Y') THEN
        M1(I) = 0.0
    ELSE IF (ECORR.EQ.'Y') THEN
        M1(I) = 0.5
    ELSE
        M1(I) = 0.1*(K-I+1)
    END IF
10 CONTINUE
C* INITIALIZE COUNTER *
IF (TYPE1.EQ.'Y') THEN
    NR1 = 0
    NR2 = 0
    NR3 = 0
    NR4 = 0
    NR5 = 0

```

```

ELSE
  DO 15 I = 1,K
    CP1(I) = 0
    CP2(I) = 0
    CP3(I) = 0
    CP4(I) = 0
    CP5(I) = 0
15  CONTINUE
END IF
DO 20 IR = 1,NROUND
  CALL GENY(Y1,Y2)
  CALL CALPV(Y1,Y2,PV,T)
  CALL ASSORT(PV,IND)
  CALL BON(PV,NRPR1)
  IF (NRPR1.NE.0) THEN
    IF (TYPE1.EQ.'Y') THEN
      NR1 = NR1 + 1
    ELSE
      CALL POWER(NRPR1,CP1)
    END IF
  END IF
  CALL JAM(PV,NRPR2)
  IF (NRPR2.NE.0) THEN
    IF (TYPE1.EQ.'Y') THEN
      NR2 = NR2 + 1
    ELSE
      CALL POWER(NRPR2,CP2)
    END IF
  END IF
  CALL CALPH(Y1,Y2,PH)
  CALL CALINP(PH,INP)
  CALL OLS(PH,T,NR3,CP3)

```

```

CALL GLS(INF,T,NR4,CP4)
CALL WFY(Y1,Y2,T,PV,IND,NRPR3)
IF (NRPR3.NE.0) THEN
  IF (TYPE1.EQ.'Y') THEN
    NR5 = NR5 + 1
  ELSE
    CALL POWER(NRPR3,CP5)
  END IF
END IF
20 CONTINUE
ROUND = NROUND
IF (TYPE1.EQ.'Y') THEN
  TE1 = NR1/ROUND
  TE2 = NR2/ROUND
  TE3 = NR3/ROUND
  TE4 = NR4/ROUND
  TE5 = NR5/ROUND
  WRITE (2,*) 'TE1=',TE1
  WRITE (2,*) 'TE2=',TE2
  WRITE (2,*) 'TE3=',TE3
  WRITE (2,*) 'TE4=',TE4
  WRITE (2,*) 'TE5=',TE5
ELSE
  DO 25 I = 1,K
    NF = FAC(K,K-I+1)
    PO1(I) = CP1(I)/(NF*ROUND)
    PO2(I) = CP2(I)/(NF*ROUND)
    PO3(I) = CP3(I)/(NF*ROUND)
    PO4(I) = CP4(I)/(NF*ROUND)
    PO5(I) = CP5(I)/(NF*ROUND)
25 CONTINUE
  WRITE (2,*) 'PO1(I)=',(PO1(I),I = 1,K)

```

```

WRITE (2,*) 'PO2(I)=',(PO2(I),I = 1,K)
WRITE (2,*) 'PO3(I)=',(PO3(I),I = 1,K)
WRITE (2,*) 'PO4(I)=',(PO4(I),I = 1,K)
WRITE (2,*) 'PO5(I)=',(PO5(I),I = 1,K)

END IF
GOTO 1

30 WRITE (2,*) 'LX =',LX
WRITE (2,*) 'LSX =',LSX
STOP
END

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c          SUBROUTINE GENY          c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

SUBROUTINE GENY(Y1,Y2)
REAL Y1(50,7),Y2(50,7),ZZ1(50,7),ZZ2(50,7),CO(7,7),
* M1(7),M2(7),P(7,7),SCC(50,7),SC(50,7),NORMAL
COMMON /SEED/LX,LL
* /CORR/P,C
* /MEAN/M1,M2
* /DVAR/K
* /NUM/N1,N2

DO 10 I = 1,N1
DO 10 J = 1,K
ZZ1(I,J) = NORMAL(0.0,1.0)
10 CONTINUE
DO 20 I = 1,N2
DO 20 J = 1,K
ZZ2(I,J) = NORMAL(0.0,1.0)
20 CONTINUE
DO 40 I = 1,K
DO 40 J = 1,I
SCC(I,J) = 0.0

```

```

SC(I,J) = 0.0
IF ((J - 1).NE.0) THEN
  J1 = J-1
  DO 30 L = 1,J1
    SCC(I,J) = SCC(I,J) + CO(I,L)*CO(J,L)
    SC(I,J) = SC(I,J) + CO(J,L)*CO(J,L)
30  CONTINUE
  END IF
  CO(I,J) = (P(I,J) - SCC(I,J))/SQRT(P(J,J)-SC(I,J))
40  CONTINUE
  K1 = K-1
  DO 50 I = 1,K1
    I1 = I+1
    DO 50 J = I1,K
      CO(I,J) = 0.0
50  CONTINUE
    DO 70 I = 1,N1
      DO 70 J = 1,K
        Y1(I,J) = 0.0
        DO 60 L = 1,J
60      Y1(I,J) = Y1(I,J) + CO(J,L)*ZZ1(I,L)
        Y1(I,J) = Y1(I,J) + M1(J)
70  CONTINUE
      DO 90 I = 1,N2
        DO 90 J = 1,K
          Y2(I,J) = 0.0
          DO 80 L = 1,J
80      Y2(I,J) = Y2(I,J) + CO(J,L)*ZZ2(I,L)
          Y2(I,J) = Y2(I,J) + M2(J)
90.  CONTINUE
  RETURN
END

```

cc

c                   FUNCTION NORMAL                   c

cc

REAL FUNCTION NORMAL(MEW,SIGMA)

REAL MEW,SIGMA

COMMON /SEED/LX,LL

\*     /Z/Z2

PI = 3.14159265358979

IF (LL.EQ.1) GOTO 20

CALL RANDOM(RAND)

R1 = RAND

CALL RANDOM(RAND)

R2 = RAND

Z1 = SQRT(-2\*ALOG(R1))\*COS(2\*PI\*R2)

Z2 = SQRT(-2\*ALOG(R1))\*SIN(2\*PI\*R2)

NORMAL = Z1\*SIGMA + MEW

LL = 1

RETURN

20   NORMAL = Z2\*SIGMA + MEW

LL = 0

RETURN

END

cc

c                   SUBROUTINE RANDOM                   c

cc

SUBROUTINE RANDOM(RAND)

COMMON /SEED/LX,LL

CON = 2147483647

LY = LX\*16807

IF (LY.LT.0) LY = LY+CON+1

RAND = LY



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

RAND = RAND/CON

LX = LY

RETURN

END

CC

c                   SUBROUTINE CALPV                   c

CC

  SUBROUTINE CALPV(Y1,Y2,PV,T)

  DOUBLE PRECISION SP(7),T(7),SS,AVRG

  REAL Y1(50,7),Y2(50,7),PV(7)

  COMMON/DVAR/K

  \*   /DOF/IDF

  \*   /NUM/N1,N2

  IDF = N1 + N2 - 2

  DF = IDF

  DO 5 J = 1,K

    SP(J) = ((N1-1)\*SS(Y1,N1,J)+(N2-1)\*SS(Y2,N2,J))/DF

    T(J) = (AVRG(Y1,N1,J) - AVRG(Y2,N2,J))

  \*    /DSQRT(SP(J)\*(N1+N2)/(N1\*N2))

    TR = T(J)

    PV(J) = 1.0 - PROBST(TR)

5   CONTINUE

  RETURN

  END

CC

c                   FUNCTION SS                   c

CC

  DOUBLE PRECISION FUNCTION SS(Y,N,J)

  DOUBLE PRECISION AVRG,YBAR,SUM

  REAL Y(50,7)

  YBAR = AVRG(Y,N,J)

  SUM = 0.0



```

DO 10 I = 1,N
    SUM = SUM + (Y(I,J) - YBAR)**2
10 CONTINUE
SS = SUM/(N-1)
RETURN
END

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c      FUNCTION AVERAGE ( AVRG)      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

DOUBLE PRECISION FUNCTION AVRG(Y,N,J)
REAL Y(50,7)
SUM = 0.0
DO 15 I = 1,N
    SUM = SUM + Y(I,J)
15 CONTINUE
AVRG = SUM/N
RETURN
END

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c      FUNCTION PROBST      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

REAL FUNCTION PROBST(T)
C
C COMPUTE THE AREA FROM -INFINITE TO T(IDF) UNDER A
C STUDENT'S CENTRAL T-DISTRIBUTION WITH IDF DEGREE OF
C FREEDOM
C
COMMON /DOF/IDF
DATA G1/0.3183098862/
F = IDF
A = T/SQRT(F)
B = F/(F+T**2)

```



```

J2 = K - JUMP
DO 30 J = 1,J2
    DO 30 I = J,1,-JUMP
25        J3 = I + JUMP
           IF (PV(I).GT.PV(J3)) THEN
               S = PV(I)
               PV(I) = PV(J3)
               PV(J3) = S
               L = IND(I)
               IND(I) = IND(J3)
               IND(J3) = L
           END IF
30        CONTINUE
           GOTO 20
END IF
RETURN
END

```

```

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

```

```

c          SUBROUTINE BON          c

```

```

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

```

```

SUBROUTINE BON(PV,RPR1)

```

```

REAL PV(7)

```

```

INTEGER RPR1

```

```

COMMON /DVAR/K

```

```

*   /ALPHA/AL

```

```

*   /NUM/N1,N2

```

```

RPR1 = 0

```

```

DO 10 I = 1,K

```

```

    CR = AL/(K-I+1)

```

```

    IF (PV(I).LE.CR) THEN

```

```

        RPR1 = RPR1 + 1

```

```

    ELSE

```

```
                GOTO 20
            END IF
.10 CONTINUE
20 RETURN
END
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c          SUBROUTINE JAM          c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
SUBROUTINE JAM(PV1,RPR2)
DOUBLE PRECISION D1,D2,D3,D4,A,PPND,PHI,G(3),
*          DONE,DTWO,DADJP
INTEGER RPR2
REAL PV1(7),LO,P(7,7)
CHARACTER E CORR,TYPEI
COMMON /DVAR/K,
* /ALPHA/AL,
* /CORR/P,C,
* /CHAR/ECORR,TYPEI
DATA G/0.14104740,0.05814822,0.03219472/
DATA ONE/1.0/,TWO/2.0/,DONE/1.0D0/,DTWO/2.0D0/
RPR2 = 0
TK = K/2.0 - 0.5
DO 10 L = 1,K
    D1 = (DONE - PV1(L))**(K-L+1)
    D2 = DONE - PV1(L)
    A = PPND(D2,IFAULT)
    D3 = (K-L+1)*(K-L)*(ONE-PV1(L))**(K-L-1)*PHI(A)**2/DTWO
    D4 = (K-L+1)*(K-L)*PHI(A)*G(TK)/DTWO
    IF (ECORR.NE.'Y') THEN
        SUMLO = 0.0
        K1 = K-1
        DO 5 I = 1,K1
```

```

      I1 = I+1
      DO 5 J = I1,K
        SUMLO = SUMLO + ABS(P(I,J))
5     CONTINUE
      LO = SUMLO/(K*(K-1)/2.0)
      ELSE
        LO = C
      END IF
      DADJP = DONE-D1*(ONE-LO**2)-D2*LO**2-D3*LO*(ONE-LO)
      *   -D4*(TWO-TWO*(ONE-LO)**(0.5)-LO-LO**2)
      ADJP = DADJP
      IF (ADJP.LE.AL) THEN
        RPR2 = RPR2 + 1
      ELSE
        GOTO 20
      END IF
10    CONTINUE
20    RETURN
      END

```

cc

c FUNCTION PPND c

cc

DOUBLE PRECISION FUNCTION PPND(P,IFAUULT)

C

C PRODUCES THE STANDARD NORMAL DEVIATE Z CORRESPONDING

C TO A GIVEN LOWER TAIL AREA OF P

C

C IFAULT = 0 IF 0<P<1.

C IFAULT = 1 OTHERWISE.

C IF IFAULT = 1 THE VALUE OF PPND IS SET EQUAL TO ZERO

C

DOUBLE PRECISION ZERO,SPLIT,HALF,ONE,

\* A0,A1,A2,A3,B1,B2,B3,B4,C0,C1,C2,C3,D1,D2,  
\* P,Q,R

DATA ZERO /0.0D0/,HALF /0.5D0/,ONE /1.0D0/

DATA SPLIT /0.42D0/

DATA A0/ 2.50662 82388 4D0/

DATA A1/ -18.61500 06252 9D0/

DATA A2/ 41.39119 77353 4D0/

DATA A3/ -25.44106 04963 7D0/

DATA B1/ -8.47351 09309 0D0/

DATA B2/ 23.08336 74374 3D0/

DATA B3/ -21.06224 10182 6D0/

DATA B4/ 3.13082 90983 3D0/

DATA C0/ -2.78718 93113 8D0/

DATA C1/ -2.29796 47913 4D0/

DATA C2/ 4.85014 12713 5D0/

DATA C3/ 2.32121 27685 8D0/

DATA D1/ 3.54388 92476 2D0/

DATA D2/ 1.63706 78189 7D0/

IFault = 0

Q = P - HALF

IF (DABS(Q).GT.SPLIT) GOTO 1

R = Q \* Q

PPND = Q \* (((A3 \* R + A2) \* R + A1) \* R + A0)/

\* (((B4 \* R + B3) \* R + B2) \* R + B1) \* R + ONE)

RETURN

1 R = P

IF (Q.GT.ZERO) R = ONE - P

IF (R.LE.ZERO) GOTO 2

R = DSQRT(-DLOG(R))

PPND = (((C3 \* R + C2) \* R + C1) \* R + C0)/

\* ((D2 \* R + D1) \* R + ONE)

IF (Q.LT.ZERO) PPND = -PPND

RETURN

2 IFAULT = 1

PFND = ZERO

RETURN

END

CC

c FUNCTION PHI c

CC

DOUBLE PRECISION FUNCTION PHI(A)

DOUBLE PRECISION A,PI

DATA PI/3.14159265358979/

PHI = DEXP(-(DABS(A))\*\*2.0/2.0)/DSQRT(2.0\*PI)

RETURN

END

CC

c SUBROUTINE POWER c

CC

SUBROUTINE POWER(NR,CP)

INTEGER NR,CP(7),FAC

COMMON /DVAR/K

CP(1) = CP(1) + 1

CP(K) = CP(K) + NR

K1 = K-1

DO 20 I = 2,K1

    KI = K - I + 1

    IF (NR.GE.I) THEN

        CP(I) = CP(I) + FAC(K,KI)

    ELSE

        KN = K - NR

        CP(I) = CP(I) + FAC(K,KI) - FAC(KN,KI)

    END IF

20 CONTINUE





cc

c                    SUBROUTINE CALPH                    c

cc

  SUBROUTINE CALPH(Y1,Y2,PH)

  DOUBLE PRECISION PH(7,7),SKK,SS

  REAL Y1(50,7),Y2(50,7)

  COMMON /DVAR/K

\*    /NUM/N1,N2

  DO 10 J1 = 1,K

  DO 10 J2 = J1,K

    IF (J1.EQ.J2) THEN

      PH(J1,J2) = 1

    ELSE

      PH(J1,J2)=(N1-1)\*SKK(Y1,N1,J1,J2)+(N2-1)\*SKK(Y2,N2,J1,J2))

\*            /DSQRT(((N1-1)\*SS(Y1,N1,J1)+(N2-1)\*SS(Y2,N2,J1))

\*            \*((N1-1)\*SS(Y1,N1,J2)+(N2-1)\*SS(Y2,N2,J2)))

    END IF

10 CONTINUE

  DO 20 J1 = 2,K

    J11 = J1 - 1

  DO 20 J2 = 1,J11

    PH(J1,J2) = PH(J2,J1)

20 CONTINUE

  RETURN

  END

cc

c                    FUNCTION SKK                    c

cc

  DOUBLE PRECISION FUNCTION SKK(Y,N,J1,J2)

  DOUBLE PRECISION AVRG,SUM

  REAL Y(50,7)

```

COMMON /DVAR/K
SUM = 0.0
DO 10 I = 1,N
    SUM = SUM+(Y(I,J1)-AVRG(Y,N,J1))*(Y(I,J2)-AVRG(Y,N,J2))
10 CONTINUE
SKK = SUM/(N-1)
RETURN
END

```

cc

c SUBROUTINE CALINP c

cc

```

SUBROUTINE CALINP(PH,INP)
DOUBLE PRECISION PH(7,7),INP(7,7),A(7,7)
COMMON /DVAR/K
DO 5 I = 1,K
DO 5 J = 1,K
    A(I,J) = PH(I,J)
5 CONTINUE
DO 25 L = 1,K
    A(L,L) = -1.0/A(L,L)
    DO 10 I = 1,K
        IF ((I-L).NE.0) A(I,L) = -A(I,L)*A(L,L)
10 CONTINUE
    DO 15 I = 1,K
    DO 15 J = 1,K
        IF (((I-L)*(J-L)).NE.0) A(I,J) = A(I,J)-A(I,L)*A(L,J)
15 CONTINUE
    DO 20 J = 1,K
        IF ((J-L).NE.0) A(L,J) = -A(L,J)*A(L,L)
20 CONTINUE
25 CONTINUE
DO 30 I = 1,K

```

```

DO 30 J = 1,K
    INP(I,J) = -A(I,J)
30 CONTINUE
RETURN
END

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c          SUBROUTINE OLS          c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

SUBROUTINE OLS(PH,T,NR3,CP3)
DOUBLE PRECISION PH(7,7),T(7),SUMT1,SUMP1,TOLS1,
*      TOLS(35),SUMT(35),SUMP(35)
INTEGER CP3(7),SR,IS(140),IS1(42),IS2(105),IS3(140),
*      IS4(105),IS5(42),C,IRBJ(35),ITEST(35),
*      IT(35,7),STEP,DIM,NUM,DIN(35,7),FAC,D1
CHARACTER ECORR,TYPEI
COMMON /DVAR/K
*      /CRIT/Z
*      /CHAR/ECORR,TYPEI
DATA IS1/1,2,4,7,11,16,
*1,3,5,8,12,17,
*2,3,6,9,13,18,
*4,5,6,10,14,19,
*7,8,9,10,15,20,
*11,12,13,14,15,21,
*16,17,18,19,20,21/
DATA IS2/1,2,5,11,21,1,3,6,12,22,1,4,7,13,23,2,3,8,14,24,2,4,9,
*15,25,3,4,10,16,26,5,6,8,17,27,5,7,9,18,28,6,7,10,19,29,8,9,10,
*20,30,11,12,14,17,31,11,13,15,18,32,12,13,16,19,33,14,15,16,20,34,
*17,18,19,20,35,21,22,24,27,31,21,23,25,28,32,22,23,26,29,33,
*24,25,26,30,34,27,28,29,30,35,31,32,33,34,35/
DATA IS3/1,2,6,16,1,3,7,17,1,4,8,18,1,5,9,19,2,3,10,20,2,4,11,21,
*2,5,12,22,3,4,13,23,3,5,14,24,4,5,15,25,6,7,10,26,6,8,11,27,6,9,

```

```

*12,28,7,8,13,29,7,9,14,30,8,9,15,31,10,11,16,32,10,12,17,33,11,12,
*18,34,13,14,19,35,16,17,20,26,16,18,21,27,16,19,22,28,17,18,23,
*29,17,19,24,30,18,19,25,31,20,21,23,32,20,22,24,33,21,22,25,34,
*23,24,25,35,26,27,29,32,26,28,30,33,27,28,31,34,29,30,31,35,
*32,33,34,35/
DATA IS4/1,2,7,1,3,8,1,4,9,1,5,10,1,6,11,2,3,12,2,4,13,2,5,14,
*2,6,15,3,4,16,3,5,17,3,6,18,4,5,19,4,6,20,5,6,21,7,8,12,7,9,13,
*7,10,14,7,11,15,8,9,16,8,10,17,8,11,18,9,10,19,9,11,20,10,11,21,
*12,13,16,12,14,17,12,15,18,13,14,19,13,15,20,14,15,21,16,17,19,
*16,18,20,17,18,21,19,20,21/
DATA IS5/1,2,1,3,1,4,1,5,1,6,1,7,2,3,2,4,2,5,2,6,2,7,3,4,3,5,3,6,
*3,7,4,5,4,6,4,7,5,6,5,7,6,7/
SUMT1 = 0.0
DO 10 J = 1,K
10    SUMT1 = SUMT1 + T(J)
SUMP1 = 0.0
DO 15 I = 1,K
DO 15 J = 1,K
    SUMP1 = SUMP1 + PH(I,J)
15  CONTINUE
TOLS1 = SUMT1/DSQRT(SUMP1)
IF (TOLS1.GT.Z) THEN
  IF (TYPE.EQ.'Y') THEN
    NR3 = NR3 + 1
  ELSE
    CP3(1) = CP3(1) + 1
    STEP = 2
    DIM = K-STEP+1
    NUM = FAC(K,DIM)
    CALL INDEX(STEP,DIM,NUM,DIN)
    SR = 0
    DO 30 I = 1,NUM

```

```

SUMT(I) = 0.0
DO 20 J = 1,DIM
    L = DIN(I,J)
    SUMT(I) = SUMT(I) + T(L)
20 CONTINUE
SUMP(I) = 0.0
DO 25 J1 = 1,DIM
DO 25 J2 = 1,DIM
    L1 = DIN(I,J1)
    L2 = DIN(I,J2)
    SUMP(I) = SUMP(I) + PH(L1,L2)
25 CONTINUE
TOLS(I) = SUMT(I)/DSQRT(SUMP(I))
IF (TOLS(I).GT.Z) THEN
    SR = SR + 1
    IREJ(SR) = I
END IF
30 CONTINUE
31 CP3(STEP) = CP3(STEP) + SR
IF (SR.GT.0.AND.STEP.LT.K) THEN
    STEP = STEP + 1
    DIM = DIM - 1
    NUM = FAC(K,DIM)
    IF (STEP.LT.K) THEN
        CALL INDEX(STEP,DIM,NUM,DIN)
    ELSE
        DO 35 I = 1,NUM
35     DIN(I,1) = I
        END IF
        D1 = DIM + 1
        MAXJ = FAC(D1,DIM)
        IF (STEP.EQ.3) THEN

```

```
DO 36 I = 1,42
36   IS(I) = IS1(I)
      C = 6
ELSE IF (STEP.EQ.4) THEN
      DO 37 I = 1, 105
37   IS(I) = IS2(I)
      C = 5
ELSE IF (STEP.EQ.5) THEN
      DO 38 I = 1,140
38   IS(I) = IS3(I)
      C = 4
ELSE IF (STEP.EQ.6) THEN
      DO 39 I = 1,105
39   IS(I) = IS4(I)
      C = 3
ELSE
      DO 40 I = 1,42
40   IS(I) = IS5(I)
      C = 2
END IF
DO 41 I = 1,SR
DO 41 J = 1,MAXJ
      L = C*(IREJ(I) - 1) + J
      IT(I,J) = IS(L)
41 CONTINUE
DO 45 J = 1,MAXJ
45   ITEST(J) = IT(1,J)
IF (SR.GE.2) THEN
      M = MAXJ
      DO 50 I = 2,SR
      DO 50 J = 1,MAXJ
          CALL INSERT (ITEST,M,IT,I,J)
```

```
50      CONTINUE
      END IF
      SR = 0
      DO 55 I = 1,M
          L = ITEST(I)
          SUMT(L) = 0.0
          SUMP(L) = 0.0
55      CONTINUE
      DO 70 I = 1,M
          L1 = ITEST(I)
          DO 60 J = 1,DIM
              L2 = DIN(L1,J)
              SUMT(L1) = SUMT(L1) + T(L2)
60      CONTINUE
          DO 65 J1 = 1,DIM
              DO 65 J2 = 1,DIM
                  L3 = DIN(L1,J1)
                  L4 = DIN(L1,J2)
                  SUMP(L1) = SUMP(L1) + PH(L3,L4)
65      CONTINUE
          TOLS(L1) = SUMT(L1)/DSQRT(SUMP(L1))
          IF (TOLS(L1).GT.Z) THEN
              SR = SR + 1
              IREJ(SR) = L1
          END IF
70      CONTINUE
          GOTO 31
      END IF
      END IF
      END IF
      RETURN
      END
```

CC

c SUBROUTINE GLS c

CC

SUBROUTINE GLS(INP,T,NR4,CP4)

DOUBLE PRECISION INP(7,7),T(7),SUMT1,SUMP1,TGLS1,

\* TGLS(35),SUMT(35),SUMP(35)

INTEGER CP4(7),SR,IS(140),IS1(42),IS2(105),IS3(140),

\* NR4,IS4(105),IS5(42),C,IREJ(35),ITEST(35),

\* IT(35,7),STBP,DIM,NUM,DIN(35,7),FAC,D1

CHARACTER E CORR,TYPEI

COMMON /DVAR/K

\* /CRIT/Z

\* /CHAR/E CORR,TYPEI

DATA IS1/1,2,4,7,11,16,

\*1,3,5,8,12,17,

\*2,3,6,9,13,18,

\*4,5,6,10,14,19,

\*7,8,9,10,15,20,

\*11,12,13,14,15,21,

\*16,17,18,19,20,21/

DATA IS2/1,2,5,11,21,1,3,6,12,22,1,4,7,13,23,2,3,8,14,24,2,4,9,

\*15,25,3,4,10,16,26,5,6,8,17,27,5,7,9,18,28,6,7,10,19,29,8,9,10,

\*20,30,11,12,14,17,31,11,13,15,18,32,12,13,16,19,33,14,15,16,20,34,

\*17,18,19,20,35,21,22,24,27,31,21,23,25,28,32,22,23,26,29,33,

\*24,25,26,30,34,27,28,29,30,35,31,32,33,34,35/

DATA IS3/1,2,6,16,1,3,7,17,1,4,8,18,1,5,9,19,2,3,10,20,2,4,11,21,

\*2,5,12,22,3,4,13,23,3,5,14,24,4,5,15,25,6,7,10,26,6,8,11,27,6,9,

\*12,28,7,8,13,29,7,9,14,30,8,9,15,31,10,11,16,32,10,12,17,33,11,12,

\*18,34,13,14,19,35,16,17,20,26,16,18,21,27,16,19,22,28,17,18,23,

\*29,17,19,24,30,18,19,25,31,20,21,23,32,20,22,24,33,21,22,25,34,

\*23,24,25,35,26,27,29,32,26,28,30,33,27,28,31,34,29,30,31,35,



\*32,33,34,35/

DATA IS4/1,2,7,1,3,8,1,4,9,1,5,10,1,6,11,2,3,12,2,4,13,2,5,14,

\*2,6,15,3,4,16,3,5,17,3,6,18,4,5,19,4,6,20,5,6,21,7,8,12,7,9,13,

\*7,10,14,7,11,15,8,9,16,8,10,17,8,11,18,9,10,19,9,11,20,10,11,21,

\*12,13,16,12,14,17,12,15,18,13,14,19,13,15,20,14,15,21,16,17,19,

\*16,18,20,17,18,21,19,20,21/

DATA IS5/1,2,1,3,1,4,1,5,1,6,1,7,2,3,2,4,2,5,2,6,2,7,3,4,3,5,3,6,

\*3,7,4,5,4,6,4,7,5,6,5,7,6,7/

SUMT1 = 0.0

SUMP1 = 0.0

DO 15 I = 1,K

DO 15 J = 1,K

SUMT1 = SUMT1 + INP(I,J)\*T(J)

SUMP1 = SUMP1 + INP(I,J)

15 CONTINUE

TGLS1 = SUMT1/DSQRT(SUMP1)

C WRITE (\*,\*) 'TGLS1=',TGLS1

IF (TGLS1.GT.Z) THEN

IF (TYPE.EQ.'Y') THEN

NR4 = NR4 + 1

ELSE

CP4(1) = CP4(1) + 1

STEP = 2

DIM = K-STEP+1

NUM = FAC(K,DIM)

CALL INDEX(STEP,DIM,NUM,DIN)

SR = 0

DO 30 I = 1,NUM

SUMT(I) = 0.0

SUMP(I) = 0.0

DO 25 J1 = 1,DIM

DO 25 J2 = 1,DIM

```

L1 = DIN(I,J1)
L2 = DIN(I,J2)
SUMT(I) = SUMT(I) + INP(L1,L2)*T(L2)
SUMP(I) = SUMP(I) + INP(L1,L2)
25 CONTINUE
TGLS(I) = SUMT(I)/DSQRT(SUMP(I))
C WRITE (*,*) 'TGLS('I,')='TGLS(I)
IF (TGLS(I).GT.Z) THEN
SR = SR + 1
IREJ(SR) = I
END IF
30 CONTINUE
31 CP4(STEP) = CP4(STEP) + SR
IF (SR.GT.0.AND.STEP.LT.K) THEN
STEP = STEP + 1
DIM = DIM - 1
NUM = FAC(K,DIM)
IF (STEP.LT.K) THEN
CALL INDEX(STEP,DIM,NUM,DIN)
ELSE
DO 35 I = 1,NUM
35 DIN(I,1) = I
END IF
D1 = DIM + 1
MAXJ = FAC(D1,DIM)
IF (STEP.EQ.3) THEN
DO 36 I = 1,42
36 IS(I) = IS1(I)
C = 6
ELSE IF (STEP.EQ.4) THEN
DO 37 I = 1, 105
37 IS(I) = IS2(I)

```

```

      C = 5
    ELSE IF (STEP.EQ.5) THEN
      DO 38 I = 1,140
38      IS(I) = IS3(I)
      C = 4
    ELSE IF (STEP.EQ.6) THEN
      DO 39 I = 1,105
39      IS(I) = IS4(I)
      C = 3
    ELSE
      DO 40 I = 1,42
40      IS(I) = IS5(I)
      C = 2
    END IF
    DO 41 I = 1,SR
    DO 41 J = 1,MAXJ
      L = C*(IREJ(I) - 1) + J
      IT(I,J) = IS(L)
41      CONTINUE
    DO 45 J = 1,MAXJ
45      ITEST(J) = IT(1,J)
    IF (SR.GE.2) THEN
      M = MAXJ
      DO 50 I = 2,SR
      DO 50 J = 1,MAXJ
        CALL INSERT (ITEST,M,IT,I,J)
50      CONTINUE
    END IF
    SR = 0
    DO 55 I = 1,M
      L = ITEST(I)
      SUMT(L) = 0.0

```

```

        SUMP(L) = 0.0
55      CONTINUE
        DO 70 I = 1,M
            L = ITEST(I)
            DO 65 J1 = 1,DIM
                DO 65 J2 = 1,DIM
                    L1 = DIN(L,J1)
                    L2 = DIN(L,J2)
                    SUMT(L) = SUMT(L) + INP(L1,L2)*T(L2)
                    SUMP(L) = SUMP(L) + INP(L1,L2)
65      CONTINUE
            TGLS(L) = SUMT(L)/DSQRT(SUMP(L))
            IF (TGLS(L).GT.Z) THEN
                SR = SR + 1
                IREJ(SR) = L
            END IF
70      CONTINUE
        GOTO 31
    END IF
END IF
END IF
RETURN
END

```

```

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

```

```

c          SUBROUTINE INDEX          c

```

```

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

```

```

SUBROUTINE INDEX(STEP,DIM,NUM,DIN)

```

```

INTEGER STEP,DIM,NUM,DIN(35,7),D1,DJ,DJ1,DJ2,FAC,

```

```

*   C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,

```

```

*   C11,C12,C14,C15,CC9,C9C,C91

```

```

COMMON /DVAR/K

```

IF (STEP.GT.2) THEN

C3 = NUM - DIM

K1 = K - 1

D1 = DIM - 1

IF (STEP.EQ.3) THEN

C6 = FAC(K1,D1)

C7 = C6 - DIM

C8 = C7 - DIM + 1

ELSE

K2 = K - 2

IF (STEP.EQ.4) THEN

CC9 = FAC(K1,D1)

C9C = FAC(K2,D1)

ELSE

K3 = K - 3

IF (STEP.EQ.5) THEN

CC9 = FAC(K1,D1) + FAC(K2,D1)

C9C = FAC(K3,D1)

ELSE

K4 = K - 4

CC9 = FAC(K1,D1) + FAC(K2,D1) + FAC(K3,D1)

C9C = FAC(K4,D1)

END IF

END IF

C9 = CC9 + C9C

C12 = C9 - DIM

C14 = K - DIM + 1

C15 = C14 + STEP - 1

END IF

END IF

DO 15 J = 1,DIM

KJ = K-J

```

DJ = DIM-J
C1 = FAC(KJ,DJ)
C2 = NUM - J
IF (STEP.GT.2) THEN
    KJ1 = K - J - 1
    KJ2 = K - J - 2
    C4 = C1 + FAC(KJ1,DJ)
    C5 = C4 + FAC(KJ2,DJ)
    IF (STEP.GT.3) THEN
        DJ2 = DIM - J + 2
        DJ1 = DIM - J + 1
        C10 = CC9 + FAC(DJ2,DJ)
        C11 = C10 + FAC(DJ1,DJ)
    END IF
END IF
DO 10 I = 1,NUM
IF (I.LE.C1) THEN
    DIN(I,J) = J
ELSE IF (I.GT.C2) THEN
    DIN(I,J) = J+STEP-1
ELSE IF (I.GE.C3.AND.I.LE.C2) THEN
    DIN(I,J) = J+STEP-2
ELSE IF (I.GT.C1.AND.I.LE.C4) THEN
    DIN(I,J) = J+1
ELSE IF (I.GT.C4.AND.I.LE.C5) THEN
    DIN(I,J) = J+2
ELSE IF (J.EQ.1) THEN
    DIN(I,J) = J+3
ELSE IF (I.EQ.CC9.OR.I.EQ.C9) THEN
    DIN(I,J) = J+STEP-1
ELSE IF (STEP.EQ.3) THEN
    GOTO 10

```

```

ELSE IF (I.GT.CC9.AND.I.LE.C10) THEN
    DIN(I,J) = J+STEP-3
ELSE IF (I.GT.C10.AND.I.LE.C11) THEN
    DIN(I,J) = J+STEP-2
END IF
10  CONTINUE
15  CONTINUE
IF (STEP.EQ.3) THEN
    DO 25 J = 3,DIM
        KJ = K-J
        DJ = DIM-J
        KJ1 = K-J-1
        KJ2 = K-J-2
        KS = FAC(KJ,DJ) + FAC(KJ1,DJ) + FAC(KJ2,DJ) + 1
        DO 20 I = KS,C6
            IF (I.GT.C7.AND.I.LE.C6) THEN
                DIN(I,J) = DIN(I+DIM,J)
            ELSE IF (I.GT.C8.AND.I.LE.C7) THEN
                DIN(I,J) = DIN(I+2*DIM-1,J)
            ELSE
                DIN(I,J) = DIN(I+2*DIM+2,J)
            END IF
20  CONTINUE
25  CONTINUE
END IF
IF (K.EQ.7) THEN
    IF (STEP.EQ.4.OR.STEP.EQ.5) THEN
        DO 40 J = 3,DIM
            C91 = C9-1
            DO 30 I = C12,C91
                DIN(I,J) = DIN(I+DIM,J)
30  CONTINUE

```

```

IF (J.EQ.4) THEN
    DO 35 I = 23,25
        DIN(I,J) = DIN(I+3,J)
35    CONTINUE
    END IF
40    CONTINUE
END IF
IF (STEP.EQ.4) THEN
    DO 55 J = 3,4
        DO 45 I = 10,19
45    DIN(I,J) = DIN(I+10,J)
        IF (J.EQ.4) THEN
            DO 50 I = 4,9
50    DIN(I,J) = DIN(I+6,J)
            END IF
55    CONTINUE
        END IF
    END IF
IF (STEP.GT.4) THEN
    J = DIM
    KJ = K-J
    DJ = DIM-J
    KJ1 = K-J-1
    KJ2 = K-J-2
    KS = FAC(KJ,DJ) + FAC(KJ1,DJ) + FAC(KJ2,DJ) + 1
    DO 60 I = KS,C15
        IF (I.LE.C14) THEN
            DIN(I,J) = I+DIM-1
        ELSE IF (I.GT.C14.AND.I.LE.C15) THEN
            DIN(I,J) = DIN(I-STEP+1,J)
        END IF
60    CONTINUE
    IF (STEP.EQ.6) THEN

```



```

        DO 65 I = 12,14
65      DIN(I,J) = DIN(I-4,J)
      ELSE
        DO 70 I = 10,12
70      DIN(I,J) = DIN(I-3,J)
        DO 90 J = 2,3
          DO 75 I = 24,20,-1
75      DIN(I,J) = DIN(I+6,J)
          DO 80 I = 19,16,-1
80      DIN(I,J) = DIN(I-10,J)
          DO 85 I = 13,15
85      DIN(I,J) = DIN(I+10,J)
90      CONTINUE
      END IF
    END IF
  END IF
  RETURN
END

```

```

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

```

```

c          SUBROUTINE INSERT          c

```

```

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

```

```

SUBROUTINE INSERT(ITEST,M,IT,I,J)
INTEGER ITEST(35),IT(35,7)
L = 1
5  IF (IT(I,J).GT.ITEST(L).AND.L.LE.M) THEN
      L = L + 1
      GOTO 5
  ELSE
      IF(L.GT.M) THEN
          ITEST(L) = IT(I,J)
          M = M + 1
      ELSE

```

```

IF (IT(I,J).EQ.ITEST(L)) THEN
    GOTO 20
ELSE
    DO 10 N = M,L,-1
10     ITEST(N+1) = ITEST(N)
        ITEST(L) = IT(I,J)
        M = M + 1
    END IF
END IF
END IF
20 RETURN
END

```

```

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

```

```

c          SUBROUTINE WFY          c

```

```

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

```

```

SUBROUTINE WFY(Y1,Y2,T,PV,IND,RPR3)
DOUBLE PRECISION T(7)
REAL Y1(50,7),Y2(50,7),POP(100,7),X1(50,7),X2(50,7),
*   PV(7),PVL(7),PADJ(7),MINP,MINPV
INTEGER IND(7),RPR3,E(7)
COMMON/DVAR/K
*   /SSBED/LSX
*   /NUM/N1,N2
*   /ALPHA/AL
DO 5 I = 1,K
5   E(I) = 0
NT = N1 + N2
DO 10 I = 1,NT
DO 10 J = 1,K
    IF (I.LE.N1) THEN
        POP(I,J) = Y1(I,J)
    ELSE

```

```

      POP(I,J) = Y2(I-N1,J)
      END IF
10  CONTINUE
      NSR = 1000
      DO 20 I = 1,NSR
          CALL SAMP(POP,NT,N1,X1)
          CALL SAMP(POP,NT,N2,X2)
          CALL CALPV(X1,X2,PVL,T)
          DO 15 J = 1,K
              MINPV = MINP(J,IND,PVL)
              IF (MINPV.LE.PV(J)) E(J) = E(J) + 1
15  CONTINUE
20  CONTINUE
      RPR3 = 0
      DO 25 I = 1,K
          R = NSR
          PADJ(I) = E(I)/R
          IF (PADJ(I).LE.AL) THEN
              RPR3 = RPR3 + 1
          ELSE
              GOTO 30
          END IF
25  CONTINUE
30  RETURN
      END

```

```

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

```

```

c          SUBROUTINE SAMP          c

```

```

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

```

```

SUBROUTINE SAMP(POP,NT,N,X)
REAL POP(100,7),X(50,7),PROB(100)
COMMON /DVAR/K
* /SSBED/LSX

```







## ประวัติผู้วิจัย

นางสาวศศิประภา หิริโศตปี เกิดวันที่ 14 ตุลาคม พ.ศ. 2516 ที่แขวงศิริราช เขตบางกอก-น้อย กรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิทยาศาสตร์บัณฑิต สาขาวิชาสถิติ ( เกียรตินิยม อันดับ 1 ) จากมหาวิทยาลัยศิลปากร ในปีการศึกษา 2537 และเข้าศึกษาต่อในหลักสูตรสถิติศาสตรมหาบัณฑิต ที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2538



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย