บรรณานุกรม

ภาษาไทย

วิทยานิพนธ์

ทรงพันธุ์ ชุณหสวัสดิกุล. "การประมาณค่าสัมประสิทธิ์การถดถอยพหุโดยที่ค่าประมาณสเกลเปลี่ยนไป" วิทยานิพนธ์ ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2532

ปราณี รัตนัง. "การประมาณสัมประสิทธิ์การถดถอยพหุเมื่อความผิดพลาดมีการแจกแจงแบบเบ้และมีแจกแจงแบบหางยาวกว่าการแจกแจงปกติ" วิทยานิพนธ์ ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2531

ภาษาต่างประเทศ

หนังสือ

Lee, E.T. Statistical Methods for Survival Data Analysis, Lifetime Learning Publication : USA, 1980.

Robinstein, R.Y. Simulation and Monte Carlo method. New York : John Wiley, 1981.

Searle S.R. Linear Models. New York : John Wiley & Sons : 1970

บทคัดย่อ

Gibbons, D.G. "A Simulation Study of Some Ridge Estimators." Journal of the American Statistical Association 76(1981) : 136-139.

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

_____. "Ridge Regression : Applications to Nonorthogonal Problems." Technometrics 12(1970) : 69-82.

Hoerl, A.E. and Kennard, R.W. and Baldwin, K.F. "Ridge Regression : Some Simulation." Communications in Statistics 4(2)(1975) : 105-123.

Lee, T.S. "A Simulation Study of Deterministic Ridge Estimators." Journal of Statistical Computation and Simulation 24(1986) : 171-183.

McDonald, G.C. and Galarneau, D.I. "A Monte Carlo Evaluation of Some Ridge type Estimators." Journal of American Statistical Association 70(1975) : 407-416.

Wichern, D.W. and Churchill, A.G. "A Comparison of Ridge Estimators." Technometrics 20(1978) : 301-311.

ภาคผนวก ก.

ภาคผนวก ก.

1. <u>วิธีการค้นหาค่า $\hat{\lambda}$ ซึ่งทำให้ $\hat{\sigma}^2_{E|\lambda}$ มีค่าน้อยที่สุด สำหรับการแปลงที่ใช้การยกกำลังของ Box และ Cox</u>

วิธีการค้นหาค่า $\hat{\lambda}$ ที่ทำให้ $\hat{\sigma}^2_{E|\lambda}$ มีค่าน้อยที่สุด สำหรับการแปลงที่ใช้การยกกำลังของ Box และ Cox นี้ ผู้วิจัยได้เขียนขึ้นเพื่อจะช่วยให้การค้นหา $\hat{\lambda}$ ได้รวดเร็วขึ้น จาก Montgomery (ค.ศ.1982 : 94 - 96) ให้ข้อสังเกตว่าโดยปกติเราจะหาค่า $\hat{\lambda}$ จำนวน 10 ถึง 20 ค่าให้เพียงพอสำหรับการประมาณค่าที่เหมาะสม

วิธีการค้นหา $\hat{\lambda}$ ซึ่งทำให้ $\hat{\sigma}^2_{E|\lambda}$ มีค่าน้อยที่สุดสำหรับการแปลงที่ใช้การยกกำลังของ Box และ Cox มีขั้นตอนดังนี้

1) กำหนดช่วงพิสัยของ λ ที่จะใช้ในการหา $\hat{\sigma}^2_{E|\lambda}$ โดยให้

λ_1 เป็นจุดเริ่มต้น λ_2 เป็นจุดกลาง และ λ_3 เป็นจุดสุดท้าย

ระหว่างจุด λ_1 กับ λ_2 สามารถแบ่งเป็นจุดเล็กๆที่ยอมรับได้ห่างกัน d = 0.1

จำนวนจุดที่อยู่ระหว่าง λ_1 กับ λ_2 = MR = $(|\lambda_1| + |\lambda_2|) / d$

โดย MR เป็นเลขจำนวนเต็มคู่มีค่าเป็น 2^b

2) หาค่า $\hat{\sigma}^2_{E|\lambda}$, $\hat{\mathfrak{L}}^2_{E|\lambda}$ ของทั้ง 3 จุดจะได้เป็น

$\hat{\sigma}^2_{E|\lambda_1}$, $\hat{\mathfrak{L}}^2_{E|\lambda_1}$

$\hat{\sigma}^2_{E|\lambda_2}$, $\hat{\mathfrak{L}}^2_{E|\lambda_2}$

$\hat{\sigma}^2_{E|\lambda_3}$, $\hat{\mathfrak{L}}^2_{E|\lambda_3}$

3) เปรียบเทียบ $\hat{\sigma}^2_{E|\lambda_1}$ กับ $\hat{\sigma}^2_{E|\lambda_2}$ และ $\hat{\sigma}^2_{E|\lambda_3}$ โดยหาค่า $\hat{\sigma}^2_{E|\lambda}$ ที่มีค่าน้อยที่สุด เมื่อค่า $\hat{\sigma}^2_{E|\lambda}$ มีค่าน้อยที่สุดจะกำหนดให้เป็น $\hat{\lambda}$ นี้เป็น λ_2 เปลี่ยนค่า $\hat{\sigma}^2_{E|\lambda}$ และ $\mathfrak{L}^2_{E|\lambda}$ ใดๆมาเป็น $\hat{\sigma}^2_{E|\lambda_2}$ และ $\mathfrak{L}^2_{E|\lambda_2}$

4) เปรียบเทียบค่า MR และ 1

ถ้า MR = 1 จะยอมรับค่า λ_2 เป็นค่าที่ให้ $\hat{\sigma}^2_{E|\lambda}$ น้อยที่สุด และจะได้ค่า

ถ้า $\lambda_1 \leq \lambda_2$ ด้วยถือว่าจบกระบวนการ (ถ้า MR ≠ 1 ให้ทำข้อ 5)

5) คำนวณค่า MR(ใหม่) = MR(เก่า)/2

คำนวณ $\lambda_1 = \lambda_2 - 0.1 * MR$ (ใหม่)

คำนวณ $\lambda_3 = \lambda_2 - 0.1 * MR$ (เก่า)

6) กลับไป 2)

สำหรับโปรแกรมนี้ λ_1 และ λ_3 สามารถเคลื่อนย้ายออกไปจากจุดที่กำหนดตั้งแต่เริ่มต้นได้ ถ้าหากว่าค่า λ ที่ต้องการไม่ได้อยู่ในพิสัยตั้งแต่เริ่มต้น และจำนวนครั้งของการค้นหาจะขึ้นอยู่กับจำนวนของ MR ครั้งแรก ซึ่งถ้า $MR = 2^b$ จำนวนครั้งของการค้นหาเท่ากับ 2b + 3 ครั้ง

2. การทดสอบว่าข้อมูลมีการแจกแจงแบบปกติของ Shapiro และ Wilk

การทดสอบของ Shapiro และ Wilk (1965 : 591 - 611) มีค่าสถิติทดสอบคือ

$$W = \frac{\sum_{i=1}^{n} a_{n-i+1}(Y_{n-i+1} - Y_i)^2}{\sum_{i=1}^{n} (Y_i - \bar{Y})^2}$$

โดย k เป็นค่าของเลขนับที่ใหญ่ที่สุดของ n/2

a_{n-i+1} เป็นสัมประสิทธิ์จากตารางของ Shapiro และ Wilk สำหรับขนาดตัวอย่าง $n < 50$

ก. ถ้า n เป็นเลขคู่ โดยที่ n = 2k คำนวณ

$$b = \sum_{i=1}^{n} a_{n-i+1}(Y_{n-i+1} - Y_i)$$

ถ้า n เป็นเลขคี่ โดยที่ $n = 2k + 1$ คำนวณ b ได้ดังนี้ เมื่อ $a_{k+1} = 0$

$$b = a_n(Y_{n-}Y_1)+\ldots+a_{k+2}(Y_{k+2}-Y_k)$$

เมื่อค่า Y_{k+1} ค่ามัธยฐาน(median) ไม่ถูกนำไปคำนวณค่า b ด้วย

ข. คำนวณค่า $s^2 = \sum_{i=1}^{n} (Y_i - \bar{Y})^2$

ค. คำนวณ w ในรูปของ $w = \frac{b^2}{s^2}$

ง. เปรียบเทียบค่า w (คำนวณ) กับ w (ตาราง) ถ้า w (คำนวณ) มีค่าน้อยอย่างมีนัยสำคัญ แสดงว่าข้อมูลไม่ได้มาจากการแจกแจงแบบปกติ

Shapiro, Wilk และ Chen(1964 : 1343 - 1372) ได้ศึกษาเปรียบเทียบตัวสถิติทดสอบ W กับ $\sqrt{b_1}$(โมเมนต์ที่ 3), b_2, KS(Kolmogorov - Smirnov), CM(Cramer-Von Mises), WCM(Weighted CM), D(modified KS), CS(Chi-squared) และ U(Studentized rang) ในการแจกแจง 45 แบบต่างๆกัน และขนาดตัวอย่างต่างๆกัน วิธี การทดสอบของ Shapiro และ Wilk เป็นตัววัดความเป็นปกติที่เหนือกว่าวิธีการอื่นๆ

สำหรับขนาดตัวอย่าง $n > 50$, Shapiro และ Francia (1972) ได้กล่าวถึงค่าสถิติ W ที่ปรับปรุงขึ้น ในรูปประมาณเป็นการทดสอบ W′

โดย

$$W = \frac{\sum_{i=1}^{n} b_i(Y_i)^2}{\sum_{i=1}^{n} (Y_i - \bar{Y})^2}$$

เมื่อ $b' = (b_1, b_2, \ldots, b_n)$

$$= \frac{m'}{(m'm)^{1/2}}$$

และ $m' = (m_1, \ldots, m_n)$ แทนเวกเตอร์ของค่าปกติมาตรฐานของค่าสถิติอันดับ (Order Statistics)

ตารางที่ 1 แสดงค่าสัมประสิทธิ์ (a_{n-i+1}) สำหรับการทดสอบ W ของการแจกแจงปกติ สำหรับ n เท่ากับ 10

i	a_{n-i+1}
1	0.5739
2	0.3291
3	0.2141
4	0.1224
5	0.0399

ตารางที่ 2 แสดงค่าสัมประสิทธิ์ (a_{n-i+1}) สำหรับการทดสอบ W ของการแจกแจงปกติ สำหรับ n เท่ากับ 30

i	a_{n-i+1}	i	a_{n-i+1}
1	0.4254	9	0.1036
2	0.2944	10	0.0862
3	0.2487	11	0.0697
4	0.2148	12	0.0537
5	0.1870	13	0.0381
6	0.1630	14	0.0227
7	0.1415	15	0.0076
8	0.1219		

ตารางที่ 3 แสดงค่าสัมประสิทธิ์ (a_{n-i+1}) สำหรับการทดสอบ W ของการแจกแจงปกติ สำหรับ n เท่ากับ 50

i	a_{n-i+1}	i	a_{n-i+1}
1	0.3751	14	0.0846
2	0.2574	15	0.0764
3	0.2260	16	0.0685
4	0.2032	17	0.0608
5	0.1847	18	0.0532
6	0.1691	19	0.0459
7	0.1554	20	0.0386
8	0.1430	21	0.0314
9	0.1317	22	0.0244
10	0.1212	23	0.0174
11	0.1113	24	0.0104
12	0.1020	25	0.0035
13	0.0932		

ตารางที่ 4 แสดง Percentage point ของการทดสอบ W สำหรับ n = 10,30 และ 50

n	ระดับนัยสำคัญ								
	0.01	0.02	0.05	0.10	0.05	0.90	0.95	0.98	0.99
10	.781	.806	.842	.869	.938	.972	.978	.983	.986
30	.900	.912	.927	.939	.967	.983	.985	.988	.900
50	.930	.938	.947	.955	.974	.985	.988	.990	.991

ตารางที่ 5 แสดง Percentage Point ของการทดสอบ W สำหรับ n = 100

n	ระดับนัยสำคัญ			
	0.01	0.05	0.10	0.50
100	0.968	0.976	0.981	0.990

ตารางที่ 6 แสดงค่า b โดย $b = \frac{m'}{(m'm)^{1/2}}$ ใช้กับขนาดของ n = 100

k	b	k	b	k	b	k	b
1	0.2543	14	0.1115	27	0.0635	39	0.0296
2	0.2178	15	0.1070	28	0.0605	40	0.0269
3	0.1974	16	0.1026	29	0.0575	41	0.0243
4	0.1827	17	0.0985	30	0.0545	42	0.0217
5	0.1711	18	0.0945	31	0.0516	43	0.0191
6	0.1613	19	0.0907	32	0.0487	44	0.0161
7	0.1529	20	0.0869	33	0.0459	45	0.0166
8	0.1454	21	0.0833	34	0.0431	46	0.0140
9	0.1386	22	0.0798	35	0.0404	47	0.0114
10	0.1324	23	0.0764	36	0.0376	48	0.0089
11	0.1267	24	0.0731	37	0.0349	49	0.0038
12	0.1213	25	0.0698	38	0.0322	50	0.0013
13	0.1163	26	0.0666				

ภาคผนวก ข.

ภาคผนวก ข.

```
C     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     :::                          MAIN1                         :::
C     :::   FOR : NORMAL & CONTAMINATED NORMAL DISTRIBUTION      :::
C     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      DOUBLE PRECISION A,S,B,X,B2,B1
      COMMON /REGRS/A(11,11),S(12,12)/VARIAB/N,M,M2
     *       /CONTA/P,IC/INTERV/XBAR1(11),SG(11)
     *       /DATAXY/X(12,150)/SEED/IX/SELECT/KK
     *       /DIST/II/RIDMSE/TRVAB(1)/BMAIN/B1A(11)
     *       /COEFF/B(11)/EIGEN/ALAMP(5)
     *       /PX/F(5,5)/RCOEFF/BRCHA(6)
     *       /UYY/OYY,SSN
     *       /CON/C,SS,TOT,TRACE,BBRID,TSEB,TSEBR
     *       /SUMDB/SQDIFB(6)
     *       /RINIT/ALP1,ALP2
     *       /SDRID/TMEAN(6),SSBRID(6)
      DIMENSION B2(11),B1(11),T(150),MSE(5),TX(5,5),A2(150),VB(5,5),
     *  BRID(5,1),VA(5,5),SSB(6),VV(5,5),ZA(5,5),QA(5,5),ZK(5,5),
     *  XP(6,100),QQ(6,100),TS(5,1),W(5,5),VAZ(5,5),
     *  BCHA(6),CK(100),SMSER(100),SDK(100),
     *  SQHKB(6),TBHKB(6),SQLEE(6),TBLEE(6),TSQMCG(6,100),SQMCG(6),
     *  TBMCG(6),CMSEH(6),CMSEL(6),CMSEM(6),SBRCHA(6,100),BRMCG(6)
      READ(1,15)N,M
   15 FORMAT(I3,I2)
      DO 20 I = 1,M
   20 READ(1,25)XBAR1(I),SG(I)
   25 FORMAT(2F7.4)
      READ(1,30) II
   30 FORMAT(I1)
```

```
      IMK = 0.0
      DO 87 IMK=1,3
      GOTO (7,8,9),IMK
C     ::::::::::::::::::::::::::::::::::::::::::::::::::
C     :::            YOU HAVE 2 CHOICES               :::
C     ::: 1.  PART I  => FOR 3 INDEPENDENT VARIABLES  :::
C     ::: 2.  PART II => FOR 5 INDEPENDENT VARIABLES  :::
C     :::                                             :::
C     ::: IF YOU SELECT PART I  => PUT C IN PART II   :::
C     ::: IF YOU SELECT PART II => PUT C IN PART I    :::
C     ::::::::::::::::::::::::::::::::::::::::::::::::::
C
C     ==  PART I : 3 INDEPENDENT VARIABLES (8 LINES)  ==
C
C   7 ALP1 = 0.99
C     ALP2 = 0.00
C     GOTO 188
C   8 ALP1 = 0.90
C     ALP2 = 0.00
C     GOTO 188
C   9 ALP1 = 0.70
C     ALP2 = 0.00
C     ..................................................
C
C     ==  PART II  :  5 INDEPENDENT VARIABLES  (8 LINES)  ==
C
    7 ALP1 = 0.99
      ALP2 = 0.99
      GOTO 188
    8 ALP1 = 0.99
      ALP2 = 0.90
      GOTO 188
    9 ALP1 = 0.70
      ALP2 = 0.30
C     ..................................................
```

```
  188 M2  = M + 1
      IX = 973253
      KK = 0
      K  = INT(FLOAT(N/2))
      MIX = M + 1
      DO 1930 I=1,MIX
      TBHKB(I) = 0.0
      TBLEE(I) = 0.0
      TBMCG(I) = 0.0
      CMSEH(I) = 0.0
      CMSEL(I) = 0.0
      CMSEM(I) = 0.0
      SQHKB(I) = 0.0
      SQLEE(I) = 0.0
      SQMCG(I) = 0.0
      DO 1930 J=1,100
 1930 TSQMCG(I,J) = 0.0
C     ------------------------------------------------------
      SAM2H  = 0.0
      SSAM2H = 0.0
      SAM2L  = 0.0
      SSAM2L = 0.0
      SAM2M  = 0.0
      SSAM2M = 0.0
      IKH  = 0.0
      IKL = 0.0
      IIM = 0.0
      S1HL  = 0.0
      SS1HL = 0.0
      S2HL  = 0.0
      SS2HL = 0.0
      S1HM  = 0.0
      SS1HM = 0.0
      S2HM  = 0.0
      SS2HM = 0.0
```

```
      S1FLH  = 0.0
      SS1FLH = 0.0
      S2FLH  = 0.0
      SS2FLH = 0.0
      S1FLM  = 0.0
      SS1FLM = 0.0
      S2FLM  = 0.0
      SS2FLM = 0.0
      S1FMMH  = 0.0
      D1FMMH  = 0.0
      S2FMMH  = 0.0
      D2FMMH  = 0.0
      S1FMML  = 0.0
      D1FMML  = 0.0
      S2FMML  = 0.0
      D2FMML  = 0.0
      S1OLS = 0.0
      SSOLS = 0.0
      ATH   = 0.0
      ATTH  = 0.0
      ATL   = 0.0
      ATTL  = 0.0
      ATM   = 0.0
      ATTM  = 0.0
C     ------------------------------------------------------
      JJ = 200
C     ------------------------------------------------------
      WRITE(3,26) N,M
   26 FORMAT(10X,'NO. OF OBSERVATION = ',I3,2X,'NO. OF VARIABLE = ',I2)
      GOTO (10,10,40,50),II
   10 WRITE(3,8333)
 8333 FORMAT(2X,' ERROR  :  PLEASE RUN MAIN 1 ')
      STOP
   40 WRITE(3,41)
   41 FORMAT(30X,'** BETA FROM CONTAMINATE NORMAL  DISTIBUTION ***')
```

```
      IC = 10
   40 WRITE(3,42)IC,P
   42 FORMAT(/30X,'    WITH C = ',I2,' & P= ',F4.2)
      GOTO 105
   50 WRITE(3,1151)
 1151 FORMAT(10X,'******* NORMAL  DISTIBUTION *******')
C     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
  105 DO 130 I=1,JJ
C     ----------------------------------------------------------
      CALL INIT
      CALL DATA
C     ----------------------------------------------------------
C     ----------------------------------------------------------
C     *****          FIND TRACE OF X'X INVERSE          *****
C
      TRACE = 0.0
      DO 291 IK=1,M
      DO 291 JK=1,M
      IF(IK-JK)291,290,291
  290 TRACE = TRACE + A(IK,JK)
  291 CONTINUE
      DO 633 IB=1,M2
      DO 633 IE=1,N
      XP(IB,IE)=XX(IB,IE)
  633 CONTINUE
      M = M2
      CALL CORR(XP,M,N,QQ,SSB)
      DO 644 IB=1,M2
      DO 644 IE=1,M2
      X(IB,IE) = QQ(IB,IE)
  644 CONTINUE
      DO 645 IB=1,N
  645 SSBRID(IB) = SSB(IB)
      M = M2 - 1
  120 CALL OLS(B1,TOT)
C     ----------------------------------------------------------
```

```
      DO 8188 IU=1,M
 8188 BCHA(IU) = B1(IU)*(SSB(MIX)/SSB(IU))
      TAFOLS = 0.0
      DO 3216 IU=1,M
 3216 TAFOLS = TAFOLS + BCHA(IU)*TMEAN(IU)
      BCHA(MIX) = TMEAN(MIX) - TAFOLS
      AOLS = 0.0
      DO 9771 IH=1,MIX
 9771 AOLS = AOLS + (BCHA(IH) - B(IH))*(BCHA(IH) - B(IH))
      AOLS = AOLS/MIX
      S1OLS = S1OLS + AOLS
      SSOLS = SSOLS + AOLS*AOLS
C     ------------------------------------------------------
      BB = 0.0
      TSBL = 0.0
      OTSBL = 0.0
      OSS = 0.0
      SEB = 0.0
      DO 355 IT=1,M
      B1A(IT) = B1(IT)
      BB = BB + B1(IT)*B1(IT)
  355 TSBL = TSBL + B1(IT)*S(M2,IT)
      SS = (TOT - TSBL)/(N-M-1)
      OTSBL = OTSBL + OYY*TSBL
      OSS = OSS + OYY*SS
      CHKB = (M*SS)/BB
      CLEE = ALAMP(M)
      SEB = SEB +SS*TRACE
C     ------------------------------------------------------
      BBOLS = 0.0
      C = CHKB
      CALL RID
C     ------------------------------------------------------
      AM2H = 0.0
      DO 9001 IH=1,MIX
```

```
      SQHKB(IH) = 0.0
      AM2H = AM2H + (BRCHA(IH) - B(IH))*(BRCHA(IH) - B(IH))
 9001 SQHKB(IH) = SQDIFB(IH)
      AM2H   = AM2H/MIX
      SAM2H  = SAM2H + AM2H
      SSAM2H = SSAM2H + AM2H*AM2H
C     ------------------------------------------------------
      DO 9002 IH=1,MIX
 9002 TBHKB(IH) = TBHKB(IH) + SQHKB(IH)
C     ------------------------------------------------------
      C = CLEE
      CALL RID
C     ------------------------------------------------------
      AM2L = 0.0
      DO 9020 IH=1,MIX
      SQLEE(IH) = 0.0
      AM2L = AM2L + (BRCHA(IH)-B(IH))*(BRCHA(IH)-B(IH))
 9020 SQLEE(IH) = SQDIFB(IH)
      AM2L  = AM2L/MIX
      SAM2L = SAM2L + AM2L
      SSAM2L = SSAM2L + AM2L*AM2L
C     ------------------------------------------------------
      DO 9021 IH=1,MIX
 9021 TBLEE(IH) = TBLEE(IH) + SQLEE(IH)
C     ------------------------------------------------------
      C = 0.0
      BBOLS = BB - SS*TRACE
C     ------------------------------------------------------
      IF(BBOLS .GT. 0) GOTO 505
      C = 0.0
      CALL RID
      DO 9050 IH=1,MIX
      BRMCG(IH) = BRCHA(IH)
      SQMCG(IH) = 0.0
 9050 SQMCG(IH) = SQDIFB(IH)
```

```
      GOTO 413
C     ------------------------------------------------------
  505 IIKM = 1
      DO 1999 IMC=1,100
      ILOOP = IMC
      C = C + 0.01
      CK(IMC) = C
      DELTA = 0.001
      CALL RID
      DO 9070 IH=1,MIX
      SBRCHA(IH,IMC) = BRCHA(IH)
 9070 TSQMCG(IH,IMC) = SQDIFB(IH)
C     ------------------------------------------------------
      DK = ABS(BBOLS - BBRID)
      SDK(IMC) = DK
      IF (DK.LE.DELTA) GOTO 1888
 1999 CONTINUE
 1888 DMIN = SDK(1)
      DO 2888 KM=2,ILOOP
      IF (SDK(KM) .GE. DMIN) GOTO 900
      DMIN = SDK(KM)
      IKM  = KM
      IIKM = IKM
  900 IKM  = IIKM
 2888 CONTINUE
C     ------------------------------------------------------
      DO 9088 IH=1,MIX
      BRMCG(IH) = 0.0
      SQMCG(IH) = 0.0
      BRMCG(IH) = SBRCHA(IH,IKM)
 9080 SQMCG(IH) = TSQMCG(IH,IKM)
C     ------------------------------------------------------
  413 AM2M = 0.0
      DO 9081 IH=1,MIX
      AM2M = AM2M + (BRMCG(IH)-B(IH))*(BRMCG(IH)-B(IH))
```

```
9081 TBMCG(IH) = TBMCG(IH) + SQMCG(IH)
     AM2M = AM2M/MIX
     SAM2M  = SAM2M + AM2M
     SSAM2M = SSAM2M + AM2M*AM2M
C    ------------------------------------------------------
     IF (AM2H .LE. AM2L .AND. AM2H .LE. AM2M) GOTO 31
     IF (AM2L .LE. AM2H .AND. AM2L .LE. AM2M) GOTO 32
     IF (AM2M .LE. AM2H .AND. AM2M .LE. AM2L) GOTO 33
  31 FMSEHL = AM2H/AM2L
     DIFFHL = AM2L/AM2H - 1
     IKH = IKH + 1
     ATH   = ATH + AM2H
     ATTH  = ATTH + AM2H*AM2H
     S1HL  = S1HL + FMSEHL
     SS1HL = SS1HL + FMSEHL*FMSEHL
     S2HL  = S2HL + DIFFHL
     SS2HL = SS2HL + DIFFHL*DIFFHL
     FMSEHM = AM2H/AM2M
     DIFFHM = AM2M/AM2H - 1
     S1HM  = S1HM + FMSEHM
     SS1HM = SS1HM + FMSEHM*FMSEHM
     S2HM  = S2HM + DIFFHM
     SS2HM = SS2HM + DIFFHM*DIFFHM
     GOTO 130
  32 FLH  = AM2L/AM2H
     DLH  = AM2H/AM2L - 1
     ATL  = ATL + AM2L
     ATTL = ATTL + AM2L*AM2L
     IKL = IKL + 1
     S1FLH  = S1FLH + FLH
     SS1FLH = SS1FLH + FLH*FFLH
     S2FLH  = S2FLH + DLH
     SS2FLH = SS2FLH + DLH*DLH
     FLM = AM2L/AM2M
     DLM = AM2M/AM2L - 1
```

```
      S1FLM  = S1FLM + FLM
      SS1FLM = SS1HM + FLM*FLM
      S2FLM  = S2FLM + DLM
      SS2FLM = SS2FLM + DLM*DLM
      GOTO 130
   33 FMMH = AM2M/AM2H
      DMMH = AM2H/AM2M - 1
      ATM  = ATM + AM2M
      ATTM = ATTM + AM2M*AM2M
      IIM = IIM + 1
      S1FMMH  = S1FMMH + FMMH
      D1FMMH = D1FMMH + FMMH*FMMH
      S2FMMH  = S2MMH + DMMH
      D2MMH  = D2MMH + DMMH*DMMH
      FMML = AM2M/AM2L
      DMML = AM2L/AM2M - 1
      S1FMML = S1FMML + FMML
      D1FMML = D1MML + FMML*FMML
      S2FMML = S2FMML + DMML
      D2FMML = D2MML + DMML*DMML
C     ------------------------------------------------------
  130 CONTINUE
C     ------------------------------------------------------
      IF (IKH .EQ. 0 .OR. (IKH-1) .EQ. 0 )GOTO 47
      AX1  = S1HL/FLOAT(IKH)
      SD1  = SQRT((SS1HL - (S1HL*AX1))/FLOAT(IKH-1))
      CV1  = SD1/AX1
      AX1M = S1HM/FLOAT(IKH)
      SD1M = SQRT((SS1HM - (S1HM*AX1M))/FLOAT(IKH-1))
      CV1M = SD1M/AX1M
      AX2  = S2HL/FLOAT(IKH)
      SD2  = SQRT((SS1HL - (S2HL*AX2))/FLOAT(IKH-1))
      CV2  = SD2/AX2
      AX2M = S2HM/FLOAT(IKH)
      SD2M = SQRT((SS2HM - (S2HM*AX2M))/FLOAT(IKH-1))
```

```
      CV2M = SD2M/AX2M
   47 IF(IKL .EQ. 0 .OR. (IKL-1) .EQ. 0 ) GOTO 48
      AX3 = S1FLH/FLOAT(IKL)
      SD3 = SQRT((SS1FLH - (S1FLH*AX3))/FLOAT(IKL-1))
      CV3 = SD3/AX3
      AX3M = S1FLM/FLOAT(IKL)
      SD3M = SQRT((SS1FLM - (S1FLM*AX3M))/FLOAT(IKL-1))
      CV3M = SD3M/AX3M
      AX4 = S2FLH/FLOAT(IKL)
      SD4 = SQRT((SS2FLH - (S2FLH*AX4))/FLOAT(IKL-1))
      CV4 = SD4/AX4
      AX4M = S2FLM/FLOAT(IKL)
      SD4M = SQRT((SS2FLM - (S2FLM*AX4M))/FLOAT(IKL-1))
      CV4M = SD4M/AX4M
   48 IF(IIM .EQ. 0 OR .(IIM-1) .EQ. 0 ) GOTO 49
      AX5  = S1FMMH/FLOAT(IIM)
      SD5  = SQRT((D1FMMH - (S1FMMH*AX5))/FLOAT(IIM-1))
      CV5  = SD5/AX5
      AX5L = S1FMML/FLOAT(IIM)
      SD5L = SQRT((D1FMML - (S1FMML*AX5L))/FLOAT(IIM-1))
      CV5M = SD5L/AX5L
      AX6  = S2FMMH/FLOAT(IIM)
      SD6  = SQRT((D2FMMH - (S2FMMH*AX6))/FLOAT(IIM-1))
      CV6  = SD6/AX6
      AX6L = S2FMML/FLOAT(IIM)
      SD6L = SQRT((D2FMML - (S2FMML*AX6L))/FLOAT(IIM-1))
      CV6M = SD6L/AX6L
      WRITE(3,1553) ALP1,ALP2
 1553 FORMAT(10X,':::::',F5.2,',',F5.2,':::::')
      IF (II .EQ. 1) GOTO 61
      WRITE(3,1122)SSN
 1122 FORMAT(22X,'SIGMA-SQ = ',F5.2)
   61   WRITE(3,51)
   51 FORMAT('     ')
   49 WRITE(3,882)
```

```
  882 FORMAT(2X,'COMPARE WITH HKB ' )
      WRITE(3,880)
  880 FORMAT(2X,' SUM(X) ==> XBAR ==> SUMMSQ(X) ==> SD ==> I ==> CV ')
      WRITE(3,881) S1HL,AX1,SS1HL,SD1,IKH,CV1
      WRITE(3,881) S2HL,AX2,SS2HL,SD2,IKH,CV2
      WRITE(3,881) S1HM,AX1M,SS1HM,SD1M,IKH,CV1M
      WRITE(3,881) S2HM,AX2M,SS2HM,SD2M,IKH,CV2M
      WRITE(3,51)
      WRITE(3,883)
  883 FORMAT(2X,'  COMPARE WITH LEE  ')
      WRITE(3,880)
      WRITE(3,881) S1FLH,AX3,SS1FLH,SD3,IKL,CV3
      WRITE(3,881) S2FLH,AX4,SS2FLH,SD4,IKL,CV4
      WRITE(3,881) S1FLM,AX3M,SS1FLM,SD3M,IKL,CV3M
      WRITE(3,881) S2FLM,AX4M,SS2FLM,SD4M,IKL,CV4M
      WRITE(3,51)
      WRITE(3,884)
  884 FORMAT(2X,'  COMPARE WITH MCG  ')
      WRITE(3,880)
      WRITE(3,881) S1FMMH,AX5,D1FMMH,SD5,IIM,CV5
      WRITE(3,881) S2FMMH,AX6,D2FMMH,SD6,IIM,CV6
      WRITE(3,881) S1FMML,AX5L,D1FMML,SD5L,IIM,CV5M
      WRITE(3,881) S2FMMH,AX6L,D2FMML,SD6L,IIM,CV6M
  881 FORMAT(2X,F15.6,2X,F15.6,2X,F15.6,2X,F15.6,2X,I3,2X,F15.6)
      AJJ = JJ
C     ---------------------------------------------------
      SDATH = SQRT((ATTH - (ATH*ATH))/IKH/(IKH-1))
      SDATL = SQRT((ATTL - (ATL*ATL))/IKL/(IKL-1))
      SDATM = SQRT((ATTM - (ATM*ATM))/IKM/(IIM-1))
      CVATH = SDATH/(ATH/IKH)
      CVATL = SDATL/(ATL/IKL)
      CVATM = SDATM/(ATM/IIM)
      WRITE(3,141) SDATH,CVATH
      WRITE(3,141) SDATL,CVATL
      WRITE(3,141) SDATM,CVATM
```

```
  141 FORMAT(2X,'  SUB-SD = ',F15.6,2X,'SUB-CV =',F15.6)
C     ------------------------------------------------------
      XBDHL = (SAM2H - SAM2L)/SQRT(AJJ)
      XBDHM = (SAM2H - SAM2M)/SQRT(AJJ)
      XBDLM = (SAM2L - SAM2M)/SQRT(AJJ)
      XBOLS = S1OLS/AJJ
      XBDH1 = SAM2H/AJJ
      XBDL1 = SAM2L/AJJ
      XBDM1 = SAM2M/AJJ
      SDOLS = SSOLS - (S1OLS*S1OLS)/AJJ
      SDH1  = SSAM2H - (SAM2H*SAM2H)/AJJ
      SDL1  = SSAM2L - (SAM2L*SAM2L)/AJJ
      SDM1  = SSAM2M - (SAM2M*SAM2M)/AJJ
      SDO11 = SQRT(SDOLS/AJJ-1)
      SDH11 = SQRT(SDH1/(AJJ-1)
      SDL11 = SQRT(SDL1/(AJJ-1)
      SDM11 = SQRT(SDM1/(AJJ-1)
      CVO = SDO11/XBOLS
      CVH = SDH11/XBDH1
      CVL = SDL11/XBDL1
      CVM = SDM11/XBDM1
      ZDHL = XBDHL/SQRT((SDH1 + SDL1)/(AJJ-1))
      ZDHM = XBDHM/SQRT((SDH1 + SDM1)/(AJJ-1))
      ZDLM = XBDLM/SQRT((SDL1 + SDM1)/(AJJ-1))
      WRITE(3,51)
      WRITE(3,887)
  887 FORMAT(2X,'  SUM-X , SUMSQ-X  ===>  HKB,LEE,MCG  ')
      WRITE(3,886)SAM2H,SSAM2H,SAM2L,SSAM2L,SAM2M,SSAM2M
      WRITE(3,51)
      WRITE(3,889)
  889 FORMAT(2X,'XBAR ==> OLS,HKB,LEE,MCG AND SD  ==> OLS,HKD,LEE,MCG ')
      WRITE(3,886)XBOLS,XBDH1,XBDL1,XBDM1,SDO11,SDH11,SDL11,SDM11
      WRITE(3,51)
```

```
      WRITE(3,809)
  809 FORMAT(2X,'CV ==> OLS,HKB,LEE,MCG AND ZD ==> HKB,LEE,MCG')
      WRITE(3,886)CVO,CVH,CVL,CVM,ZDHL,ZDHM,ZDLM
  886 FORMAT(2X,7(2X,F15.6))
C     ------------------------------------------------------
      IF(XBDH1 .LE. XBDL1 .AND. XBDH1 .LE. XBDM1) GOTO 73
      IF(XBDL1 .LE. XBDH1 .AND. XBDL1 .LE. XBDM1) GOTO 74
      IF(XBDM1 .LE. XBDH1 .AND. XBDM1 .LE. XBDL1) GOTO 75
   73 TAN1 = XBDL1/XBDH1-1
      TAN11 = TAN1*100
      TAN2 = XBDM1/XBDH1-1
      TAN22 = TAN2 * 100
      TANA = XBDH1/XBDL1
      TANB = XBDH1/XBDM1
      WRITE(3,703)TAN1,TAN11,TAN2,TAN22,TANA,TANB
  703 FORMAT(2X,'HKB=>MIN',1X,'(L/H)-1 =',F10.6,1X,F10.6,
     * 1X,'(M/H)-1 =',F10.6,1X,F10.6,2X,'H/L =',F10.6,2X
     * ,'H/M =',F10.6)
       GOTO 737
   74 TAN1 = XBDH1/XBDL1 - 1
      TAN11 = TAN1*100
      TAN2 = XBDM1/XBDL1 - 1
      TAN22 = TAN2*100
      TANA = XBDL1/XBDH1
      TANB = XBDL1/XBDM1
      WRITE(3,704)TAN1,TAN2,TAN22,TANA,TANB
  704 FORMAT(2X,'LEE=>MIN',1X,'(H/L)-1 = ',F10.6,1X,F10.6,
     * 1X,'(M/L)-1 = ',F10.6,1X,F10.6,2X,'L/H =',F10.6,2X
     * ,'L/M =',F10.6)
      GOTO 737
   75 TAN1 = XBDH1/XBDM1 - 1
      TAN11 = TAN1*100
      TAN2 = XBDL1/XBDM1 - 1
      TAN22 = TAN2*100
      TANA = XBDM1/XBDH1
```

```
      TANB = XBDM1/XBDL1
      WRITE(3,705)TAN1,TAN11,TAN2,TAN22,TANA,TANB
  705 FORMAT(2X,'MCG=>MIN',1X,'(H/M)-1 = ',F10.6,1X,F10.6,
     * 1X,'(L/M)-1 = ',F10.6,1X,F10.6,2X,'M/H =',F10.6,2X
     * ,'M/L =',F10.6)
  737 TANOH1 = XBDH1/XBOLS - 1
      TANOL1 = XBDL1/XBOLS - 1
      TANOM1 = XBDM1/XBOLS - 1
      TOH = XBOLS/XBDH1
      TOL = XBOLS/XBDL1
      TOM = XBOLS/XBDM1
      WRITE(3,709)TANOH1,TANOL1,TANOM1,TOH,TOL,TOM
  709 FORMAT(2X,'H/O-1  L/O-1  M/O-1 ==>',3(2X,F8.4),2X,
     * 'O/H  O/L  O/M  ==>',3(2X,F8.4))
C     ------------------------------------------------------
  139 AJJ = JJ
      WRITE(3,8007)AJJ
 8007 FORMAT(2X,F4.0,2X,'ITERATION',/2X,4X,'---------')
      DO 9900 IH=1,MIX
      CMSEH(IH) = TBHKB(IH)/AJJ
      CMSEL(IH) = TBLEE(IH)/AJJ
      CMSEM(IH) = TBMCG(IH)/AJJ
 9900 WRITE(3,1990)CMSEH(IH),CMSEL(IH),CMSEM(IH)
 1990 FORMAT(2X,'CMSEH=',F15.6,2X,'CMSEL=',F15.6,2X,'CMSEM=',F15.6)
      EMSEH = 0.0
      EMSEL = 0.0
      EMSEM = 0.0
      TMSEH = 0.0
      TMSEL = 0.0
      TMSEM = 0.0
      DO 9901 IH=1,MIX
      TMSEH = TMSEH + CMSEH(IH)
      TMSEL = TMSEL + CMSEL(IH)
 9901 TMSEM = TMSEM + CMSEM(IH)
      EMSEH = TMSEH/MIX
```

```
      EMSEL = TMSEL/MIX
      EMSEM = TMSEM/MIX
      WRITE(3,8850)
      WRITE(3,8050)EMSEH,EMSEL,EMSEM
 8050 FORMAT(2X,'MSE(HKB) =',F15.6,2X,'MSE(LEE) =',F15.6,2X,'MSE(MCG)='
     * ,F15.6)
      WRITE(3,8850)
      RHL = (TMSEL/TMSEH)*100
      RHM = (TMSEM/TMSEH)*100
      RLM = (TMSEM/TMSEL)*100
      ABRHL = ABS((TMSEH - TMSEL)/TMSEL)
      ABRHM = ABS((TMSEH - TMSEM)/TMSEM)
      ABRLM = ABS((TMSEL - TMSEM)/TMSEM)
      WRITE(3,8850)
 8850 FORMAT(2X,'----------------------------------------')
      WRITE(3,9350)RHL
 9350 FORMAT(10X,'(MSEL/MSEH)X100 = ',F15.6)
      WRITE(3,3450)RHM
 3450 FORMAT(10X,'(MSEM/MSEH)X100 = ',F15.6)
      WRITE(3,9550)RLM
 9550 FORMAT(10X,'(MSEM/MSEL)X100 = ',F15.6)
      WRITE(3,8850)
      WRITE(3,9251)ABRHL
 9251 FORMAT(10X,'|(H-L)/L| = ',F15.6)
      WRITE(3,9351)ABRHM
 9351 FORMAT(10X,'|(H-M)/M| = ',F15.6)
      WRITE(3,9451)ABRLM
 9451 FORMAT(10X,'|(L-M)/M| = ',F15.6)
   87 CONTINUE
      STOP
      END
```

```
C     ::::::::::::::::::::::::::::::::::::::::::::::::::
C     :::                  MAIN2                       :::
C     :::   FOR : LOG-NORMAL & WEIBULL DISTRIBUTION    :::
C     ::::::::::::::::::::::::::::::::::::::::::::::::::
      DOUBLE PRECISION A,S,B,X,B2,B1,X1,BC,FL,T,A2,SW,Y1,BMEST
      COMMON /REGRS/A(11,11),S(12,12)
     *       /CONTA/P,IC/INTERV/XBAR1(11),SG(11)
     *       /DATAXY/X(12,150)/SEED/IX/SELECT/KK
     *       /DIST/II
     *       /COEFF/B(11)/EIGEN/ALAMP(5)
     *       /PX/F(5,5)
     *       /UYY/OYY,SSN
     *       /CON/C,SS,TOT,TRACE,BBRID
     *       /SUMDB/SQDIFB(6)
     *       /RINIT/ALP1,ALP2
     *       /SDRID/TMEAN(6),SSBRID(6)
     *       /SKEWD/ALPHA,BETA
     *       /TABLE/A2(150),SW(9)
     *       /TRANSF/Y1(150),FL(3),ST,FIN
     *       /DATXY1/X1(12,150)/RCOEFF/BRCHA(6)
      DIMENSION B2(11),B1(11),T(150),MSE(5),TX(5,5),
     *  BRID(5,1),VA(5,5),SSB(6),BC(11),
     *  XP(6,100),QQ(6,100),
     *  BCHA(6),CK(100),SMSER(100),SDK(100),
     *  SQHKB(6),TBHKB(6),SQLEE(6),TBLEE(6),TSQMCG(6,100),SQMCG(6),
     *  TBMCG(6),CMSEH(6),CMSEL(6),CMSEM(6),SBRCHA(6,100),BRMCG(6)
      READ(1,15)N,M
   15 FORMAT(I3,I2)
      DO 20 I = 1,M
   20 READ(1,25)XBAR1(I),SG(I)
   25 FORMAT(2F7.4)
      READ(1,30) II
   30 FORMAT(I1)
      K = INT(FLOAT(N/2))
      DO 16 I=1,K
```

```
   16 READ(1,17) A2(N-I-1)
   17 FORMAT(F6.4)
      DO 2 I=1,4
    2 READ(1,3) SW(I)
    3 FORMAT(F6.3)
      IMK = 0.0
C
      DO 87 IMK=1,3
      GOTO (7,8,9),IMK
C     ::::::::::::::::::::::::::::::::::::::::::::::::::
C     :::            YOU HAVE 2 CHOICES               :::
C     ::: 1.  PART I  => FOR 3 INDEPENDENT VARIABLES  :::
C     ::: 2.  PART II => FOR 5 INDEPENDENT VARIABLES  :::
C     :::                                             :::
C     ::: IF YOU SELECT PART I  => PUT C IN PART II   :::
C     ::: IF YOU SELECT PART II => PUT C IN PART I    :::
C     ::::::::::::::::::::::::::::::::::::::::::::::::::
C
C     ==  PART I : 3 INDEPENDENT VARIABLES (8 LINES)  ==
C
C   7 ALP1 = 0.99
C     ALP2 = 0.00
C     GOTO 188
C   8 ALP1 = 0.90
C     ALP2 = 0.00
C     GOTO 188
C   9 ALP1 = 0.70
C     ALP2 = 0.00
C     ..................................................
C
C     ==  PART II  :  5 INDEPENDENT VARIABLES  (8 LINES)  ==
C
    7 ALP1 = 0.99
      ALP2 = 0.99
      GOTO 188
```

```
    8 ALP1 = 0.99
      ALP2 = 0.90
      GOTO 188
    9 ALP1 = 0.70
      ALP2 = 0.30
C     ..................................................
  188 M2  = M + 1
      III = 0
      F1  = 0.0
      FL1 = 0.0
      FLL1 = 0.0
      IX = 973253
      KK = 0
      K  = INT(FLOAT(N/2))
      MIX = M + 1
      DO 1930 I=1,MIX
      TBHKB(I) = 0.0
      TBLEE(I) = 0.0
      TBMCG(I) = 0.0
      CMSEH(I) = 0.0
      CMSEL(I) = 0.0
      CMSEM(I) = 0.0
      SQHKB(I) = 0.0
      SQLEE(I) = 0.0
      SQMCG(I) = 0.0
      DO 1930 J=1,100
 1930 TSQMCG(I,J) = 0.0
C     --------------------------------------------------
      SAM2H  = 0.0
      SSAM2H = 0.0
      SAM2L  = 0.0
      SSAM2L = 0.0
      SAM2M  = 0.0
      SSAM2M = 0.0
      IKH  = 0.0
```

```
      IKL = 0.0
      IIM = 0.0
      S1HL  = 0.0
      SS1HL = 0.0
      S2HL  = 0.0
      SS2HL = 0.0
      S1HM  = 0.0
      SS1HM = 0.0
      S2HM  = 0.0
      SS2HM = 0.0
      S1FLH  = 0.0
      SS1FLH = 0.0
      S2FLH  = 0.0
      SS2FLH = 0.0
      S1FLM  = 0.0
      SS1FLM = 0.0
      S2FLM  = 0.0
      SS2FLM = 0.0
      S1FMMH  = 0.0
      D1FMMH  = 0.0
      S2FMMH  = 0.0
      D2FMMH  = 0.0
      S1FMML  = 0.0
      D1FMML  = 0.0
      S2FMML  = 0.0
      D2FMML  = 0.0
      S1OLS = 0.0
      SSOLS = 0.0
      ATH   = 0.0
      ATTH  = 0.0
      ATL   = 0.0
      ATTL  = 0.0
      ATM   = 0.0
      ATTM  = 0.0
C     ------------------------------------------------------
```

```
      JJ = 200
C     ------------------------------------------------------
      WRITE(3,26) N,M
   26 FORMAT(10X,'NO. OF OBSERVATION = ',I3,2X,'NO. OF VARIABLE = ',I2)
      GOTO (10,10,60,80),II
   10 WRITE(3,8333)
 8333 FORMAT(2X,' ERROR  :  PLEASE RUN MAIN 1 ')
      STOP
   60 WRITE(3,61)
   61 FORMAT(30X,'******** LOG-NORMAL  DISTIBUTION ********')
      GOTO 105
   80 WRITE(3,81)
   81 FORMAT(/30X,'******* WEIBULL  DISTIBUTION *******')
      ALPHA = 2
      BETA  = 1
      WRITE(3,62) BETA,ALPHA
   62 FORMAT(/30X,'     WITH BETA = ',F4.1,' & ALPHA = ',F4.1)
  105 DO 130 I=1,JJ
C     ------------------------------------------------------
      CALL INIT
      CALL DATA
C     ------------------------------------------------------
      DO 1000 J=1,M2
      DO 1000 LK=1,N
 1000 X1(J,LK) = X(J,LK)
      CALL BOXCOX(BC,FL1)
      F1 = F1 + FL1
      FLL1 = FLL1 + FL1*FL1
      FL1(1) = FL1
      CALL BCOX(FL(1),1)
      DO 210 J=1,N
      OYY = X1(M2,J)*X1(M2,J)
  210 T(J) = X1(M2,J)
      CALL SHAPWK(N,T,ITEST)
      IF(ITEST .EQ. 2 ) GOTO 220
```

```
      III = III + 1
C     ------------------------------------------------------
  220 CALL INIT1
C     ------------------------------------------------------
C     *****          FIND TRACE OF X'X INVERSE          *****
C
      TRACE = 0.0
      DO 291 IK=1,M
      DO 291 JK=1,M
      IF(IK-JK)291,290,291
  290 TRACE = TRACE + A(IK,JK)
  291 CONTINUE
      DO 633 IB=1,M2
      DO 633 IE=1,N
      XP(IB,IE)=QQ(IB,IE)
  633 CONTINUE
      M = M2
      CALL CORR(XP,M,N,QQ,SSB)
      DO 644 IB=1,M2
      DO 644 IE=1,M2
      X(IB,IE) = QQ(IB,IE)
  644 CONTINUE
      DO 645 IB=1,N
  645 SSBRID(IB) = SSB(IB)
      M = M2 - 1
  120 CALL OLS(B1,TOT)
C     ------------------------------------------------------
      DO 8188 IU=1,M
 8188 BCHA(IU) = B1(IU)*(SSB(MIX)/SSB(IU))
      TAFOLS = 0.0
      DO 3216 IU=1,M
 3216 TAFOLS = TAFOLS + BCHA(IU)*TMEAN(IU)
      BCHA(MIX) = TMEAN(MIX) - TAFOLS
      AOLS = 0.0
      DO 9771 IH=1,MIX
```

```
 9771 AOLS = AOLS + (BCHA(IH) - B(IH))*(BCHA(IH) - B(IH))
      AOLS = AOLS/MIX
      S1OLS = S1OLS + AOLS
      SSOLS = SSOLS + AOLS*AOLS
C     ------------------------------------------------------
      BB = 0.0
      TSBL = 0.0
      OTSBL = 0.0
      OSS = 0.0
      SEB = 0.0
      DO 355 IT=1,M
      BB = BB + B1(IT)*B1(IT)
  355 TSBL = TSBL + B1(IT)*S(M2,IT)
      SS = (TOT - TSBL)/(N-M-1)
      OTSBL = OTSBL + OYY*TSBL
      OSS = OSS + OYY*SS
      CHKB = (M*SS)/BB
      CLEE = ALAMP(M)
      SEB = SEB +SS*TRACE
C     ------------------------------------------------------
      BBOLS = 0.0
      C = CHKB
      CALL RID
C     ------------------------------------------------------
      AM2H = 0.0
      DO 9001 IH=1,MIX
      SQHKB(IH) = 0.0
      AM2H = AM2H + (BRCHA(IH) - B(IH))*(BRCHA(IH) - B(IH))
 9001 SQHKB(IH) = SQDIFB(IH)
      AM2H   = AM2H/MIX
      SAM2H  = SAM2H + AM2H
      SSAM2H = SSAM2H + AM2H*AM2H
C     ------------------------------------------------------
      DO 9002 IH=1,MIX
 9002 TBHKB(IH) = TBHKB(IH) + SQHKB(IH)
```

```
C     ------------------------------------------------------
      C = CLEE
      CALL RID
C     ------------------------------------------------------
      AM2L = 0.0
      DO 9020 IH=1,MIX
      SQLEE(IH) = 0.0
      AM2L = AM2L + (BRCHA(IH)-B(IH))*(BRCHA(IH)-B(IH))
 9020 SQLEE(IH) = SQDIFB(IH)
      AM2L  = AM2L/MIX
      SAM2L = SAM2L + AM2L
      SSAM2L = SSAM2L + AM2L*AM2L
C     ------------------------------------------------------
      DO 9021 IH=1,MIX
 9021 TBLEE(IH) = TBLEE(IH) + SQLEE(IH)
C     ------------------------------------------------------
      C = 0.0
      BBOLS = BB - SS*TRACE
C     ------------------------------------------------------
      IF(BBOLS .GT. 0) GOTO 505
      C = 0.0
      CALL RID
      DO 9050 IH=1,MIX
      BRMCG(IH) = BRCHA(IH)
      SQMCG(IH) = 0.0
 9050 SQMCG(IH) = SQDIFB(IH)
      GOTO 413
C     ------------------------------------------------------
  505 IIKM = 1
      DO 1999 IMC=1,100
      ILOOP = IMC
      C = C + 0.01
      CK(IMC) = C
      DELTA = 0.001
      CALL RID
```

```
      DO 9070 IH=1,MIX
      SBRCHA(IH,IMC) = BRCHA(IH)
 9070 TSQMCG(IH,IMC) = SQDIFB(IH)
C     ------------------------------------------------------
      DK = ABS(BBOLS - BBRID)
      SDK(IMC) = DK
      IF (DK.LE.DELTA) GOTO 1888
 1999 CONTINUE
 1888 DMIN = SDK(1)
      DO 2888 KM=2,ILOOP
      IF (SDK(KM) .GE. DMIN) GOTO 900
      DMIN = SDK(KM)
      IKM  = KM
      IIKM = IKM
  900 IKM  = IIKM
 2888 CONTINUE
C     ------------------------------------------------------
      DO 9088 IH=1,MIX
      BRMCG(IH) = 0.0
      SQMCG(IH) = 0.0
      BRMCG(IH) = SBRCHA(IH,IKM)
 9080 SQMCG(IH) = TSQMCG(IH,IKM)
C     ------------------------------------------------------
  413 AM2M = 0.0
      DO 9081 IH=1,MIX
      AM2M = AM2M + (BRMCG(IH)-B(IH))*(BRMCG(IH)-B(IH))
 9081 TBMCG(IH) = TBMCG(IH) + SQMCG(IH)
      AM2M = AM2M/MIX
      SAM2M  = SAM2M + AM2M
      SSAM2M = SSAM2M + AM2M*AM2M
C     ------------------------------------------------------
      IF (AM2H .LE. AM2L .AND. AM2H .LE. AM2M) GOTO 31
      IF (AM2L .LE. AM2H .AND. AM2L .LE. AM2M) GOTO 32
      IF (AM2M .LE. AM2H .AND. AM2M .LE. AM2L) GOTO 33
   31 FMSEHL = AM2H/AM2L
```

```
      DIFFHL = AM2L/AM2H - 1
      IKH = IKH + 1
      ATH   = ATH + AM2H
      ATTH  = ATTH + AM2H*AM2H
      S1HL  = S1HL + FMSEHL
      SS1HL = SS1HL + FMSEHL*FMSEHL
      S2HL  = S2HL + DIFFHL
      SS2HL = SS2HL + DIFFHL*DIFFHL
      FMSEHM = AM2H/AM2M
      DIFFHM = AM2M/AM2H - 1
      S1HM  = S1HM + FMSEHM
      SS1HM = SS1HM + FMSEHM*FMSEHM
      S2HM  = S2HM + DIFFHM
      SS2HM = SS2HM + DIFFHM*DIFFHM
      GOTO 130
   32 FLH  = AM2L/AM2H
      DLH  = AM2H/AM2L - 1
      ATL  = ATL + AM2L
      ATTL = ATTL + AM2L*AM2L
      IKL = IKL + 1
      S1FLH  = S1FLH + FLH
      SS1FLH = SS1FLH + FLH*FFLH
      S2FLH  = S2FLH + DLH
      SS2FLH = SS2FLH + DLH*DLH
      FLM = AM2L/AM2M
      DLM = AM2M/AM2L - 1
      S1FLM  = S1FLM + FLM
      SS1FLM = SS1HM + FLM*FLM
      S2FLM  = S2FLM + DLM
      SS2FLM = SS2FLM + DLM*DLM
      GOTO 130
   33 FMMH = AM2M/AM2H
      DMMH = AM2H/AM2M - 1
      ATM  = ATM + AM2M
      ATTM = ATTM + AM2M*AM2M
```

```
      IIM = IIM + 1
       S1FMMH  = S1FMMH + FMMH
       D1FMMH = D1FMMH + FMMH*FMMH
       S2FMMH  = S2MMH + DMMH
       D2MMH  = D2MMH + DMMH*DMMH
       FMML = AM2M/AM2L
       DMML = AM2L/AM2M - 1
       S1FMML = S1FMML + FMML
       D1FMML = D1MML + FMML*FMML
       S2FMML = S2FMML + DMML
       D2FMML = D2MML + DMML*DMML
C      ------------------------------------------------------
  130 CONTINUE
C      ------------------------------------------------------
       IF (IKH .EQ. 0 .OR. (IKH-1) .EQ. 0 )GOTO 47
       AX1  = S1HL/FLOAT(IKH)
       SD1  = SQRT((SS1HL - (S1HL*AX1))/FLOAT(IKH-1))
       CV1  = SD1/AX1
       AX1M = S1HM/FLOAT(IKH)
       SD1M = SQRT((SS1HM - (S1HM*AX1M))/FLOAT(IKH-1))
       CV1M = SD1M/AX1M
       AX2  = S2HL/FLOAT(IKH)
       SD2  = SQRT((SS1HL - (S2HL*AX2))/FLOAT(IKH-1))
       CV2  = SD2/AX2
       AX2M = S2HM/FLOAT(IKH)
       SD2M = SQRT((SS2HM - (S2HM*AX2M))/FLOAT(IKH-1))
       CV2M = SD2M/AX2M
   47  IF(IKL .EQ. 0 .OR. (IKL-1) .EQ. 0 ) GOTO 48
       AX3 = S1FLH/FLOAT(IKL)
       SD3 = SQRT((SS1FLH - (S1FLH*AX3))/FLOAT(IKL-1))
       CV3 = SD3/AX3
       AX3M = S1FLM/FLOAT(IKL)
       SD3M = SQRT((SS1FLM - (S1FLM*AX3M))/FLOAT(IKL-1))
       CV3M = SD3M/AX3M
       AX4 = S2FLH/FLOAT(IKL)
```

```
      SD4 = SQRT((SS2FLH - (S2FLH*AX4))/FLOAT(IKL-1))
      CV4 = SD4/AX4
      AX4M = S2FLM/FLOAT(IKL)
      SD4M = SQRT((SS2FLM - (S2FLM*AX4M))/FLOAT(IKL-1))
      CV4M = SD4M/AX4M
   48 IF(IIM .EQ. 0 OR .(IIM-1) .EQ. 0 ) GOTO 49
      AX5  = S1FMMH/FLOAT(IIM)
      SD5  = SQRT((D1FMMH - (S1FMMH*AX5))/FLOAT(IIM-1))
      CV5  = SD5/AX5
      AX5L = S1FMML/FLOAT(IIM)
      SD5L = SQRT((D1FMML - (S1FMML*AX5L))/FLOAT(IIM-1))
      CV5M = SD5L/AX5L
      AX6  = S2FMMH/FLOAT(IIM)
      SD6  = SQRT((D2FMMH - (S2FMMH*AX6))/FLOAT(IIM-1))
      CV6  = SD6/AX6
      AX6L = S2FMML/FLOAT(IIM)
      SD6L = SQRT((D2FMML - (S2FMML*AX6L))/FLOAT(IIM-1))
      CV6M = SD6L/AX6L
      WRITE(3,1553) ALP1,ALP2
 1553 FORMAT(10X,':::::',F5.2,',',F5.2,':::::')
      WRITE(3,1122)SSN
 1122 FORMAT(22X,'SIGMA-SQ = ',F5.2)
      WRITE(3,51)
   51 FORMAT('     ')
   49 WRITE(3,882)
  882 FORMAT(2X,'COMPARE WITH HKB ' )
      WRITE(3,880)
  880 FORMAT(2X,' SUM(X) ==> XBAR ==> SUMMSQ(X) ==> SD ==> I ==> CV ')
      WRITE(3,881) S1HL,AX1,SS1HL,SD1,IKH,CV1
      WRITE(3,881) S2HL,AX2,SS2HL,SD2,IKH,CV2
      WRITE(3,881) S1HM,AX1M,SS1HM,SD1M,IKH,CV1M
      WRITE(3,881) S2HM,AX2M,SS2HM,SD2M,IKH,CV2M
      WRITE(3,51)
      WRITE(3,883)
  883 FORMAT(2X,'  COMPARE WITH LEE  ')
```

```
      WRITE(3,880)
      WRITE(3,881) S1FLH,AX3,SS1FLH,SD3,IKL,CV3
      WRITE(3,881) S2FLH,AX4,SS2FLH,SD4,IKL,CV4
      WRITE(3,881) S1FLM,AX3M,SS1FLM,SD3M,IKL,CV3M
      WRITE(3,881) S2FLM,AX4M,SS2FLM,SD4M,IKL,CV4M
      WRITE(3,51)
      WRITE(3,884)
  884 FORMAT(2X,'  COMPARE WITH MCG  ')
      WRITE(3,880)
      WRITE(3,881) S1FMMH,AX5,D1FMMH,SD5,IIM,CV5
      WRITE(3,881) S2FMMH,AX6,D2FMMH,SD6,IIM,CV6
      WRITE(3,881) S1FMML,AX5L,D1FMML,SD5L,IIM,CV5M
      WRITE(3,881) S2FMMH,AX6L,D2FMML,SD6L,IIM,CV6M
  881 FORMAT(2X,F15.6,2X,F15.6,2X,F15.6,2X,F15.6,2X,I3,2X,F15.6)
      AJJ = JJ
C     ------------------------------------------------------
      SDATH = SQRT((ATTH - (ATH*ATH))/IKH/(IKH-1))
      SDATL = SQRT((ATTL - (ATL*ATL))/IKL/(IKL-1))
      SDATM = SQRT((ATTM - (ATM*ATM))/IKM/(IIM-1))
      CVATH = SDATH/(ATH/IKH)
      CVATL = SDATL/(ATL/IKL)
      CVATM = SDATM/(ATM/IKM)
      WRITE(3,141) SDATH,CVATH,ATH,ATTH
      WRITE(3,141) SDATL,CVATL,ATL,ATTL
      WRITE(3,141) SDATM,CVATM,ATM,ATTM
  141 FORMAT(2X,'  SUB-SD = ',F15.6,'SUB-CV =',F15.6,2X,'AT ATT'
     *,2X,F15.6,2X,F15.6)
C     ------------------------------------------------------
      XBDHL = (SAM2H - SAM2L)/SQRT(AJJ)
      XBDHM = (SAM2H - SAM2M)/SQRT(AJJ)
      XBDLM = (SAM2L - SAM2M)/SQRT(AJJ)
      XBOLS = S1OLS/AJJ
      XBDH1 = SAM2H/AJJ
      XBDL1 = SAM2L/AJJ
      XBDM1 = SAM2M/AJJ
```

```
      SDOLS = SSOLS - (S1OLS*S1OLS)/AJJ
      SDH1  = SSAM2H - (SAM2H*SAM2H)/AJJ
      SDL1  = SSAM2L - (SAM2L*SAM2L)/AJJ
      SDM1  = SSAM2M - (SAM2M*SAM2M)/AJJ
      SDO11 = SQRT(SDOLS/AJJ-1)
      SDH11 = SQRT(SDH1/(AJJ-1)
      SDL11 = SQRT(SDL1/(AJJ-1)
      SDM11 = SQRT(SDM1/(AJJ-1)
      CVO = SDO11/XBOLS
      CVH = SDH11/XBDH1
      CVL = SDL11/XBDL1
      CVM = SDM11/XBDM1
      ZDHL = XBDHL/SQRT((SDH1 + SDL1)/(AJJ-1))
      ZDHM = XBDHM/SQRT((SDH1 + SDM1)/(AJJ-1))
      ZDLM = XBDLM/SQRT((SDL1 + SDM1)/(AJJ-1))
      WRITE(3,51)
      WRITE(3,887)
  887 FORMAT(2X,'  SUM-X , SUMSQ-X  ===>  HKB,LEE,MCG  ')
      WRITE(3,886)SAM2H,SSAM2H,SAM2L,SSAM2L,SAM2M,SSAM2M
      WRITE(3,51)
      WRITE(3,889)
  889 FORMAT(2X,'XBAR ==> OLS,HKB,LEE,MCG AND SD  ==> OLS,HKD,LEE,MCG ')
      WRITE(3,886)XBOLS,XBDH1,XBDL1,XBDM1,SDO11,SDH11,SDL11,SDM11
      WRITE(3,51)
      WRITE(3,809)
  809 FORMAT(2X,'CV ==> OLS,HKB,LEE,MCG AND ZD ==> HKB,LEE,MCG')
      WRITE(3,886)CVO,CVH,CVL,CVM,ZDHL,ZDHM,ZDLM
  886 FORMAT(2X,7(2X,F15.6))
C     ---------------------------------------------------
      IF(XBDH1 .LE. XBDL1 .AND. XBDH1 .LE. XBDM1) GOTO 73
      IF(XBDL1 .LE. XBDH1 .AND. XBDL1 .LE. XBDM1) GOTO 74
      IF(XBDM1 .LE. XBDH1 .AND. XBDM1 .LE. XBDL1) GOTO 75
   73 TAN1 = XBDL1/XBDH1-1
      TAN11 = TAN1*100
      TAN2 = XBDM1/XBDH1-1
```

```
      TAN22 = TAN2 * 100
      TANA = XBDH1/XBDL1
      TANB = XBDH1/XBDM1
      WRITE(3,703)TAN1,TAN11,TAN2,TAN22,TANA,TANB
  703 FORMAT(2X,'HKB=>MIN',1X,'(L/H)-1 = ', F10.6,1X,F10.6,
     * 1X,'(M/H)-1 =',F10.6,1X,F10.6,2X,'H/L =',F10.6,2X
     * ,'H/M =',F10.6)
       GOTO 737
   74 TAN1 = XBDH1/XBDL1 - 1
      TAN11 = TAN1*100
      TAN2 = XBDM1/XBDL1 - 1
      TAN22 = TAN2*100
      TANA = XBDL1/XBDH1
      TANB = XBDL1/XBDM1
      WRITE(3,704)TAN1,TAN2,TAN22,TANA,TANB
  704 FORMAT(2X,'LEE=>MIN',1X,'(H/L)-1 = ',F10.6,1X,F10.6,
     * 1X,'(M/L)-1 = ',F10.6,1X,F10.6,2X,'L/H =',F10.6,2X
     * ,'L/M =',F10.6)
      GOTO 737
   75 TAN1 = XBDH1/XBDM1 - 1
      TAN11 = TAN1*100
      TAN2 = XBDL1/XBDM1 - 1
      TAN22 = TAN2*100
      TANA = XBDM1/XBDH1
      TANB = XBDM1/XBDL1
      WRITE(3,705)TAN1,TAN11,TAN2,TAN22,TANA,TANB
  705 FORMAT(2X,'MCG=>MIN',1X,'(H/M)-1 = ',F10.6,1X,F10.6,
     * 1X,'(L/M)-1 = ',F10.6,1X,F10.6,2X,'M/H =',F10.6,2X
     * ,'M/L =',F10.6)
  737 TANOH1 = XBDH1/XBOLS - 1
      TANOL1 = XBDL1/XBOLS - 1
      TANOM1 = XBDM1/XBOLS - 1
      TOH = XBOLS/XBDH1
      TOL = XBOLS/XBDL1
      TOM = XBOLS/XBDM1
```

```
      WRITE(3,709)TANOH1,TANOL1,TANOM1,TOH,TOL,TOM
  709 FORMAT(2X,'H/O-1  L/O-1  M/O-1 ==>',3(2X,F8.4),2X,
     * 'O/H  O/L  O/M  ==>',3(2X,F8.4))
C     ------------------------------------------------------
  139 AJJ = JJ
      WRITE(3,8007)AJJ
 8007 FORMAT(2X,F4.0,2X,'ITERATION',/2X,4X,'---------')
      DO 9900 IH=1,MIX
      CMSEH(IH) = TBHKB(IH)/AJJ
      CMSEL(IH) = TBLEE(IH)/AJJ
      CMSEM(IH) = TBMCG(IH)/AJJ
 9900 WRITE(3,1990)CMSEH(IH),CMSEL(IH),CMSEM(IH)
 1990 FORMAT(2X,'CMSEH=',F15.6,2X,'CMSEL=',F15.6,2X,'CMSEM=',F15.6)
      EMSEH = 0.0
      EMSEL = 0.0
      EMSEM = 0.0
      TMSEH = 0.0
      TMSEL = 0.0
      TMSEM = 0.0
      DO 9901 IH=1,MIX
      TMSEH = TMSEH + CMSEH(IH)
      TMSEL = TMSEL + CMSEL(IH)
 9901 TMSEM = TMSEM + CMSEM(IH)
      EMSEH = TMSEH/MIX
      EMSEL = TMSEL/MIX
      EMSEM = TMSEM/MIX
      WRITE(3,8850)
      WRITE(3,8050)EMSEH,EMSEL,EMSEM
 8050 FORMAT(2X,'MSE(HKB) =',F15.6,2X,'MSE(LEE) =',F15.6,2X,'MSE(MCG)='
     * ,F15.6)
      WRITE(3,8850)
      RHL = (TMSEL/TMSEH)*100
      RHM = (TMSEM/TMSEH)*100
      RLM = (TMSEM/TMSEL)*100
      ABRHL = ABS((TMSEH - TMSEL)/TMSEL)
```

```
      ABRHM = ABS((TMSEH - TMSEM)/TMSEM)
      ABRLM = ABS((TMSEL - TMSEM)/TMSEM)
      WRITE(3,8850)
 8850 FORMAT(2X,'----------------------------------------')
      WRITE(3,9350)RHL
 9350 FORMAT(10X,'(MSEL/MSEH)X100 = ',F15.6)
      WRITE(3,3450)RHM
 3450 FORMAT(10X,'(MSEM/MSEH)X100 = ',F15.6)
      WRITE(3,9550)RLM
 9550 FORMAT(10X,'(MSEM/MSEL)X100 = ',F15.6)
      WRITE(3,8850)
      WRITE(3,9251)ABRHL
 9251 FORMAT(10X,'¦(H-L)/L¦ = ',F15.6)
      WRITE(3,9351)ABRHM
 9351 FORMAT(10X,'¦(H-M)/M¦ = ',F15.6)
      WRITE(3,9451)ABRLM
 9451 FORMAT(10X,'¦(L-M)/M¦ = ',F15.6)
   87 CONTINUE
      STOP
      END
```

```
C     ********************************************************
C     ***    FUNCTION NORMAL(DMEANSIGMA DISTRIBUTION    ***
C     ********************************************************
      FUNCTION NORMAL(DMEAN,SIGMA)
      REAL NORMAL
      COMMON /SEED/IX/SELECT/KK
      PI = 3.1415926
      IF (KK.EQ.1) GOTO 10
         CALL RAND(IX,IY,YFL)
         RONE = YFL
         CALL RAND(IX,IY,YFL)
         RTWO = YFL
      ZONE = SQRT(-2*ALOG(RONE))*COS(2*PI*RTWO)
      ZTWO = SQRT(-2*ALOG(RONE))*SIN(2*PI*RTWO)
      NORMAL = ZONE*SIGMA + DMEAN
      KK = 1
      RETURN
   10 NORMAL = ZTWO*SIGMA + DMEAN
      KK = 0
      RETURN
      END
C     ********************************************************
C     ***   FUNCTION WEIBUL(ALPHA1,BETA1) DISTRIBUTION   ***
C     ********************************************************
      FUNCTION WEIBUL(ALPHA1,BETA1)
      COMMON /SEED/IX
      CALL RAND(IX,IY,YFL)
      WEIBULL = BETA1*(-ALOG(1.0-YFL))**(1.0/ALPHA1)
      RETURN
      END
C     ********************************************************
C     ***              SUBROUTINE  RANDOM                  ***
C     ********************************************************
      SUBROUTINE RAND(IX,IY,YFL)
      IY = IX*65539
```

```
      IF(IY) 5,6,6
    5 IY = IY + 2147483647 + 1
    6 YFL = IY
      YFL = YFL/2147483647
      IX = IY
      RETURN
      END
C     ***********************************************************
C     ***                SUBROUTINE  INIT                     ***
C     ***********************************************************
      SUBROUTINE INIT
      REAL NORMAL,LAMBDA(11)
      DOUBLE PRECISION B,X,A,S,XA
      COMMON /SEED/IX/DATAXY/X(12,150)/COEFF/B(11)
     *       /CONTA/P,IC/INTERV/XBAR1(11),SG(11)
     *       /SKEWD/ALPHA,BETA
     *       /SELECT/KK/VARIAVB/N,M,M2
     *       /REGRS/A(11,11),S(12,12)
     *       /EIGEN/ALAMP(5)
     *       /PX/F(5,5)
     *       /RINIT/ALP1,ALP2
      DIMENSION XA(10,100),Q(5,5),XP(6,100),QQ(6,100),SSB(6)
     *       ,AI(20,20)
C     ==========  GENERRATE  INDEPENDENT  VARIABLE  =======
      XBAR1(M2) = 0.0
      SG(M2) = 1
      DO 100 I=1,M2
      DMEAN = XBAR1(I)
      SIGMA = SG(I)
      DO 100 J=1,N
      XA(I,J) = NORMAL(DMEAN,SIGMA)
  100 CONTINUE
      ALMIX1 = SQRT(ALP1)
      ALMIX2 = SQRT(ALP2)
      DO 101 I=1,3
```

```
      DO 101 J=1,N
      X(I,J) = SQRT(1-ALP1)*XA(I,J) + ALP1*XA(M2,J)
  101 CONTINUE
C     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     :::          FOR 3 INDEPENDENT VARIABLE                :::
C     :::        PUT C IN DO LOOP 102 (4 LINES)              :::
C     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      DO 102 I=4,5
      DO 102 J=1,N
      X(I,J) = SQRT(1-ALP2)*XA(I,J) + ALP2*XA(M2,I)
  102 CONTINUE
C     ----------------------------------------------------------
C     ---                  CALL CORRETION                     ---
C     ----------------------------------------------------------
      DO 88 I=1,M
      DO 88 J=1,N
   88 XP(I,J) = X(I,J)
      CALL CORR(XP,M,N,QQ,SSB)
C     ========================
      DO 110 I=1,M
      DO 110 J=1,M
      F(I,J) = QQ(I,J)
      F(J,I) = QQ(I,J)
      Q(I,J) = QQ(I,J)
  110 Q(J,I) = QQ(I,J)
      SCB = 0.0
      DO 1140 JK=1,M
      CALL RAND(IX,IY,YFL)
      CB = 0.0
      CB = YFL
      SCB = SCB + CB*CB
 1140 B(JK) = CB
      DO 1969 JM=1,M
 1969 B(JM) = B(JM)/SQRT(SCB)
```

```
C     ------------------------------------------------------
C     --- FOR MAIN 1  ===> PUT C IN LOOP 120 (7 LINES)  ---
C     ------------------------------------------------------
C     ::::::::::::::    X'X   AND   X'Y   ::::::::::::::::
      DO 120 I=1,M
      DO 120 K=1,M
      SIK = 0.0
      DO 121 J=1,N
  121 SIK = SIK + X (I,J) * X (K,J)
      S(I,K) = SIK
  120 S(K,I) = SIK
C     ------------------------------------------------------
C     -    FOR MAIN1 ==> CHANGE MATRIX  S TO MATRIX QQ     -
C     -                ( IN LOOP 140 ONLY)                 -
C     ------------------------------------------------------
C     ::::::::::::::  INVERSE  MATRIX OF X'X  ::::::::::::
  600 DO 140 I=1,M
      DO 140 J=1,M
      A(I,J) = S(I,J)
  140 A(J,I) = S(I,J)
      DO 145 K=1,M
      IF (A(K,K)) 145,146,145
  146 WRITE(3,150)
  150 FORMAT('A(K,K) HAS ZERO ON DIAGONAL ')
      STOP
  145 CONTINUE
      CALL INV(M,A)
```

```
C     ------------------------------------------------------
C     -      FOR MAIN1    ===>  DELETE CA IN COLUMN 1      -
C     ------------------------------------------------------
CA    DO 759 I=1,M
CA    DO 759 J=1,M
CA759 AI(I,J) = A(I,J)
CA    CALL LAMP(AI,LAMBDA,M)
CA    ALAMP(M) = LAMBDA(1)
C     ------------------------------------------------------
      RETURN
      END
C
C     ************************************************************
C     ***                  SUBROUTINE  INIT1                   ***
C     ************************************************************
      SUBROUTINE INIT1
      REAL NORMAL,LAMBDA(11)
      DOUBLE PRECISION B,X,A,S,XA
      COMMON /SEED/IX/DATAXY/X(12,150)/COEFF/B(11)
     *       /CONTA/P,IC/INTERV/XBAR1(11),SG(11)
     *       /SKEWD/ALPHA,BETA
     *       /SELECT/KK/VARIAVB/N,M,M2
     *       /REGRS/A(11,11),S(12,12)
     *       /EIGEN/ALAMP(5)
     *       /PX/F(5,5)
      DIMENSION AI(20,20)
  600 DO 140 I=1,M
      DO 140 J=1,M
      A(I,J) = 0.0
      A(I,J) = F(I,J)
  140 A(J,I) = F(I,J)
      DO 145 K=1,M
      IF (A(K,K)) 145,146,145
  146 WRITE(3,150)
```

```
  150 FORMAT(' A(K,K) HAS ZERO ON DIANGNAL ')
      STOP
  145 CONTINUE
      CALL INVS(M,A)
      DO 6759 I=1,M
      DO 6759 J=1,M
 6759 AI(I,J) = A(I,J)
      CALL LAMP(AI,LAMBDA,M)
      ALAMP(M) = LAMBDA(1)
      RETURN
      END
C
C     ************************************************************
C     ***                  SUBROUTINE DATA                     ***
C     ************************************************************
      SUBROUTINE DATA
      REAL NORMAL
      DOUBLE PRECISION B,X,Y,E,T
      COMMON /SEED/IX/COEFF/B(11)/DIST/II
     *       /CONTA/P,IC/INTERV/XBAR1(11),SG(11)
     *       /SKEWD/ALPHA,BETA
     *       /DATAXY/X(12,150)/DATAY/Y(150)
     *       /SELECT/KK/VARIAVB/N,M,M2
     *       /OYY/OYY,SSN
      DIMENSION E(150),T(150)
      GOTO (10,20,30,50),II
   10 DMEAN = XBAR1(M)
      SIGMA = SG(M)
      SG2   = IC*SIGMA
      DO 15 J=1,N
      CALL RAND(IX,IY,YFL)
      IF (YFL - P) 12,12,13
   12 E(J) = NORMAL(DMEAN,SG2)
      GO TO 15
   13 E(J) = NORMAL (DMEAN,SIGMA)
```

```
 15 CONTINUE
    GOTO 60
 20 DMEAN = 0.0
    SIGMA = 1.0
    SSN =SIGMA
    DO 25 J=1,N
    E(J) = NORMAL(DMEAN,SIGMA)
 25 CONTINUE
    GOTO 60
 30 SIGMA = SQRT(.70)
    DO 35 J=1,N
    DMEAN1 = 0.0
    SNN = SIGMA**2
    X(M2,J) = EXP(NORMAL(DMEAN1,SIGMA))
 35 CONTINUE
    GOTO 60
 50 DO 55 J=1,N
    X(M2,J) = WEIBUL(ALPHA,BETA)
 55 CONTINUE
 60 GOTO (100,100,110,110),II
100 DO 108 J=1,N
    SUM = 0.0
    DO 105 I =1,M
    SUM = SUM + X(I,J)*B(I)
105 CONTINUE
    X(M2,J) = SUM +E(J)
108 CONTINUE
    GOTO 1209
110 DO 115 J=1,N
115 Y(J) = X(M2,J)
1209 OYY = 0.0
    DO 879 J=1,N
879 OYY = OYY  + X(M2,J)*X(M2,J)
    RETURN
    END
```

```
C     ********************************************************
C     ***              SUBROUTINE INVERSE MATRIX           ***
C     ********************************************************
      SUBROUTINE INVS(M,A)
      DOUBLE PRECISION A(11,11)
      DO 20 K=1,M
      A(K,K) = -1.0/A(K,K)
      DO 5 I=1,M
      IF (I-K) 3,5,3
    3 A(I,K) = - A(I,K)*A(K,K)
    5 CONTINUE
      DO 10 I=1,M
      DO 10 J=1,M
      IF ((I - K)*(J - K)) 9,10,9
    9 A(I,J) = A(I,J) - A(I,K)*A(K,J)
   10 CONTINUE
      DO 20 J=1,M
      IF (J-K) 18,20,18
   18 A(K,J) = -A(K,J)*A(K,K)
   20 CONTINUE
      DO 25 I=1,M
      DO 25 J=1,M
   25 A(I,J) = -A(I,J)
      RETURN
      END
C
C     ********************************************************
C     ***                  SUBROUTINE LAMP                 ***
C     ********************************************************
      SUBROUTINE LAMP(AI,LAMBDA,M)
      REAL LAMBDA(11)
      INTEGER SLL,SP1
      DIMENSION AI(20,20),XLL(20),D(20),ZLL(11,20),XKLL(11,20),CLL(11),
     * G(10),SUMMLL(20),YLL(20),X1LL(11,20),XXLL(20),XK1LL(11,20)
      DIFF = 0.0
```

```
      DO 496 I=1,11
      LAMBDA(I) = 0.0
      CLL(I) = 0.0
      DO 496 J=1,20
      YLL(J)  = 0.0
      XLL(J)  = 0.0
      XXLL(J) = 0.0
      ZLL(I,J) = 0.0
      XK1LL(I,J) = 0.0
  496 X1LL(I,J) = 0.0
      NLL = M
      SLL = M-1
      EPSI = 0.1
      DO 4 I=1,NLL
    4 XLL(I) = 1
C     ------------------------------------------------------
C     ==  CALCULATE COMPONENTS OF THE VECTOR 1/LAMBDA*X  ==
C     ------------------------------------------------------
      IT =  0
    5 DO 6 I=1,NLL
      D(I) = 0
      DO 6 J=1,NLL
    6 D(I) = D(I) + AI(I,J)*XLL(J)
      IT = IT + 1
C     ------------------------------------------------------
C     ==              NORMALIZE THE VECTOR D(I)           ==
C     ------------------------------------------------------
      DO 7 I=1,NLL
    7 ZLL(1,I) = D(I)/D(1)
C     ------------------------------------------------------
C     ==CHECK TO SEE IF REQUIRED ACCURACY HAS BEEN OBTAIN==
C     ------------------------------------------------------
      DO 8 I=1,NLL
      DIFF = XLL(I) - ZLL(1,I)
      IF (ABS(DIFF).GT.EPSI*ZLL(1,I)) GOTO 9
```

```
    8 CONTINUE
      GOTO 11
    9 DO 10 I=1,NLL
   10 XLL(I) = ZLL(1,I)
      IF (IT.GE.50) GOTO 11
      GOTO 5
   11 LAMBDA(1)=1./D(1)
C     ==  THE SMALLEST EIGENVALUE AND ASSOCIATED VECTOR  ==
C     ==          ARE NOW SWEEP OUT S EIGENVECTOR        ==
      DO 100 I=1,SLL
      IP1 = I + 1
C     ==          ASSIGN THE VALUE1 TO XK COMPONENT      ==
      DO 12 L=1,SLL
   12 XKLL(IP1,L) = 1
C     ==                CALCULATE G(I) VALUES            ==
      G(I) = 0.0
      DO 14 L=1,NLL
   14 G(I) = G(I) + ZLL(I,L)**2
C     ==                CALCULASTE VALUE OF C            ==
      IT = 0
   15 DO 99 II=1,I
      SUMLL = 0.0
      DO 16 L=1,NLL
   16 SUMLL = SUMLL + ZLL(II,L)*XKLL(IP1,L)
   99 CLL(II) = SUMLL/G(II)
      IT = IT + 1
C     ==                CALCULATE SUM OF CX'S            ==
      DO 17 L=1,NLL
   17 SUMMLL(L) = 0.0
      DO 19 K=1,I
      DO 18 L=1,NLL
      YLL(L) = CLL(K)*ZLL(K,L)
   18 SUMMLL(L) = SUMMLL(L) + YLL(L)
   19 CONTINUE
```

```
C     ------------------------------------------------------
C     ==                CALCULATE NEW TRAIL X               ==
C     ------------------------------------------------------
      DO 20 L=1,NLL
   20 X1LL(IP1,L) = XKLL(IP1,L) - SUMMLL(I)
C     ------------------------------------------------------
C     ==              MULTIPLY A INVERSE TIMES X            ==
C     ------------------------------------------------------
      DO 21 II=1,NLL
      XXLL(II) = 0.0
      DO 21 KK=1,NLL
   21 XXLL(II) = XXLL(II)+AI(II,KK)*X1LL(IP1,KK)
C     ------------------------------------------------------
C     ==NORMALIZE TO GET XK FOR NEXT ITERATION,OR FINAL XK==
C     ------------------------------------------------------
      DO 22 L=1,NLL
   22 XK1LL(IP1,L) = XXLL(L)/XXLL(1)
C     ------------------------------------------------------
C     ==CHECK TO SEE IF REQUIRED ACCURACY HAS BEEN ATTAINED==
C     ------------------------------------------------------
      DO 23 L=1,NLL
      DIFF = XKLL(IP1,L) - XK1LL(IP1,L)
      IF (ABS(DIFF).GE.EPSI*XK1LL(IP1,L)) GOTO 24
   23 CONTINUE
      GOTO 26
C     ------------------------------------------------------
C     ==              ASSIGN XK1 VALUES TO NAME XK          ==
C     ------------------------------------------------------
   24 DO 25 L=1,NLL
   25 XKLL(IP1,L) = XK1LL(IP1,L)
      IF (IT.GE.50) GOTO 26
      GOTO 15
```

```
C     ------------------------------------------------------
C     ==                  CALCULATE EIGENVALUE             ==
C     ------------------------------------------------------
   26 LAMBDA(IP1) = 1./XXLL(1)
C     ------------------------------------------------------
C     ==              ASSIGN EIGENVECTOR TO NAME Z         ==
C     ------------------------------------------------------
      DO 27 L=1,NLL
   27 ZLL(IP1,L) = XK1LL(IP1,L)
  100 CONTINUE
C     WRITE OUT THE EIGENVALUES AND EIGENVECTORS
C     SP1=SLL+1
C     DO 104 i=1,SP1
C     WRITE(3,101) I,LAMBDA(I)
C 101 FORMAT(2X,'LAMBDA =',F14.7)
C     WRITE(3,102)
C 102 FORMAT(2X,'THE ASSOIATED EIGENVECTOR COMPONENTS ARE'/)
C     WRITE(3,103) (ZLL(I,L),L=1,NLL)
C 103 FORMAT(13X,F120.9)
C 104 CONTINUE
      RETURN
      END
C
C     ********************************************************
C     ***              SUBROUTINE LEAST SQURE(B)           ***
C     ********************************************************
      SUBROUTINE OLS(B2,TOT)
      DOUBLE PRECISION A,S,X,B2
      COMMON /REGRS/A(11,11),S(12,12)/VARIAB/N,M,M2
     *       /DATAXY/X(12,150)
      DIMENSION B2(11)
      DO 40 J=1,M
      S(M2,J) = 0.0
   40 S(M2,J) = X(M2,J)
      TOT =1.0
```

```
      DO 50 I = 1,M
      B2(I) = 0.0
      DO 50 J=1,M
   50 B2(I) = B2(I) + A(J,I)*S(M2,J)
      RETURN
      END
C
C     ************************************************************
C     ***              SUBROUTINE CORRELATION                  ***
C     ************************************************************
      SUBROUTINE CORR(XP,M,N,QQ,SSB)
      COMMON /SDRID/TMEAN(6),SSBRID(6)
      DIMENSION XP(6,100),SUM(6),SSA(6),QQ(6,6),
     *          SE(6,6),XX(6,100)
      DO 30 I=1,M
      SUM(I) = 0.0
      DO 30 J=1,N
   30 SUM(I) = SUM(I) + XP(I,J)
      DO 40 I=1,M
   40 TMEAN(I) = SUM(I)/N
      DO 50 I=1,M
      SSA(I) = 0.0
      DO 50 J=1,N
   50 XX(I,J) = XP(I,J) - TMEAN(I)
      DO 60 I=1,M
      SSB(I) = 0.0
      DO 60 J=1,N
   60 SSA(I) = SSA(I) + XX(I,J)*XX(I,J)
C     ::::::::::::        CALCULATE X'X        ::::::::::::
      DO 71 I=1,M
   71 SSB(I) = SQRT(SSA(I)/(N-1))
      DO 70 I=1,M
      DO 70 K=1,M
      SIK = 0.0
      DO 80 J=1,N
```

```
   80 SIK = SIK + XX(I,J)*XX(K,J)
      SE(I,K) = SIK
   70 SE(K,I) = SIK
      DO 90 I=1,M
      DO 90 J=1,M
   90 QQ(I,J) = SE(I,J)/SQRT(SSA(I)*SSA(J))
      RETURN
      END
C     ********************************************************
C     ***                SUBROUTINE BOX-COX                ***
C     ********************************************************
      SUBROUTINE BOXCOX(B,FL1)
      DOUBLE PRECISION Y1,Y,B,SLG,BMEST
      COMMON /DATAY/Y(150)/TRANSF/Y1(150),FL(3),ST,FIN
     *       /VARIAB/N,M,M2/STORE/BMEST(3,11)
     *       /DATXY1/X1(12,150)
      DIMENSION B(11)
      DO 30 J=1,N
      IF (Y(J)) 20,20,30
   20 WRITE(3,25)
   25 FORMAT('Y(J) IS NEGATIVE OR ZERO ==> RETURN TO MAIN PROGRAM ')
      RETURN
   30 CONTINUE
      SLG = 0.0
      DO 50 J=1,N
   50 SLG = SLG + DLOG(Y(J))
      SLG = SLG/N
      G = DEXP(SLG)
      DO 60 J=1,N
   60 Y1(J) = Y(J)/G
      ST = -1.0
      FIN = 2.0
      FD = 0.5
      MR = 16
      CALL SUMSQ(ST,SSE1,1)
```

```
      CALL SUMSQ(FD,SSE2,2)
      CALL SUMSQ(FIN,SSE3,3)
   70 IF(SSE1.LE.SSE2 .AND. SSE1.LE.SSE3) GOTO 71
      IF(SSE2.LE.SSE1 .AND. SSE2.LE.SSE3) GOTO 72
      IF(SSE3.LE.SSE1 .AND. SSE3.LE.SSE2) GOTO 73
   71 FM = ST
      SSE2 = SSE1
      DO 80 I=1,M
   80 BMEST(2,I) = BMEST(1,I)
      GOTO 100
   72 FM = FD
      GOTO 100
   73 FM = FIN
      SSE2 = SSE3
      DO 90 I=1,M
   90 BMEST(2,I) = BMEST(3,I)
  100 IF (MR .EQ. 1) GOTO 110
      MR = MR/2
      ST = FM - MR*0.1
      FD =FM
      FIN = FM + MR*0.1
      CALL SUMSQ(ST,SSE1,1)
      CALL SUMSQ(FIN,SSE3,3)
      GOTO 70
  110 FL1 = FM
      DO 120 I=1,M
  120 B(I) = BNEST(2,I)
      FL(1) = FM
      RETURN
      END
```

```
C     ********************************************************
C     ***                  SUBROUTINE SUMSQ                ***
C     ********************************************************
      SUBROUTINE SUMSQ(FLX,SSE,K)
      DOUBLE PRECISION Y1,A,S,Y,B,X1,BMEST,A1,S1,Z,ZZ,FL
      COMMON /REGRS/A(11,11),S(12,12)
     *       /REGES1/A1(11,11),S1(12,12)
     *       /DATXY1/X1(12,150)/DATAY/Y(150)/STORE/BMEST(3,11)
     *       /TRANSF/Y1(150),FL(3),ST,FIN,VARIAB/N,M,M2
      DIMENSION B(11),Z(11)
      FL(1) = FLX
      CALL BCOX(FL(1),1)
      CALL OLS1(B)
      DO 15 I=1,M
   15 Z(I) = S(M2,I)
      ZZ   = S(M2,M2)
      DO 80 I=1,M
   80 BMEST(K,I) = B(I)
      SSR = 0.0
      DO 90 I=1,M
   90 SSR = SSR + B(I)*Z(I)
      SSE = ZZ - SSR
      SSE = SSE/FLOAT(N)
      RETURN
      END
C     ********************************************************
C     ***                  SUBROUTINE BCOX                 ***
C     ********************************************************
      SUBROUTINE BCOX(FLI,K)
      DOUBLE PRECISION Y1,X1,FL,FLI
      COMMON /DATXY1/X1(12,150)
     *       /TRANSF/Y1(150),FL(3),ST,FIN/VARIAB/N,M,M2
      FL(K) = FLI
      IF (DABS(FL(K))) 15,5,15
    5 DO 10 I=1,N
```

```
   10 X1(M2,I) = DLOG(Y1(I))
      GOTO 30
   15 DO 20 I=1,N
   20 X1(M2,I) = ((Y1(I)**FL(K)) - 1.0)/FL(K)
   30 RETURN
      END
C     ********************************************************
C     ***                  SUBROUTINE OLS                  ***
C     ********************************************************
      SUBROUTINE OLS1(B2)
      DOUBLE PRECISION A,S,X1,B2
      COMMON /REGRS/A(11,11),S(12,12)/VARIAB/N,M,M2
     *       /DATXY1/X1(12,150)
      DIMENSION B2(11)
      DO 20 I=1,M2
      SIK = 0.0
      DO 10 J=1,N
   10 SIK = SIK + X1(I,J)*X1(M2,J)
   20 S(M2,I) = SIK
      DO 50 I=1,M
      B2(I) = 0.0
      DO 50 J=1,M
   50 B2(I) = B2(I) + A(J,I)*S(M2,J)
      RETUEN
      END
C     ********************************************************
C     ***                SUBROUTINE SHAPWK                 ***
C     ********************************************************
      SUBROUTINE SHAPWK(N,Y,ITEST)
      DOUBLE PRECISION A2,SW,Y,S2
      COMMON /TABLE/A2(150),SW(2)
      DIMENSION Y(150),S(11)
      CALL RANK(N,Y)
      YSUM = 0.0
      YSS = 0.0
```

```
      DO 5 I=1,N
      YSUM = YSUM + Y(I)
    5 YSS = YSS + Y(I)*Y(I)
      S2 = YSS - (YSUM*YSUM/FLOAT(N))
      K = INT(FLOAT(N/2)
      B = 0.0
      DO 20 I=1,K
      JJ = N - I + 1
   20 B = B + A2(JJ)*(Y(JJ) - Y(I))
      W = B*B/S2
      IF (W-SW(3)) 30,30,40
   30 ITEST = 2
      GOTO 50
   40 ITEST = 1
   50 RETURN
      END
C     ***********************************************************
C     ***                SUBROUTINE  RANK                     ***
C     ***********************************************************
      SUBROUTINE RANK(N,X)
      DOUBLE PRECISION X,T
      DIMENSION X(150)
      N1 = N - 1
      DO 10 I=1,N1
      II = I + 1
      DO 10 K=II,N
      IF (X(I) .LE. X(K)) GOTO 10
            T = X(I)
            X(I) = X(K)
            X(K) = T
   10 CONTINUE
      RETURN
      END
```

```
C     ********************************************************
C     ***                SUBROUTINE RID                    ***
C     ********************************************************
      SUBROUTINE RID
      DOUBLE PRECISION A,S,B,X,B2,B1
      COMMON /SEED/IX/DATAXY/X(12,150)/COEFF/B(11)
     *       /CONTA/P,IC/INTERV/XBAR1(11),SG(11)
     *       /SKEWD/ALPHA/BETA
     *       /SELECT/KK/VARIAVB/N,M,M2
     *       /REGRS/A(11,11),S(12,12)
     *       /EIGEN/ALAMP(5)
     *       /PX/F(5,5)
     *       /DIST/II
     *       /CON/C,SS,TOT,TRACE,BBRID
     *       /SDRID/TMEAN(6),SSBRID(6)
     *       /SUMDB/SQDIFB(6)
     *       /RCOEFF/BRCHA(6)
C     ==      FOR MAIN1  ==>  DELETE CA FROM COL.1 AND    ==
C     ==              ADD TSEB,TSEBR,IN  /CON/            ==
CA           /UYY/OYY,SSN
CA           /RIDMSE/TRVAB(1)
      DIMENSION XP(6,100),QQ(6,100),SSB(6),VA(5,5),BRID(5,1),
     *       B2(11),B1(11),T(150),A2(150),MSE(5),TX(5,5)
C     ==      FOR MAIN1  ==> DELETE CA FROM COL.1         ==
CA    *       ,ZA(5,5),VB(5,5),QA(5,5),VV(5,5),ZK(5,5),TS(5,1)
CA    *       ,W(5,5),VAZ(5,5)
      MIX = M + 1
      B(MIX) = 0.0
      DO 2100 I=1,M
      DO 2100 J=1,M
      IF (I-J) 2200,2300,2200
 2200 TX(I,J) = F(I,J)
      GOTO 2100
 2300 TX(I,J) = F(I,J) + C
```

```
2100 CONTINUE
     DO 2500 I=1,M
     DO 2500 J=1,M
2500 A(I,J) = TX(I,J)
     CALL INVS(M,A)
     DO 4000 I=1,M
     BBRID(I,1) = 0.0
     BRID(I,1) = 0.0
     DO 4000 K=1,M
4000 BRID(I,1) = BRID(I,1) + A(I,K)*S(M2,K)
     DO 4200 IK=1,M
     BBRID = BBRID + BRID(IK,1)*BRID(IK,1)
     BRCHA(IK) = (BRID(IK,1)*SSBRID(MIX))/SSBRID(IK)
4200 CONTINUE
     TAFTER = 0.0
     DO 8200 IK=1,M
8200 TAFTER = TAFTER + BRCHA(IK)*TMEAN(IK)
     BRCHA(MIX) = TMEAN(MIX) - TAFTER
C    ..............        FINE MSE       ..................
     DO 8400 IK=1,MIX
     SQDIFB(IK) = 0.0
8400 SQDIFB(IK) = (BRCHA(IK) - B(IK))**2
     RETURN
     END
```

ประวัติผู้เขียน

นางสาวเจษฎาพร ยุทธนวิบูลย์ชัย เกิดเมื่อวันที่ 28 กรกฎาคม พ.ศ.2504 จบการศึกษาระดับปริญญาตรีจากมหาวิทยาลัยศิลปากร สาขาสถิติ คณะวิทยาศาสตร์ เมื่อปี พ.ศ. 2527 และจบการศึกษาระดับปริญญาตรีจากมหาวิทยาลัยรามคำแหง สาขาการโฆษณาและประชาสัมพันธ์ คณะบริหารธุรกิจ เมื่อปี พ.ศ.2530 ได้เข้าศึกษาในภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ.2530