NASA Technical Memorandum 78704

(NASA-TM-78704) A COMPUTER PROGRAM FOR ESTIMATION FROM INCOMPLETE MULTINOMIAL DATA (NASA) 77 p HC A05/MF A01 CSCL 09B

N80-27109

Unclas
G3/61 23583

A COMPUTER PROGRAM FOR ESTIMATION FROM INCOMPLETE MULTINOMIAL DATA

Karen R. Credeur

May 1978

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

A COMPUTER PROGRAM FOR ESTIMATION FROM INCOMPLETE MULTINOMIAL DATA

By

Karen R. Credeur

NASA

LANGLEY RESEARCH CENTER

HAMPTON, VIRGINIA

SUMMARY

This paper presents a computer program for maximum likelihood and Bayesian estimation of the vector $\underset{\sim}{p}$ of multinomial cell probabilities from incomplete data. Also included is coding to calculate exact and approximate elements of the posterior mean and covariance matrices. The program is written in FORTRAN IV language for the Control Data CYBER 170 series digital computer system with network operating system (NOS) 1.1. The program requires approximately 44000 octal locations of core storage. A typical case requires from 72 seconds to 92 seconds on CYBER 175 depending on the value of the prior parameter.

A COMPUTER PROGRAM FOR ESTIMATION FROM INCOMPLETE MULTINOMIAL DATA

By

Karen R. Credeur

TABLE OF CONTENTS

DESCRIPTION

This paper describes the main computer program used in reference 1 for estimating the vector $\underset{\sim}{p}$ of multinomial cell probabilities from incomplete data. The data is incomplete in that it contains partially classified observations. Each such partially classified observation is observed to fall in one of two or more selected categories but is not classified further. The estimation criterion is minimization of risk for quadratic loss $L(\underset{\sim}{p}-\dot{\underset{\sim}{p}}) = (\underset{\sim}{p}-\dot{\underset{\sim}{p}})'(\underset{\sim}{p}-\dot{\underset{\sim}{p}})$ for $\dot{\underset{\sim}{p}}$ an estimator of $\underset{\sim}{p}$.

In addition, elements of the posterior mean and covariance matrices are calculated exactly and approximately. A Taylor-series function is used to approximate the posterior covariance matrix. A Taylor-series function, the maximum-likelihood estimate, and the posterior mode are used to approximate the posterior mean.

Monte-Carlo simulation studies are performed for small- and medium-size samples to assess

(1) which of the maximum-likelihood estimate, posterior mode, and Taylor-series approximate posterior mean best minimizes risk for specified values of $\underset{\sim}{p}$;

(2) how well each of these functions approximates the exact posterior mean; and

(3) how well a Taylor-series function approximates elements of the posterior covariance matrix.

Samples are of size 25 and 50, percentage of incomplete data varies around 15 and 40, and probabilities range from the center of the probability simplex P_2 to one of its corners. Probabilities equal the means of the prior distributions for varying parameters or are randomly generated from these distributions. An exploratory robustness study is conducted by using the correct prior, a uniform prior, and a perturbed prior in the Bayesian estimators. The iterative algorithm of Dempster, Laird, and Rubin (ref. 2) is used to evaluate all three estimators.

Other discussion, analysis, and results are given in reference 1. Included in the discussion in reference 1 are descriptions of pseudorandom-number generators for the Dirichlet, uniform, and trinomial distributions. Also given are tree diagrams (figs. 5.1 - 5.3 of ref. 1) that illustrate the flow of the computer program.

The computer used is a Control Data Corporation (CDC) CYBER 170 series digital computer system with network operating system (NOS) 1.1. This computer operates with a 60-bit word and single-precision accuracy of about 14.5 significant figures. The programing language is FORTRAN Extended, Version 4.6. The program requires approximately 44000 octal locations of core storage. A typical case requires from 72 seconds to 92 seconds depending on the prior parameter.

A listing of the main computer program is given in the third section. An index precedes the listing. Symbols are defined in the listing. Note that the program is written for DESIGN 2 of reference 1 but can be modified to be DESIGN 1 of reference 1 by deleting the Dirichlet-generation level and changing the dimensioning of the QLMSIJ matrix. DESIGN 1 has a fixed-effects model constituting full factorials ($4 \times 2^2 \times 3$) having four levels* and two replications per cell. DESIGN 2 has a mixed-effects model constituting nested factorials ($4\text{-}10 \times 2^2 \times 3$) having four levels and two replications per cell within each of four variations of the prior parameter $\underset{\sim}{\nu}$. The ten generations of the Dirichlet probability $\underset{\sim}{p}$ in DESIGN 2 are considered random; the remaining factors are considered fixed.

---

*level 1: four Dirichlet probabilities $\underset{\sim}{p}$
level 2: two sample sizes
level 3: two percentages of incomplete data
level 4: three estimators

To run a case, any necessary changes are made to the following lines in MAIN

```
XNU1 =
XNU2 =
  IP =
SEED =
```

In addition, the subroutine GAM, which is a function of the prior parameter ν (NU), and the Hollerith labels

```
DATA ALABEL/10HA.  NU = (,10H0.1,0.1,9.,10H8).        /
DATA TLABEL/10HTABLE 7.1 /
```

are changed as needed. For values of y that are less than 10, GAM(y+1) is calculated from the relationship GAM(y+1) = y GAM(y) and a starting value. For y an integer, the starting value is GAM(2) = 1 and for y = x 1/3, for x an integer, the starting value is GAM(3 1/3) = 2.7781584804296. For y greater than or equal to 10, Stirling's formula is used to approximate the gamma function to 11 significant figures of accuracy.

Because a MODIFY system (ref. 3) is used to maintain the program on a permanent file, a new case is easier to make by changing lines of code rather than reading data cards. Outputs consist of printouts and tapes. Some tapes are directly used as tables. Tapes are also usually input to canned programs for calculating analyses of variance and to a program for summing biases or mean squared error over replication, sample size, percentage of incomplete data, and/or generated Dirichlet probability.

Subroutines URAN, URANV, and MATINV shown in the coding are from the NASA, Langley Research Center, mathematics computer library. They are described in Appendices A, B, and C, respectively. Subroutine URAN gives a single uniform random number according to the algorithm described in Appendix A. Subroutine URANV gives a vector of uniform random numbers from URAN. Subroutine MATINV solves a system of simultaneous linear equations.

In addition, other computer programs not given in the listing have been written. Among these are programs to test the gamma, Dirichlet, trinomial, and uniform pseudorandom-number generators; to calculate analyses of variance; and to sum mean squared errors and biases over one or more of replication, percentage of incomplete data, sample size, and generated Dirichlet probability. The sums have been used for plots in reference 1. Note that a number of subroutines from IMSL (International Mathematical and Statistical Libraries, Inc.; ref. 4) have been used in calculating the analyses of variance.

INDEX OF COMPUTER PROGRAM

LISTING OF COMPUTER PROGRAM

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5,TAPE12)
C
C
C
C     PROGRAMED BY KAREN RACKLEY CREDEUR,  SPRING 1977,   CDC 6600,
C     FORTRAN EXTENDED VERSION 4.6, NASA, LANGLEY RESEARCH CENTER
C
      INTEGER COVSKIP
      REAL MSE(6,7)
      DIMENSION BESTQL(4,2,3),CTSEQL(3)
      DIMENSION R(2,7,2),TUKEY(10),UU(50)
      DIMENSION QLMS1(2,2,2),QLMS2(2,2,2),QLMS3(2,2,2),QLMS4(2,2,2)
      DIMENSION QLMS5(2,2,2),QLMS6(2,2,2),EBIAS1(2,2,2),EBIAS2(2,2,2)
      DIMENSION EBIAS3(2,2,2),EMS1(2,2,2),EMS2(2,2,2),EMS3(2,2,2)
      DIMENSION QLMS11(10,2,2),QLMS12(10,2,2),QLMS21(10,2,2)
      DIMENSION QLMS31(10,2,2),QLMS32(10,2,2),QLMS22(10,2,2)
      DIMENSION QLMS41(10,2,2),QLMS42(10,2,2),QLMS51(10,2,2)
      DIMENSION QLMS61(10,2,2),QLMS62(10,2,2),QLMS52(10,2,2)
      DIMENSION T11(2,2,11),T12(2,2,11),T21(2,2,11),T22(2,2,11)
      DIMENSION T31(2,2,11),T32(2,2,11),T41(2,2,11),T42(2,2,11)
      DIMENSION T51(2,2,11),T52(2,2,11),T61(2,2,11),T62(2,2,11)
      DIMENSION ALABEL(3),TLABEL(1)
      COMMON DEP(3,3),DQL(4,3),E(4,3),IROBUST,NTS,P1,P2,P3,TPID
      COMMON/BEST/BESTEP(3,2),CTRDEP,CTRDQL(3),PRDEP(9,3),PRDQL(9,7),SBI
     1ASEP(3,3),SBIASQL(3,7)
      COMMON/BIASRD/COUNTB(3,8),COUNTRD(8,8)
      COMMON/CALEST/APMC11,APMC12,APMC22,CONVCRI,COVSKIP,DMLC1,DMLC2,DML
     1C3,DPID,EAPMC11,EAPMC12,EAPMC22,EPMC11,EPMC12,EPMC22,ISTOP,N12,N13
     2,N23,PID,PMLC1,PMLC2,PMLC3,SS,SSN,TIMAP,TIMEP,TIM(2),TIM21,TIM31,X
     3NU1,XNU2,XNU3,Z1,Z2,Z3,Z12,Z13,Z23,Z1N,Z2N,Z3N
      COMMON/ITKT/AVNUMIT(6),CTNUMIT(6,10)
      EQUIVALENCE (E(1,1),PEPM1),(E(1,2),PEPM2),(E(1,3),PEPM3)
      EQUIVALENCE (E(2,1),PMLE1),(E(2,2),PMLE2),(E(2,3),PMLE3)
      EQUIVALENCE (E(3,1),PPMD1),(E(3,2),PPMD2),(E(3,3),PPMD3)
      EQUIVALENCE (E(4,1),PAPM1),(E(4,2),PAPM2),(E(4,3),PAPM3)
      EQUIVALENCE (DEP(1,1),EML1),   (DEP(1,2),EML2),   (DEP(1,3),EML3)
      EQUIVALENCE (DEP(2,1),EPMD1),  (DEP(2,2),EPMD2),  (DEP(2,3),EPMD3)
      EQUIVALENCE (DEP(3,1),EAPMN1),(DEP(3,2),EAPMN2),(DEP(3,3),EAPMN3)
      EQUIVALENCE (DQL(1,1),DEPMN1),(DQL(1,2),DEPMN2),(DQL(1,3),DEPMN3)
      EQUIVALENCE (DQL(2,1),DML1),   (DQL(2,2),DML2),   (DQL(2,3),DML3)
      EQUIVALENCE (DQL(3,1),DPMD1),  (DQL(3,2),DPMD2),  (DQL(3,3),DPMD3)
      EQUIVALENCE (DQL(4,1),DAPMN1),(DQL(4,2),DAPMN2),(DQL(4,3),DAPMN3)
C
      DATA ALABEL/10HA.  NU = (,10H0.1,0.1,9.,10H8)         /
      DATA TLABEL/10HTABLE 7.1 /
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C
C     CONVCRI  RELATIVE-ERROR CONVERGENCE CRITERION (USUALLY 0.0001)
C     IP       DENOTES, IN THE FOLLOWING ORDER, ONE OF THE EXPECTED P"S
C              (.01,.01,.98),(.1,.1,.8),(.2,.3,.5), AND (1/3,1/3,1/3)
C     NSS      INTEGER SS
C     NXZ      NUMBER OF TRINOMIAL (X AND Z DATA) SIMULATIONS
C     PAPMI    I-TH T.S. APPROXIMATED POSTERIOR MEAN
C     PEPMI    I-TH EXACT POSTERIOR MEAN
C     PI       I-TH GENERATED P
C     PID      PERCENTAGE OF INCOMPLETE DATA
C     PMLCI    I-TH COMPLETE M.L.E.
C     PMLEI    I-TH INCOMPLETE-DATA M.L.E.
C     PPMDI    I-TH POSTERIOR MODE
C     SS       SAMPLE SIZE
C     SSN      SS + SUM OF PRIOR PARAMETERS XNUI
C     XNU      VECTOR OF PRIOR PARAMETERS XNU=(XNU1,XNU2,XNU3)
C     ZI       NUMBER OF OBSERVATIONS FALLING IN CATEGORY I
C     ZIJ      NUMBER OF OBSERVATIONS SUCH THAT EACH OBSERVATION IS
C              KNOWN TO FALL IN ONE OF CATEGORIES I AND J BUT IS NOT
C              FURTHER CLASSIFIED
C     ZIN      ZI+XNUI
C
C
      XNU1=0.1
      XNU2=0.1
      IP=2
      SEED=24158739.
      GSEED=SEED+100.
C
C     INITIALIZE ONE-DIMENSIONAL FORM OF UNIFORM RANDOM-NUMBER
C     GENERATOR FOR GENERATING GAMMA RANDOM VARIABLES
C
      UN=URAN(GSEED)
      PRINT 4, GSEED,UN
      XNU3=10.-XNU1-XNU2
      XNU=XNU1+XNU2+XNU3
C
C     GENERATE A 3-COMPONENT (2-DIM) VECTOR OF DIRICHLET PROBABILITIES
C
      DO 9910 IGEN=1,10
    2 G1=GAMMA(XNU1)
      G2=GAMMA(XNU2)
      G3=GAMMA(XNU3)
      G=G1+G2+G3
```

```
      P1=G1/G
      P2=G2/G
      P3=1.-P1-P2
      IF (P3-1.) 7,7,5
    5 PRINT 3, XNU1,XNU2,XNU3,XNU,IP,P1,P2,P3,PID,NSS,NXZ,CONVCRI
      PRINT 6
    6 FORMAT(* P3 IS NEGATIVE.  REGENERATE DIRICHLET.*/////)
      GO TO 2
    7 PID=0.15
      NSS=25
      NXZ=200
      CONVCRI=0.0001
      KASE=0
      IPID=1
      ISS=1
      IPRINT=0
    1 SEED=SEED+2.
      PRINT 3, XNU1,XNU2,XNU3,XNU,IP,P1,P2,P3,PID,NSS,NXZ,CONVCRI
    3 FORMAT(1H1,* XNU1=*F6.3* XNU2=*F6.3* XNU3=*F6.3* XNU=*F6.2* IP=*I2
     1* P1=*F6.4* P2=*F6.4* P3=*F6.4* PID=*F4.2* NSS=*I3* NXZ=*I3* CONVC
     2RI=*F7.5//)
C
C     INITIALIZE VECTOR FORM OF UNIFORM RANDOM-NUMBER GENERATOR
C
      CALL URANV(SEED,1,UN)
      PRINT 4, SEED,UN
    4 FORMAT(* SEED GIVEN URANV IS*E23.14*  SEED TRANSFORMED BY URAN FRO
     1M THIS SEED IS*E23.14//)
      SS=NSS
      SSN=SS+XNU
C
      DO 95 NREPLIC=1,2
C
      NUMXZ=NXZR1=NXZR2=NCOV=NXZ
      ISTOP=0
      COVSKIP=0
      AVTPID=AVDPID=0.
C
C     INITIALIZE COUNTERS FOR % RELATIVE DIFFERENCE AND SIGN OF BIAS
C
      DO 28 I=1,9
      DO 27 J=1,7
      PRODQL(I,J)=0.
   27 CONTINUE
      SBIASEP(I)=0.
```

```
   28 CONTINUE
      DO 29 I=1,21
      SBIASQL(I)=0.
   29 CONTINUE
      DO 30 I=1,27
      PRDEP(I)=0.
   30 CONTINUE
      CTRDEP=CTRDQL(1)=CTRDQL(2)=CTRDQL(3)=0.
C
C
C     INITIALIZE COUNTERS FOR NUMBER OF ITERATIONS FOR CONVERGENCE
C
      DO 40 I1=1,6
      AVNUMIT(I1)=0.
      DO 39 I2=1,7
   39 CTNUMIT(I1,I2)=0.
      CTNUMIT(I1,8)=CTNUMIT(I1,9)=CTNUMIT(I1,10)=0.
   40 CONTINUE
      TIM(1)=TIM(2)=TIMEP=TIMAP=0.
C
      DO 42 J=1,2
      BESTEP(1,J)=BESTQL(1,J,1)=BESTQL(1,J,2)=BESTQL(1,J,3)=0.
      BESTEP(2,J)=BESTQL(2,J,1)=BESTQL(2,J,2)=BESTQL(2,J,3)=0.
      BESTEP(3,J)=BESTQL(3,J,1)=BESTQL(3,J,2)=BESTQL(3,J,3)=0.
                  BESTQL(4,J,1)=BESTQL(4,J,2)=BESTQL(4,J,3)=0.
   42 CONTINUE
C
C     INITIALIZE COUNTERS FOR BIAS AND RELATIVE DIFFERENCES
C
      DO 45 K2=1,8
      DO 44 K1=1,8
      COUNTRD(K1,K2)=0.
      IF (K1-3) 43,43,44
   43 COUNTB(K1,K2)=0.
   44 CONTINUE
   45 CONTINUE
C
C     INITIALIZE AVERAGES AND ERROR SUMMARIES TO ZERO
C
C     AVERAGE ESTIMATORS AND THEIR STANDARD ERRORS (S.E.)
C
       APMLE1= APEPM1= APAPM1= APPMD1= APMLC1= AP1=0.
      XAPMLE1=XAPEPM1=XAPAPM1=XAPPMD1=XAPMLC1=XAP1=0.
       APMLE2= APEPM2= APAPM2= APPMD2= APMLC2= AP2=0.
      XAPMLE2=XAPEPM2=XAPAPM2=XAPPMD2=XAPMLC2=XAP2=0.
```

```
      AVPMN1=AVPMN2=AQPMN1=AQPMN2=AQPMD1=AQPMD2=0.
      AFPMN1=AFPMN2=AGPMN1=AGPMN2= PGMD1= PGMD2=0.
C
C     AV BIAS, MSE, & VAR(MSE) OF ESTIMATORS FROM EXACT POSTERIOR MEAN
C
      XAPMN1=XAPMN2=XAPMN4=XAPMNSQ=0.
      XPMD1 =XPMD2 =XPMD4 =XPMDSQ =0.
      XML1  =XML2  =XML4  =XMLSQ  =0.
C
C     AV BIAS, MSE, & VAR(MSE) OF ESTIMATORS FROM GIVEN OR GENERATED P
C
      YEPMN1 =YAPMN1 =YPMD1 =YML1 =YMLC1 =VAPMN1 =QAPMN1 =QPMD1 =0.
      YEPMN2 =YAPMN2 =YPMD2 =YML2 =YMLC2 =VAPMN2 =QAPMN2 =QPMD2 =0.
      YEPMN4 =YAPMN4 =YPMD4 =YML4        =VAPMN4 =QAPMN4 =QPMD4 =0.
      YEPMSQ =YAPMSQ =YPMDSQ=YMLSQ       =VAPMSQ =QAPMSQ =QPMDSQ=0.
      REGAPM0=REGPMD0=REGEPM0=REGML0= REGMLC0=YMLC20=0.
      REGAPM1=REGPMD2=                REGMLC1=YMLC21=0.
      REGAPM2=                        REGMLC2=YMLC22=0.
C
C     INITIALIZE FOR S.E. OF AVERAGE BIAS
C
      WAPMN1=WPMD1=WML1=              WAPMN2=WPMD2=WML2=0.
      UMLC1=UEPMN1=UAPMN1=UPMD1=UML1=FAPMN1=GAPMN1=GPMD1=0.
      UMLC2=UEPMN2=UAPMN2=UPMD2=UML2=FAPMN2=GAPMN2=GPMD2=0.
C
C     EST AVERAGE, BIAS, S.E., AV % REL DIFF, & MSE FOR COV EST FROM EP
C
      AEPMC11=AAPMC11=BAPMC11=C11MSE=VC11MSE=APRDC11=0.
      AEPMC12=AAPMC12=BAPMC12=C12MSE=VC12MSE=APRDC12=0.
      AEPMC22=AAPMC22=BAPMC22=C22MSE=VC22MSE=APRDC22=0.
      A2EPC11=A2EPC12=A2EPC22=VARMSE=0.
      A2APC11=A2APC12=A2APC22=  VAR4=0.
C-----------------------------------------------------------------------
C
C
C     GENERATE NUMXZ SETS OF TRINOMIAL DATA, CALCULATE ESTIMATORS, AND
C     COMPARE THEM AS APPROXIMATIONS FOR THE EXACT POSTERIOR MEAN.
C     ALSO ASSESS HOW WELL THE T.S. APPROXIMATION DOES FOR THE EXACT
C     POSTERIOR COVARIANCE MATRIX.  COMPARE ESTIMATORS WITH GENERATED
C     P AND DO EXPLORATORY ROBUSTNESS STUDY
C
C
C
      DO 65 NT=1,NXZ
      NTS=NT
```

```
      IROBUST=0
C
C     GENERATE MULTINOMIAL COMPLETE DATA X AND INCOMPLETE DATA Z
C     CALCULATE COMPLETE-DATA MAXIMUM LIKELIHOOD ESTIMATE
C
      CALL GENXZ(UU,NSS)
      IF (ISTOP-1) 46,58,58
C
C     CALCULATE EXACT POSTERIOR MEAN
C
   46 CALL EPM
C
C     FOR INCOMPLETE DATA Z, CALCULATE MAXIMUM-LIKELIHOOD ESTIMATE,
C     POSTERIOR MODE, AND TAYLOR-SERIES APPROXIMATED POSTERIOR MEAN
C     AND TAYLOR-SERIES APPROXIMATED POSTERIOR COVARIANCE MATRICES
C
      CALL METHODS
C
      IF (ISTOP-1) 49,58,58
C
C     AVERAGE ESTIMATORS
C
   49 APMLE1=APMLE1+PMLE1
      APMLE2=APMLE2+PMLE2
      APEPM1=APEPM1+PEPM1
      APEPM2=APEPM2+PEPM2
      APAPM1=APAPM1+PAPM1
      APAPM2=APAPM2+PAPM2
      APPMD1=APPMD1+PPMD1
      APPMD2=APPMD2+PPMD2
      APMLC1=APMLC1+PMLC1
      APMLC2=APMLC2+PMLC2
      AP1=AP1+P1
      AP2=AP2+P2
C
C     TO CALCULATE S.E. OF AVERAGE ESTIMATORS (WANT S.E. SMALL RELATIVE
C     TO DIFFERENCE BETWEEN AVERAGE BIAS)
C
      XAPMLE1=XAPMLE1+PMLE1*PMLE1
      XAPMLE2=XAPMLE2+PMLE2*PMLE2
      XAPEPM1=XAPEPM1+PEPM1*PEPM1
      XAPEPM2=XAPEPM2+PEPM2*PEPM2
      XAPAPM1=XAPAPM1+PAPM1*PAPM1
      XAPAPM2=XAPAPM2+PAPM2*PAPM2
      XAPPMD1=XAPPMD1+PPMD1*PPMD1
```

```
      XAPPMD2=XAPPMD2+PPMD2*PPMD2
      XAPMLC1=XAPMLC1+PMLC1*PMLC1
      XAPMLC2=XAPMLC2+PMLC2*PMLC2
      XAP1=XAP1+P1*P1
      XAP2=XAP2+P2*P2
C
C     DIFFERENCES OF ESTIMATORS FROM EXACT POSTERIOR MEAN
C
C     BIAS OF T.S. APPROX POSTERIOR MEAN FROM EXACT POSTERIOR MEAN
C
      XAPMN1=XAPMN1+EAPMN1
      XAPMN2=XAPMN2+EAPMN2
      XXAPMN=EAPMN1*EAPMN1+EAPMN2*EAPMN2+EAPMN3*EAPMN3
C
C     MEAN SQUARE ERROR OF T.S. APM FROM EXACT POSTERIOR MEAN
C
      XAPMNSQ=XAPMNSQ+XXAPMN
C
C     FOR VARIANCE OF MEAN SQUARE ERROR
C
      XAPMN4=XAPMN4+XXAPMN*XXAPMN
C
C     BIAS OF POSTERIOR MODE FROM EXACT POSTERIOR MEAN
C
      XPMD1=XPMD1+EPMD1
      XPMD2=XPMD2+EPMD2
      XXPMD=EPMD1*EPMD1+EPMD2*EPMD2+EPMD3*EPMD3
C
C     MEAN SQUARE ERROR OF POSTERIOR MODE FROM EXACT POSTERIOR MEAN
C
      XPMDSQ=XPMDSQ+XXPMD
C
C     FOR VARIANCE OF MEAN SQUARE ERROR
C
      XPMD4=XPMD4+XXPMD*XXPMD
C
C     BIAS OF MLE FROM EXACT POSTERIOR MEAN (INCOMPLETE DATA)
C
      XML1=XML1+EML1
      XML2=XML2+EML2
      XXML=EML1*EML1+EML2*EML2+EML3*EML3
C
C     MEAN SQUARE ERROR OF IC-D MLE FROM EXACT POSTERIOR MEAN
C
      XMLSQ=XMLSQ+XXML
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C     FOR VARIANCE OF MSE
C
      XML4=XML4+XXML*XXML
C
C     DIFFERENCE OF APPROXIMATED POSTERIOR COVARIANCES FROM EXACT
C     POSTERIOR COVARIANCES
C
      IF (COVSKIP-0) 51,51,50
   50 NCOV=NCOV-1
      COVSKIP=0
      GO TO 53
C
C     CUMULATE VALUES TO AVERAGE EXACT POSTERIOR VARIANCES
C
   51 AEPMC11=AEPMC11+EPMC11
      AEPMC12=AEPMC12+EPMC12
      AEPMC22=AEPMC22+EPMC22
C
C     CUMULATE VALUES TO AVERAGE APPROXIMATE POSTERIOR VARIANCES
C
      AAPMC11=AAPMC11+APMC11
      AAPMC12=AAPMC12+APMC12
      AAPMC22=AAPMC22+APMC22
C
C     FOR S.E. OF COVARIANCE AVERAGES
C
      A2EPC11=A2EPC11+EPMC11*EPMC11
      A2EPC12=A2EPC12+EPMC12*EPMC12
      A2EPC22=A2EPC22+EPMC22*EPMC22
      A2APC11=A2APC11+APMC11*APMC11
      A2APC12=A2APC12+APMC12*APMC12
      A2APC22=A2APC22+APMC22*APMC22
C
C     CUMULATE DIFFERENCES BETWEEN EXACT AND APPROXIMATE POSTERIOR
C     VARIANCES
C
      BAPMC11=BAPMC11+EAPMC11
      BAPMC12=BAPMC12+EAPMC12
      BAPMC22=BAPMC22+EAPMC22
C
C     FOR AVERAGE PERCENT RELATIVE DIFFERENCE
C
      PRDC11=ABS(EAPMC11)/EPMC11
      PRDC12=ABS(EAPMC12/EPMC12)
```

```
      PRDC22=ABS(EAPMC22)/EPMC22
      APRDC11=APRDC11+PRDC11
      APRDC12=APRDC12+PRDC12
      APRDC22=APRDC22+PRDC22
C
C     FOR MSE OF ELEMENTS OF APPROXIMATED POSTERIOR COVARIANCE MATRIX
C
      EAP2C11=EAPMC11*EAPMC11
      EAP2C12=EAPMC12*EAPMC12
      EAP2C22=EAPMC22*EAPMC22
       C11MSE= C11MSE+EAP2C11
       C12MSE= C12MSE+EAP2C12
       C22MSE= C22MSE+EAP2C22
      VVV=EAP2C11+EAP2C12+EAP2C22
      VARMSE=VARMSE+VVV
C
C     FOR VARIANCE OF MSE
C
      VC11MSE=VC11MSE+EAP2C11*EAP2C11
      VC12MSE=VC12MSE+EAP2C12*EAP2C12
      VC22MSE=VC22MSE+EAP2C22*EAP2C22
      VAR4=VAR4+VVV*VVV
C
C     ADD TO BIAS-SIGN AND %-RELATIVE-DIFFERENCE COUNTS
C
      CALL COUNTS(EAPMC11,PRDC11,1)
      CALL COUNTS(EAPMC12,PRDC12,2)
      CALL COUNTS(EAPMC22,PRDC22,3)
C
C     ALSO CHECK DIRECTION OF BIAS OF DEPENDENT COVARIANCES
C
      EC33=EPMC11+EPMC22+2.*EPMC12
      EC13=-EPMC11-EPMC12
      EC23=-EPMC22-EPMC12
      AC33=APMC11+APMC22+2.*APMC12
      AC13=-APMC11-APMC12
      AC23=-APMC22-APMC12
      BC33=AC33-EC33
      BC13=AC13-EC13
      BC23=AC23-EC23
      PRDC13=ABS(BC13/EC13)
      PRDC23=ABS(BC23/EC23)
      PRDC33=ABS(BC33)/EC33
      CALL COUNTS(BC13,PRDC13,6)
      CALL COUNTS(BC23,PRDC23,7)
```

```
      CALL COUNTS(BC33,PROC33,8)
C
C
C     DIFFERENCE OF ESTIMATORS FROM GENERATED P
C
C
   53 YMLC1=YMLC1+DMLC1
      YMLC2=YMLC2+DMLC2
      YYMLCD=DMLC1*DMLC1+DMLC2*DMLC2+DMLC3*DMLC3
      YMLC2O=YMLC2O+YYMLCD
      SQMLCD=YYMLCD*YYMLCD
      REGMLCO=REGMLCO+SQMLCD
C
C     NOTE THAT FOR REMAINING COMPARISONS WITH GENERATED P WE ARE
C     MAINLY INTERESTED IN MSE BECAUSE MAIN CONCERN IS DETERMINING WHICH
C     ESTIMATOR BEST MINIMIZES QUADRATIC LOSS.  HOWEVER, WE WILL ALSO
C     CALCULATE BIAS FROM GENERATED (OR GIVEN) P.
C
C     BIAS OF EXACT POSTERIOR MEAN FROM GENERATED P
C
      YEPMN1=YEPMN1+DEPMN1
      YEPMN2=YEPMN2+DEPMN2
      YYEPMN=DEPMN1*DEPMN1+DEPMN2*DEPMN2+DEPMN3*DEPMN3
C
C     FOR USUAL MEAN SQUARE ERROR OF EPM FROM GENERATED P
C
      YEPMSQ=YEPMSQ+YYEPMN
C
C     FOR REGRESSION ESTIMATION AND CONTROL VARIATE MSE EPM
C
      REGEPMO=REGEPMO+YYEPMN*YYMLCD
C
C     FOR VARIANCE OF ALL 3 MEAN SQUARE ERRORS
C
      YEPMN4=YEPMN4+YYEPMN*YYEPMN
C
C     BIAS OF APPROX POSTERIOR MEAN FROM GENERATED P
C
      YAPMN1=YAPMN1+DAPMN1
      YAPMN2=YAPMN2+DAPMN2
      YYAPMN=DAPMN1*DAPMN1+DAPMN2*DAPMN2+DAPMN3*DAPMN3
C
C     USUAL MEAN SQUARE ERROR FOR APMN FROM GENERATED P
C
      YAPMSQ=YAPMSQ+YYAPMN
```

```
C
C     FOR REGRESSION ESTIMATION AND CONTROL VARIATE MSE
C
      REGAPMO=REGAPMO+YYAPMN*YYMLCD
C
C     FOR VARIANCE OF MEAN SQUARE ERRORS
C
      YAPMN4=YAPMN4+YYAPMN*YYAPMN
C
C     FOR BIAS OF POSTERIOR MODE FROM GENERATED P
C
      YPMD1=YPMD1+DPMD1
      YPMD2=YPMD2+DPMD2
      YYPMD=DPMD1*DPMD1+DPMD2*DPMD2+DPMD3*DPMD3
      YPMDSQ=YPMDSQ+YYPMD
      REGPMDO=REGPMDO+YYPMD*YYMLCD
      YPMD4=YPMD4+YYPMD*YYPMD
C
C     BIAS OF MLE FROM GENERATED P
C
      YML1=YML1+DML1
      YML2=YML2+DML2
      YYML=DML1*DML1+DML2*DML2+DML3*DML3
      YMLSQ=YMLSQ+YYML
      REGMLO=REGMLO+YYML*YYMLCD
      YML4=YML4+YYML*YYML
C
C     BY TWO DIFF CRITERION CALC BEST EST FOR THIS TRINOMIAL TRIAL
C
      CALL BESTEST(BESTQL(1))
C
C     FOR S.E. OF AVERAGE BIAS
C
      WAPMN1=WAPMN1+EAPMN1*EAPMN1
      WAPMN2=WAPMN2+EAPMN2*EAPMN2
      WPMD1=WPMD1+EPMD1*EPMD1
      WPMD2=WPMD2+EPMD2*EPMD2
      WML1=WML1+EML1*EML1
      WML2=WML2+EML2*EML2
      UMLC1=UMLC1+DMLC1*DMLC1
      UMLC2=UMLC2+DMLC2*DMLC2
      UEPMN1=UEPMN1+DEPMN1*DEPMN1
      UEPMN2=UEPMN2+DEPMN2*DEPMN2
      UAPMN1=UAPMN1+DAPMN1*DAPMN1
      UAPMN2=UAPMN2+DAPMN2*DAPMN2
```

```
      UPMD1=UPMD1+DPMD1*DPMD1
      UPMD2=UPMD2+DPMD2*DPMD2
      UML1=UML1+DML1*DML1
      UML2=UML2+DML2*DML2
C
C     CALCULATE ROBUSTNESS ESTIMATORS (FOR QUADRATIC-LOSS COMPARISON
C     ONLY).  USE WRONG PRIOR IN ESTIMATES.
C     EXCLUDE EPM BECAUSE OF EXPENSE.
C
C     ROBUSTNESS SET 1.  UNIFORM PRIOR.  RECALCULATE ONLY APPROXIMATE
C     POSTERIOR MEAN BECAUSE  FOR A UNIFORM PRIOR  THE POSTERIOR MODE
C     WILL EQUAL THE ALREADY CALCULATED M.L.E.  (FOR INCOMPLETE DATA)
C
      Z1N=Z1+1.
      Z2N=Z2+1.
      Z3N=Z2+1.
      SSN=SS+3.
      IROBUST=1
      CALL METHODS
      IF (ISTOP-1) 55,58,58
   55 VAPMN1=VAPMN1+DAPMN1
      VAPMN2=VAPMN2+DAPMN2
      VVAPMN=DAPMN1*DAPMN1+DAPMN2*DAPMN2+DAPMN3*DAPMN3
      AVPMN1=AVPMN1+PAPM1
      AVPMN2=AVPMN2+PAPM2
      FAPMN1=FAPMN1+DAPMN1*DAPMN1
      FAPMN2=FAPMN2+DAPMN2*DAPMN2
      AFPMN1=AFPMN1+PAPM1*PAPM1
      AFPMN2=AFPMN2+PAPM2*PAPM2
      VAPMSQ=VAPMSQ+VVAPMN
      REGAPM1=REGAPM1+VVAPMN*YYMLCD
      VAPMN4=VAPMN4+VVAPMN*VVAPMN
      YMLC21=YMLC21+YYMLCD
      REGMLC1=REGMLC1+SQMLCD
C
C     BY TWO DIFF CRITERION CALC BEST EST FOR THIS TRINOMIAL TRIAL
C
      CALL BESTEST(BESTQL(9))
C
C     ROBUSTNESS SET 2.   IN BAYESIAN ESTIMATORS APMN AND PMD, USE
C     PRIOR PARAMETERS 10.(NU/10.+(0.09,0.05,-0.14))
C
   56 Z1N=Z1+XNU1+0.9
      Z2N=Z2+XNU2+0.5
      Z3N=Z3+XNU3-1.4
```

```
      SSN=SS+XNU
      IROBUST=2
      CALL METHODS
      IF (ISTOP-1) 57,58,58
   57 QAPMN1=QAPMN1+DAPMN1
      QAPMN2=QAPMN2+DAPMN2
      AQPMN1=AQPMN1+PAPM1
      AQPMN2=AQPMN2+PAPM2
      QQAPMN=DAPMN1*DAPMN1+DAPMN2*DAPMN2+DAPMN3*DAPMN3
      QAPMSQ=QAPMSQ+QQAPMN
      REGAPM2=REGAPM2+QQAPMN*YYMLCD
      QAPMN4=QAPMN4+QQAPMN*QQAPMN
      YMLC22=YMLC22+YYMLCD
      REGMLC2=REGMLC2+SQMLCD
C
      QPMD1=QPMD1+DPMD1
      QPMD2=QPMD2+DPMD2
      AQPMD1=AQPMD1+PPMD1
      AQPMD2=AQPMD2+PPMD2
      QQPMD=DPMD1*DPMD1+DPMD2*DPMD2+DPMD3*DPMD3
      QPMDSQ=QPMDSQ+QQPMD
      REGPMD2=REGPMD2+QQPMD*YYMLCD
      QPMD4=QPMD4+QQPMD*QQPMD
C
      GAPMN1=GAPMN1+DAPMN1*DAPMN1
      GAPMN2=GAPMN2+DAPMN2*DAPMN2
      GPMD1=GPMD1+DPMD1*DPMD1
      GPMD2=GPMD2+DPMD2*DPMD2
      AGPMN1=AGPMN1+PAPM1*PAPM1
      AGPMN2=AGPMN2+PAPM2*PAPM2
      PGMD1=PGMD1+PPMD1*PPMD1
      PGMD2=PGMD2+PPMD2*PPMD2
C
C     BY TWO DIFF CRITERION CALC BEST EST FOR THIS TRINOMIAL TRIAL
C
      CALL BESTEST(BESTQL(17))
C
      GO TO 65
   58 ISTOP=0
      IF (IROBUST-1) 60,61,62
   60 NUMXZ=NUMXZ-1
      NCOV=NCOV-1
      NXZR1=NXZR1-1
      NXZR2=NXZR2-1
      GO TO 65
```

```
   61 NXZR1=NXZR1-1
      GO TO 56
   62 NXZR2=NXZR2-1
   65 CONTINUE
C
C     AVERAGE OVER NUMXZ TRINOMIAL RESULTS FOR FIXED XNU=(XNU1,XNU2,XNU3
C     GENERATED P=(P1,P2,P3), PID, AND SS
C
C
C     ESTIMATOR MEANS (ESTIMATORS AVERAGED OVER NUMXZ TRIALS)
C
      APMLE1=APMLE1/NUMXZ
      APMLE2=APMLE2/NUMXZ
      APEPM1=APEPM1/NUMXZ
      APEPM2=APEPM2/NUMXZ
      APAPM1=APAPM1/NUMXZ
      APAPM2=APAPM2/NUMXZ
      APPMD1=APPMD1/NUMXZ
      APPMD2=APPMD2/NUMXZ
      APMLC1=APMLC1/NUMXZ
      APMLC2=APMLC2/NUMXZ
      AP1=AP1/NUMXZ
      AP2=AP2/NUMXZ
      AVPMN1=AVPMN1/NXZR1
      AVPMN2=AVPMN2/NXZR1
      AQPMN1=AQPMN1/NXZR2
      AQPMN2=AQPMN2/NXZR2
      AQPMD1=AQPMD1/NXZR2
      AQPMD2=AQPMD2/NXZR2
C
      ND=NUMXZ*(NUMXZ-1)
      NC=NCOV*(NCOV-1)
      NDR1=NXZR1*(NXZR1-1)
      NDR2=NXZR2*(NXZR2-1)
C
C     STANDARD ERRORS OF ESTIMATOR MEANS
C
      SE1=SQRT((XAPMLE1-NUMXZ*APMLE1*APMLE1)/ND)
      SE2=SQRT((XAPMLE2-NUMXZ*APMLE2*APMLE2)/ND)
      SE3=SQRT((XAPEPM1-NUMXZ*APEPM1*APEPM1)/ND)
      SE4=SQRT((XAPEPM2-NUMXZ*APEPM2*APEPM2)/ND)
      SE5=SQRT((XAPAPM1-NUMXZ*APAPM1*APAPM1)/ND)
      SE6=SQRT((XAPAPM2-NUMXZ*APAPM2*APAPM2)/ND)
      SE7=SQRT((XAPPMD1-NUMXZ*APPMD1*APPMD1)/ND)
      SE8=SQRT((XAPPMD2-NUMXZ*APPMD2*APPMD2)/ND)
```

```
      SE9=SQRT((XAPMLC1-NUMXZ*APMLC1*APMLC1)/ND)
      SE10=SQRT((XAPMLC2-NUMXZ*APMLC2*APMLC2)/ND)
      SE11=SQRT((XAP1-NUMXZ*AP1*AP1)/ND)
      SE12=SQRT((XAP2-NUMXZ*AP2*AP2)/ND)
      SE13=SQRT((AFPMN1-NXZR1*AVPMN1*AVPMN1)/NDR1)
      SE14=SQRT((AFPMN2-NXZR1*AVPMN2*AVPMN2)/NDR1)
      SE15=SQRT((AGPMN1-NXZR2*AQPMN1*AQPMN1)/NDR2)
      SE16=SQRT((AGPMN2-NXZR2*AQPMN2*AQPMN2)/NDR2)
      SE17=SQRT((PGMD1-NXZR2*AQPMD1*AQPMD1)/NDR2)
      SE18=SQRT((PGMD2-NXZR2*AQPMD2*AQPMD2)/NDR2)
      PRINT 1079, NUMXZ
      PRINT 1080, APMLE1,SE1,APEPM1,SE3,APAPM1,SE5,APPMD1,SE7,APMLC1,SE9
     1,AP1,SE11,APMLE2,SE2,APEPM2,SE4,APAPM2,SE6,APPMD2,SE8,APMLC2,SE10,
     2AP2,SE12
 1079 FORMAT(//* AVERAGE ESTIMATORS (MEANS) AND THEIR STANDARD ERRORS OV
     1ER*,I4,* TRIALS*//)
 1080 FORMAT(10X,*APMLE*,7X,*S.E.*,6X,*APEPM*,7X,*S.E.*,6X,*APAPM*,7X,*S
     1.E.*,6X,*APPMD*,7X,*S.E.*,6X,*APMLC*,7X,*S.E.*,9X,*AP*,7X,*S.E.*/*
     2 P1 *12F11.7/* P2 *12F11.7/)
C
C     AVERAGE BIASES AND COVARIANCE ESTIMATORS
C
      XAPMN1=XAPMN1/NUMXZ
      XAPMN2=XAPMN2/NUMXZ
      XPMD1=XPMD1/NUMXZ
      XPMD2=XPMD2/NUMXZ
      XML1=XML1/NUMXZ
      XML2=XML2/NUMXZ
      AEPMC11=AEPMC11/NCOV
      AEPMC12=AEPMC12/NCOV
      AEPMC22=AEPMC22/NCOV
      AAPMC11=AAPMC11/NCOV
      AAPMC12=AAPMC12/NCOV
      AAPMC22=AAPMC22/NCOV
      BAPMC11=BAPMC11/NCOV
      BAPMC12=BAPMC12/NCOV
      BAPMC22=BAPMC22/NCOV
      YMLC1=YMLC1/NUMXZ
      YMLC2=YMLC2/NUMXZ
      YEPMN1=YEPMN1/NUMXZ
      YEPMN2=YEPMN2/NUMXZ
      YAPMN1=YAPMN1/NUMXZ
      YAPMN2=YAPMN2/NUMXZ
      YPMD1=YPMD1/NUMXZ
      YPMD2=YPMD2/NUMXZ
```

```
      YML1=YML1/NUMXZ
      YML2=YML2/NUMXZ
      VAPMN1=VAPMN1/NXZR1
      VAPMN2=VAPMN2/NXZR1
      QAPMN1=QAPMN1/NXZR2
      QAPMN2=QAPMN2/NXZR2
      QPMD1=QPMD1/NXZR2
      QPMD2=QPMD2/NXZR2
      EBIAS1(NREPLIC,ISS,IPID)=XAPMN1
      EBIAS2(NREPLIC,ISS,IPID)=XPMD1
      EBIAS3(NREPLIC,ISS,IPID)=XML1
C
C     CALCULATE S.E. OF BIAS.  WANT THIS TO BE SMALL RELATIVE TO THE
C     DIFFERENCE BETWEEN THE BIASES
C
      SE21=SQRT((WAPMN1-NUMXZ*XAPMN1*XAPMN1)/ND)
      SE22=SQRT((WAPMN2-NUMXZ*XAPMN2*XAPMN2)/ND)
      SE23=SQRT((WPMD1-NUMXZ*XPMD1*XPMD1)/ND)
      SE24=SQRT((WPMD2-NUMXZ*XPMD2*XPMD2)/ND)
      SE25=SQRT((WML1-NUMXZ*XML1*XML1)/ND)
      SE26=SQRT((WML2-NUMXZ*XML2*XML2)/ND)
      SE27=SQRT((A2EPC11-NCOV*AEPMC11*AEPMC11)/NC)
      SE28=SQRT((A2EPC12-NCOV*AEPMC12*AEPMC12)/NC)
      SE29=SQRT((A2EPC22-NCOV*AEPMC22*AEPMC22)/NC)
      SE30=SQRT((A2APC11-NCOV*AAPMC11*AAPMC11)/NC)
      SE31=SQRT((A2APC12-NCOV*AAPMC12*AAPMC12)/NC)
      SE32=SQRT((A2APC22-NCOV*AAPMC22*AAPMC22)/NC)
      SE33=SQRT((C11MSE-NCOV*BAPMC11*BAPMC11)/NC)
      SE34=SQRT((C12MSE-NCOV*BAPMC12*BAPMC12)/NC)
      SE35=SQRT((C22MSE-NCOV*BAPMC22*BAPMC22)/NC)
      SE36=SQRT((UMLC1-NUMXZ*YMLC1*YMLC1)/ND)
      SE37=SQRT((UMLC2-NUMXZ*YMLC2*YMLC2)/ND)
      SE38=SQRT((UEPMN1-NUMXZ*YEPMN1*YEPMN1)/ND)
      SE39=SQRT((UEPMN2-NUMXZ*YEPMN2*YEPMN2)/ND)
      SE40=SQRT((UAPMN1-NUMXZ*YAPMN1*YAPMN1)/ND)
      SE41=SQRT((UAPMN2-NUMXZ*YAPMN2*YAPMN2)/ND)
      SE42=SQRT((UPMD1-NUMXZ*YPMD1*YPMD1)/ND)
      SE43=SQRT((UPMD2-NUMXZ*YPMD2*YPMD2)/ND)
      SE44=SQRT((UML1-NUMXZ*YML1*YML1)/ND)
      SE45=SQRT((UML2-NUMXZ*YML2*YML2)/ND)
      SE46=SQRT((FAPMN1-NXZR1*VAPMN1*VAPMN1)/NDR1)
      SE47=SQRT((FAPMN2-NXZR1*VAPMN2*VAPMN2)/NDR1)
      SE48=SQRT((GAPMN1-NXZR2*QAPMN1*QAPMN1)/NDR2)
      SE49=SQRT((GAPMN2-NXZR2*QAPMN2*QAPMN2)/NDR2)
      SE50=SQRT((GPMD1-NXZR2*QPMD1*QPMD1)/NDR2)
```

```
      SE51=SQRT((GPMD2-NXZR2*QPMD2*QPMD2)/NDR2)
C
      PRINT 1150, NUMXZ,XAPMN1,SE21,XPMD1,SE23,XML1,SE25,XAPMN2,SE22,XPM
     1D2,SE24,XML2,SE26,YEPMN1,SE38,YAPMN1,SE40,YPMD1,SE42,YML1,SE44,YML
     2C1,SE36,YEPMN2,SE39,YAPMN2,SE41,YPMD2,SE43,YML2,SE45,YMLC2,SE37
 1150 FORMAT(//* AVERAGE BIASES AND THEIR STANDARD ERRORS OVER*,I4,* TRI
     1ALS*//* FROM EPM*15X*XAPM*16X*S.E.*16X*XPMD*16X*S.E.*17X*XML*16X*S
     2.E.*/10X,*P1 *,6F20.14/10X,*P2 *,6F20.14//* FROM  P*9X,*YEPM*8X*S.
     3E.*8X*YAPM*8X*S.E.*8X*YPMD*8X*S.E.*9X*YML*8X*S.E.*8X*YMLC*8X*S.E.*
     4/10X*P1 *10F12.10/10X*P2 *10F12.10/)
      PRINT 1154, AVPMN1,SE13,APMLE1,SE1,AQPMN1,SE15,AQPMD1,SE17,AVPMN2,
     1SE14,APMLE2,SE2,AQPMN2,SE16,AQPMD2,SE18
 1154 FORMAT(//* ROBUST-ESTIMATOR AVERAGES AND S.E."S.  WANT S.E. SMALL
     1RELATIVE TO DIFFERENCE BETWEEN ESTIMATORS.*/9X*APMR1*11X*S.E.*9X*P
     2MDR1=MLE*8X*S.E.*11X*APMR2*10X*S.E.*11X*PMDR2*10X*S.E.*/* P1 *4(F1
     32.8,F18.14)/* P2 *4(F12.8,F18.14)//)
      PRINT 1155, VAPMN1,SE46,YML1,SE44,QAPMN1,SE48,QPMD1,SE50,VAPMN2,SE
     147,YML2,SE45,QAPMN2,SE49,QPMD2,SE51
 1155 FORMAT(//* ROBUST-ESTIMATORS BIASES AND STANDARD ERRORS.*/10X,    *
     1VAPM*11X,*S.E.*,9X,*VPMD=YML*9X,*S.E.*11X*QAPM*11X*S.E.*11X*QPMD*1
     21X*S.E.*/* P1 *4(F12.8,F18.14)/* P2 *4(F12.8,F18.14)//)
C
C     MEAN SQUARE ERRORS AND THEIR STANDARD ERRORS
C
      XAPMNSQ=XAPMNSQ/NUMXZ
      XPMDSQ=XPMDSQ/NUMXZ
      XMLSQ=XMLSQ/NUMXZ
      EMS1(NREPLIC,ISS,IPID)=XAPMNSQ
      EMS2(NREPLIC,ISS,IPID)=XPMDSQ
      EMS3(NREPLIC,ISS,IPID)=XMLSQ
      YMLC20=YMLC20/NUMXZ
      YEPMSQ=YEPMSQ/NUMXZ
      YAPMSQ=YAPMSQ/NUMXZ
      YPMDSQ=YPMDSQ/NUMXZ
      YMLSQ=YMLSQ/NUMXZ
      SD1=SQRT((XAPMN4-NUMXZ*XAPMNSQ*XAPMNSQ)/ND)
      SD2=SQRT((XPMD4-NUMXZ*XPMDSQ*XPMDSQ)/ND)
      SD3=SQRT((XML4-NUMXZ*XMLSQ*XMLSQ)/ND)
      SD4=SQRT((REGMLCO-NUMXZ*YMLC20*YMLC20)/ND)
      SD5=SQRT((YEPMN4-NUMXZ*YEPMSQ*YEPMSQ)/ND)
      SD6=SQRT((YAPMN4-NUMXZ*YAPMSQ*YAPMSQ)/ND)
      SD7=SQRT((YPMD4-NUMXZ*YPMDSQ*YPMDSQ)/ND)
      SD8=SQRT((YML4-NUMXZ*YMLSQ*YMLSQ)/ND)
C
C     DIFFERENCES BETWEEN MEAN SQUARE ERRORS
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
      DM1=XAPMNSQ-XPMDSQ
      DM2=XAPMNSQ-XMLSQ
      DM3=XPMDSQ-XMLSQ
      DM4=YMLC20-YEPMSQ
      DM5=YMLC20-YAPMSQ
      DM6=YMLC20-YPMDSQ
      DM7=YMLC20-YMLSQ
      DM8=YEPMSQ-YAPMSQ
      DM9=YEPMSQ-YPMDSQ
      DM10=YEPMSQ-YMLSQ
      DM11=YAPMSQ-YPMDSQ
      DM12=YAPMSQ-YMLSQ
      DM13=YPMDSQ-YMLSQ
      PRINT 1200, NUMXZ,XAPMNSQ,SD1,YEPMSQ,SD5,YMLC20,SD4,XPMDSQ,SD2,YAP
     1MSQ,SD6,YMLSQ,SD8,XMLSQ,SD3,YPMDSQ,SD7
 1200 FORMAT(//* AVERAGE MEAN SQUARE ERRORS AND THEIR STANDARD ERRORS OV
     1ER*I4* TRIALS*//17X,*MSE*12X*S.E.*25X*MSE*12X*S.E.*25X*MSE*12X*S.E
     2.*/2X*APM-EPM*F14.9,F18.13,7X,*EPM-P*F14.9,F18.13,7X,*MLC-P*F14.9,
     3F18.13/2X*PMD-EPM*F14.9,F18.13,7X,*APM-P*F14.9,F18.13,8X,*ML-P*F14
     4.9,F18.13/3X,*ML-EPM*F14.9,F18.13,7X,*PMD-P*F14.9,F18.13/)
      PRINT 1300, DM1,DM8,DM11,DM5,DM2,DM9,DM12,DM6,DM3,DM10,DM13,DM7
 1300 FORMAT(//* DIFFERENCE BETWEEN AVERAGE MSE"S.  WANT DIFFERENCE LARG
     1E RELATIVE TO SE(MSE)*//*  EAPM-EPMD*F14.9,12X*DEPM-DAPM*F14.9,12X
     2*DAPM-DPMD*F14.9,12X*DMLC-DAPM*F14.9/*  EAPM-EMLE*F14.9,12X*DEPM-D
     3PMD*F14.9,12X*DAPM-DMLE*F14.9,12X*DMLC-DPMD*F14.9/*  EPMD-EMLE*F14
     4.9,12X*DEPM-DMLE*F14.9,12X*DPMD-DMLE*F14.9,12X*DMLC-DMLE*F14.9/)
      XN=NUMXZ
      DO 71 I=1,6
      IF (I-4) 69,67,68
   67 XN=NXZR1
      GO TO 69
   68 XN=NXZR2
   69 AVNUMIT(I)=AVNUMIT(I)/XN
      DO 70 J=1,10
   70 CTNUMIT(I,J)=CTNUMIT(I,J)/XN
   71 CONTINUE
      PRINT 2000, IP,PID,SS,NREPLIC,NUMXZ,NXZR1,NXZR2,(AVNUMIT(I),I=1,6)
 2000 FORMAT(///* NUMBER OF ITER FOR CONVERGENCE AVERAGED OVER NUMBER OF
     1 TRINOMIAL SIMULATIONS.  IP=*I3,* PID=*F4.2,* SS=*F3.0,* NREPLIC=*
     2I2/* NUMXZ=*I3,* NXZR1=*I3,* NXZR2=*I3,* AV NUM ITER FOR   MLE=*F7
     3.3* PMDR0=*F7.3* APMR0=*F7.3* APMR1=*F7.3* PMDR2=*F7.3* APMR2=*F7.
     43//)
      PRINT 2010, ((CTNUMIT(I,J),J=1,10),I=1,4,3),((CTNUMIT(I,J),J=1,10)
     1,I=2,5,3),((CTNUMIT(I,J),J=1,10),I=3,6,3)
```

```
 2010 FORMAT(* PROPORTION OF DATA SETS FOR WHICH NUMBER OF ITERATIONS WA
     1S OF SPECIFIED AMOUNTS*/10X*1      2      3      4      5      6      7
     2  8-10 11-15 GT 15*11X*1      2      3      4      5      6      7      8
     3-10 11-15 GT 15*/*   MLE *10F6.3*  APMR1 *10F6.3/* PMDR0 *10F6.3*
     4 PMDR2 *10F6.3/* APMR0 *10F6.3*  APMR2 *10F6.3//)
      YEPMSQ=NUMXZ*YEPMSQ
      YAPMSQ=NUMXZ*YAPMSQ
      YPMDSQ=NUMXZ*YPMDSQ
      YMLSQ=NUMXZ*YMLSQ
      YMLC20=NUMXZ*YMLC20
      T=(1.-(P1*P1+P2*P2+P3*P3))/SS
C
      CALL ESTMSE(YEPMSQ,YEPMN4,REGEPMO,YMLC20,REGMLCO,T,NUMXZ,MSE(1))
      CALL ESTMSE(YAPMSQ,YAPMN4,REGAPMO,YMLC20,REGMLCO,T,NUMXZ,MSE(7))
      CALL ESTMSE(YPMDSQ,YPMD4,REGPMDO,YMLC20,REGMLCO,T,NUMXZ,MSE(13))
      CALL ESTMSE(YMLSQ,YML4,REGMLO,YMLC20,REGMLCO,T,NUMXZ,MSE(19))
      CALL ESTMSE(VAPMSQ,VAPMN4,REGAPM1,YMLC21,REGMLC1,T,NXZR1,MSE(25))
      CALL ESTMSE(OAPMSQ,OAPMN4,REGAPM2,YMLC21,REGMLC2,T,NXZR2,MSE(31))
      CALL ESTMSE(QPMDSQ,QPMD4,REGPMD2,YMLC22,REGMLC2,T,NXZR2,MSE(37))
C
      PRINT 2030, IP,PID,SS,NREPLIC,((MSE(I,J),J=1,7),I=1,6)
 2030 FORMAT(* THREE KINDS OF MSE (AND THEIR VARIANCES) FOR QUADRATIC-LO
     1SS COMPARISONS.  IP=*I2,* PID=*F4.2,* SS=*F3.0,* NREPLIC=*I2//18X*
     2EPM*15X*APM*15X*PMD*15X*MLE*14X*APMR1*13X*APMR2*13X*PMDR2*/* REG M
     3SE *7E18.7/* VAR(MSE)*7E18.7/*  CV MSE *7E18.7/* VAR(MSE)*7E18.7/*
     4  RE MSE *7E18.7/* VAR(MSE)*7E18.7//)
C
      IF (ISS-1) 2035,2035,2040
 2035 QLMS11(IGEN,IPID,NREPLIC)=MSE(5,2)
      QLMS21(IGEN,IPID,NREPLIC)=MSE(5,3)
      QLMS31(IGEN,IPID,NREPLIC)=MSE(5,4)
      QLMS41(IGEN,IPID,NREPLIC)=MSE(5,5)
      QLMS51(IGEN,IPID,NREPLIC)=MSE(5,6)
      QLMS61(IGEN,IPID,NREPLIC)=MSE(5,7)
      GO TO 2045
 2040 QLMS12(IGEN,IPID,NREPLIC)=MSE(5,2)
      QLMS22(IGEN,IPID,NREPLIC)=MSE(5,3)
      QLMS32(IGEN,IPID,NREPLIC)=MSE(5,4)
      QLMS42(IGEN,IPID,NREPLIC)=MSE(5,5)
      QLMS52(IGEN,IPID,NREPLIC)=MSE(5,6)
      QLMS62(IGEN,IPID,NREPLIC)=MSE(5,7)
C
C     PROPORTIONS FOR BEST ESTIMATOR (BEST IN TERMS OF SMALLEST SUMMED
C     SQUARED ERROR AND % REL DIFF FOR SUM BEING OVER THE THREE
C     COMPONENTS OF AN ESTIMATOR) AND FOR SIGN OF BIAS
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
 2045 CTSEQL(1)=NUMXZ
      CTSEQL(2)=NXZR1
      CTSEQL(3)=NXZR2
      DO 75 I=1,4
      DO 72 IR=1,3
      BESTQL(I,1,IR)=BESTQL(I,1,IR)/CTSEQL(IR)
      BESTQL(I,2,IR)=BESTQL(I,2,IR)/CTRDQL(IR)
      SBIASQL(IR,I)=SBIASQL(IR,I)/NUMXZ
   72 CONTINUE
      IF (I-4) 73,75,75
   73 BESTEP(I,1)=BESTEP(I,1)/NUMXZ
      BESTEP(I,2)=BESTEP(I,2)/CTRDEP
      DO 74 K=1,3
      SBIASEP(K,I)=SBIASEP(K,I)/NUMXZ
   74 CONTINUE
   75 CONTINUE
      DO 76 K=1,3
      SBIASQL(K,5)=SBIASQL(K,5)/NXZR1
      SBIASQL(K,6)=SBIASQL(K,6)/NXZR2
      SBIASQL(K,7)=SBIASQL(K,7)/NXZR2
   76 CONTINUE
C
C     CALCULATE % ABS REL DIFF LESS THAN (INSTEAD OF BETWEEN) SPECIFIED
C     AMOUNTS
C
      DO 80 I=1,7
      DO 79 II=2,8
      PRDQL(II,I)=PRDQL(II,I)+PRDQL(II-1,I)
      IF (I-4) 78,79,79
   78 PRDEP(II,I)=PRDEP(II,I)+PRDEP(II-1,I)
   79 CONTINUE
   80 CONTINUE
      IR=1
      DO 85 I=1,7
      IF (I-5) 83,81,82
   81 IR=2
      GO TO 83
   82 IR=3
   83 DO 84 II=1,8
      PRDQL(II,I)=PRDQL(II,I)/CTRDQL(IR)
   84 CONTINUE
   85 CONTINUE
      DO 88 I=1,3
      DO 88 II=1,8
```

```
      PRDEP(II,I)=PRDEP(II,I)/CTRDEP
   88 CONTINUE
      PRINT 2050, ((BESTEP(I,J),I=1,3), ((BESTQL(I,J,K),I=2,4),K=1,3),J=
     11,2)
 2050 FORMAT(* PROPORTION OF CASES THAT AN ESTIMATOR IS BEST.  FIRST 3 C
     1OLNS ARE RESULTS FOR ESTIMATING EPM.  REMAINING COLNS, FOR MIN QUA
     2D LOSS.*/43X*ORIG.-PRIOR ROBUST SET     UNIFORM-PRIOR ROBUST SET
     3   PERTURB-PRIOR ROBUST SET*/20X*MLE   PMD   APM*11X,    *MLE   PM
     4D   APM*14X*MLE = PMD   APM*13X*MLE   PMD   APM*/*  SQD ERR  CRIT
     5 *3F6.2,8X3F6.2,3X,2(9X,3F6.2)/*  REL DIFF CRIT  *3F6.2,8X,3F6.2,3
     6X,2(9X,3F6.2)//)
      PRINT 2070, ((SBIASEP(I,J),J=1,3),(SBIASQL(I,K),K=1,7),I=1,3)
 2070 FORMAT(* PROPORTION OF CASES IN WHICH DIFFERENCE BETWEEN FIRST COM
     1PONENT OF ESTIMATOR AND THAT OF ESTIMATED IS OF A CERTAIN SIGN*//1
     25X*EMLE*6X*EPMD*6X*EAPM*16X*QEPM*6X*QMLE*5X*QPMDRO*4X*QAPMRO*9X*QA
     3PMR1*9X*QPMDR2*4X*QAPMR2*/6X*NEG *3F10.4,10X,4F10.4,5X,F10.4,5X,2F
     410.4/5X*ZERO *3F10.4,10X,4F10.4,5X,F10.4,5X,2F10.4/6X*POS *3F10.4,
     510X,4F10.4,5X,F10.4,5X,2F10.4//)
      PRINT 2080, ((PRDEP(II,I),I=1,3),(PRDQL(II,K),K=1,7),II=1,8)
 2080 FORMAT(* PROPORTION OF CASES FOR WHICH % ABSOLUTE RELATIVE DIFFERE
     1NCE FOR EACH OF THE THREE ESTIMATOR COMPONENTS IS LESS THAN SPECIF
     2IED AMOUNTS*/15X*EMLE*6X*EPMD*6X*EAPM*16X*QEPM*6X*QMLE*5X*QPMDRO*4
     3X*QAPMRO*9X*QAPMR1*9X*QPMDR2*4X*QAPMR2*/6X*0.01*3F10.4,10X,4F10.4,
     45X,F10.4,5X,2F10.4/6X*0.1 *3F10.4,10X,4F10.4,5X,F10.4,5X,2F10.4/6X
     5*1.0 *3F10.4,10X,4F10.4,5X,F10.4,5X,2F10.4/5X,* 5.0 *3F10.4,10X,4F
     610.4,5X,F10.4,5X,2F10.4/5X,*10.0 *3F10.4,10X,4F10.4,5X,F10.4,5X,2F
     710.4/5X,*15.0 *3F10.4,10X,4F10.4,5X,F10.4,5X,2F10.4/5X*20.0 *3F10.
     84,10X,4F10.4,5X,F10.4,5X,2F10.4/5X,*25.0 *3F10.4,10X,4F10.4,5X,F10
     9.4,5X,2F10.4//)
C
C     PERCENT AVERAGE RELATIVE DIFFERENCE FOR COVARIANCE ESTIMATES
C
      PARDC11=100.*(AEPMC11-AAPMC11)/AEPMC11
      PARDC12=100.*(AEPMC12-AAPMC12)/AEPMC12
      PARDC22=100.*(AEPMC22-AAPMC22)/AEPMC22
C
C     AVERAGE PERCENT RELATIVE DIFFERENCE
C
      APRDC11=100.*APRDC11/NCOV
      APRDC12=100.*APRDC12/NCOV
      APRDC22=100.*APRDC22/NCOV
C
C     SQUARE ROOT MSE DIVIDED BY AVERAGE EPV
C
      C11MSE=C11MSE/NCOV
```

```
      C12MSE=C12MSE/NCOV
      C22MSE=C22MSE/NCOV
      C11RT1=SQRT(C11MSE)/AEPMC11
      C12RT1=SQRT(C12MSE)/AEPMC12
      C22RT1=SQRT(C22MSE)/AEPMC22
C
C     SE(MSE)/MSE
C
      SE11MSE=SQRT((VC11MSE- NCOV*C11MSE*C11MSE)/NC)
      SE12MSE=SQRT((VC12MSE- NCOV*C12MSE*C12MSE)/NC)
      SE22MSE=SQRT((VC22MSE- NCOV*C22MSE*C22MSE)/NC)
      C11RT2=SE11MSE/C11MSE
      C12RT2=SE12MSE/C12MSE
      C22RT2=SE22MSE/C22MSE
      DO 91 J=1,8
      DO 90 I=1,8
      COUNTRD(I,J)=COUNTRD(I,J)/NCOV
      IF (I-3) 89,89,90
   89 COUNTB(I,J)=COUNTB(I,J)/NCOV
   90 CONTINUE
   91 CONTINUE
C
      PRINT 3000, AEPMC11,SE27,AAPMC11,SE30,PARDC11,APRDC11,C11RT1,C11RT
     12,AEPMC12,SE28,AAPMC12,SE31,PARDC12,APRDC12,C12RT1,C12RT2,AEPMC22,
     2SE29,AAPMC22,SE32,PARDC22,APRDC22,C22RT1,C22RT2
 3000 FORMAT(////12X,*AVERAGE EPV*8X*S.E.*9X*AVERAGE APV*8X*S.E.*7X*% AV
     1 REL DIFF   AV % REL DIFF   SQRT(MSE)/(1)    SE(MSE)/MSE*//3X*C11
     2*8E16.7/3X*C12 *8E16.7/3X*C22 *8E16.7)
      PRINT 3005,  NCOV,IP,PID,SS,NREPLIC,(COUNTRD(I,1),I=1,6),(COUNTRD(
     1I,3),I=1,6),(COUNTRD(I,8),I=1,6),(COUNTRD(I,2),I=1,6),(COUNTRD(I,6
     2),I=1,6),(COUNTRD(I,7),I=1,6)
 3005 FORMAT(///* PROPORTION OF*I4* CASES IN WHICH PERCENT REL DIFF WAS
     1LESS THAN VARYING AMOUNTS.  IP=*I2*  PID=*F4.2*  SS=*F4.0*  NREPLI
     2C=*I2//8X*.01    0.1    1.0    5.0    10.    15.*12X*.01    0.1    1.0
     35.0    10.    15.*12X*.01    0.1    1.0    5.0    10.    15.*/* C11*6F6.
     43,6X*C22*6F6.3,6X*C33*6F6.3/*  C12*6F6.3,6X*C13*6F6.3,6X*C23*6F6.3
     5//)
      PRINT 3010, NCOV,(COUNTB(I,1),I=1,3),(COUNTB(I,3),I=1,3),(COUNTB(I
     1,8),I=1,3),(COUNTB(I,2),I=1,3),(COUNTB(I,6),I=1,3),(COUNTB(I,7),I=
     21,3)
 3010 FORMAT(* PROPORTN OF*I4* CASES IN WHICH BIAS IS OF A CERTAIN SIGN*
     1//7X*NEG   ZERO    POS*7X*NEG   ZERO    POS*7X*NEG   ZERO    POS*7X*NEG
     2 ZERO    POS*7X*NEG   ZERO    POS*7X*NEG   ZERO    POS*/* C11*3F6.3* C2
     32*3F6.3* C33*3F6.3* C12*3F6.3* C13*3F6.3* C23*3F6.3)
C
```

```
C     AVERAGE TRUE PERCENT (PROPORTION) INCOMPLETE DATA
C
      AVTPID=AVTPID+TPID
      AVDPID=AVDPID+DPID
      PRINT 3070, PID,NUMXZ,AVTPID,AVDPID
 3070 FORMAT(* GIVEN PID IS*F4.2* AVERAGE OVER NUMXZ=*I3* TRIALS OF TRUE
     1 PID IS*F6.2* AV DIFF BETWEEN TRUE AND GIVEN PID OVER THESE TRIALS
     2 IS*F6.2/////)
C
   95 CONTINUE
C
      KASE=KASE+1
      GO TO (9902,9903,9904,9905) KASE
 9902 PID=0.40
      IPID=2
      IPRINT=1
      GO TO 1
 9903 PID=0.15
      NSS=50
      IPID=1
      ISS=2
      GO TO 1
 9904 PID=0.40
      IPID=2
      ISS=2
      GO TO 1
 9905 CONTINUE
C
      WRITE(12,8000) TLABEL(1),(ALABEL(I),I=1,3)
 8000 FORMAT(63X,A10///7X*REGRESSION-ESTIMATE MSE DATA OVER 200 TRINOMIA
     1L SIMULATIONS.  TWO REPLICATIONS PER CELL.  DESIGN 2.  *2A10,A2//)
      WRITE(12,8001)
 8001 FORMAT(47X*A.  ORIGINAL PRIOR IN BAYESIAN ESTIMATORS.*////)
      WRITE(12,8010)
 8010 FORMAT(12X,27H************************** ,*SS=25*,27H ************
     1**************,4X,28H ************************** ,*SS=50*,27H ****
     2**********************/* ESTI- DIR. *,9H*********,* PID=15 *,10H**
     3********,5X,9H*********,* PID=40 *,10H**********,5X,9H*********,*
     4PID=15 *,10H**********,5X,9H*********,* PID=40 *,10H**********/* M
     5ATOR GEN. REPLICATION1   REPLICATION2     REPLICATION1   REPLICATI
     6ON2     REPLICATION1   REPLICATION2     REPLICATION1   REPLICATION
     72*/)
      WRITE(12,8011)  (I,((QLMS11(I,J,K),K=1,2),J=1,2),((QLMS12(I,J,K),K
     1=1,2),J=1,2),I=1,10)
 8011 FORMAT(* APMRO *I2,4(2E15.6,2X)/(9X,4(2E15.6,2X)))
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      WRITE(12,8012)  (I,((QLMS21(I,J,K),K=1,2),J=1,2),((QLMS22(I,J,K),K
     1=1,2),J=1,2),I=1,10)
 8012 FORMAT(* PMDRO *I2,4(2E15.6,2X)/(9X,4(2E15.6,2X)))
      WRITE(12,8013)  (I,((QLMS31(I,J,K),K=1,2),J=1,2),((QLMS32(I,J,K),K
     1=1,2),J=1,2),I=1,10)
 8013 FORMAT(* MLE   *I2,4(2E15.6,2X)/(9X,4(2E15.6,2X)))
      WRITE(12,4997)
      WRITE(12,8000) TLABEL(1),(ALABEL(I),I=1,3)
      WRITE(12,8020)
 8020 FORMAT(40X*B.  UNIFORM AND PERTURBED PRIOR IN BAYESIAN ESTIMATORS.
     1*////)
      WRITE(12,8010)
      WRITE(12,8021)  (I,((QLMS41(I,J,K),K=1,2),J=1,2),((QLMS42(I,J,K),K
     1=1,2),J=1,2),I=1,10)
 8021 FORMAT(* APMR1 *I2,4(2E15.6,2X)/(9X,4(2E15.6,2X)))
      WRITE(12,8022)  (I,((QLMS51(I,J,K),K=1,2),J=1,2),((QLMS52(I,J,K),K
     1=1,2),J=1,2),I=1,10)
 8022 FORMAT(* APMR2 *I2,4(2E15.6,2X)/(9X,4(2E15.6,2X)))
      WRITE(12,8023)  (I,((QLMS61(I,J,K),K=1,2),J=1,2),((QLMS62(I,J,K),K
     1=1,2),J=1,2),I=1,10)
 8023 FORMAT(* PMDR2 *I2,4(2E15.6,2X)/(9X,4(2E15.6,2X)))
C
 9910 CONTINUE
C
C     CALCULATE TUKEY DATA SUMMARIES, MEAN, AND STANDARD ERROR (S.E.)
C
      DO 9925 IPID=1,2
      DO 9925 NREPLIC=1,2
      SUM=0.
      SQSM=0.
      DO 9911 I=1,10
      SUM = SUM+QLMS11(I,IPID,NREPLIC)
      SQSM=SQSM+QLMS11(I,IPID,NREPLIC)*QLMS11(I,IPID,NREPLIC)
 9911 TUKEY(I)=QLMS11(I,IPID,NREPLIC)
      CALL SUMMARY(TUKEY,10,T11(IPID,NREPLIC,1),T11(IPID,NREPLIC,2),T11(
     1IPID,NREPLIC,3),T11(IPID,NREPLIC,4),T11(IPID,NREPLIC,5))
      T11(IPID,NREPLIC,6)=SUM/10.
      T11(IPID,NREPLIC,7)=SQRT((SQSM-10.*T11(IPID,NREPLIC,6)*T11(IPID,NR
     1EPLIC,6))/90.)
      T11(IPID,NREPLIC,8)=T11(IPID,NREPLIC,3)
      T11(IPID,NREPLIC,9)=(T11(IPID,NREPLIC,2)+2.*T11(IPID,NREPLIC,3)+T1
     11(IPID,NREPLIC,4))/4.
      T11(IPID,NREPLIC,10)=T11(IPID,NREPLIC,4)-T11(IPID,NREPLIC,2)
      T11(IPID,NREPLIC,11)=T11(IPID,NREPLIC,5)-T11(IPID,NREPLIC,1)
      SUM=0.
```

```
      SQSM=0.
      DO 9912 I=1,10
      SUM = SUM+QLMS12(I,IPID,NREPLIC)
      SQSM=SQSM+QLMS12(I,IPID,NREPLIC)*QLMS12(I,IPID,NREPLIC)
 9912 TUKEY(I)=QLMS12(I,IPID,NREPLIC)
      CALL SUMMARY(TUKEY,10,T12(IPID,NREPLIC,1),T12(IPID,NREPLIC,2),T12(
     1IPID,NREPLIC,3),T12(IPID,NREPLIC,4),T12(IPID,NREPLIC,5))
      T12(IPID,NREPLIC,6)=SUM/10.
      T12(IPID,NREPLIC,7)=SQRT((SQSM-10.*T12(IPID,NREPLIC,6)*T12(IPID,NR
     1EPLIC,6))/90.)
      T12(IPID,NREPLIC,8)=T12(IPID,NREPLIC,3)
      T12(IPID,NREPLIC,9)=(T12(IPID,NREPLIC,2)+2.*T12(IPID,NREPLIC,3)+T1
     12(IPID,NREPLIC,4))/4.
      T12(IPID,NREPLIC,10)=T12(IPID,NREPLIC,4)-T12(IPID,NREPLIC,2)
      T12(IPID,NREPLIC,11)=T12(IPID,NREPLIC,5)-T12(IPID,NREPLIC,1)
      SUM=0.
      SQSM=0.
      DO 9913 I=1,10
      SUM = SUM+QLMS21(I,IPID,NREPLIC)
      SQSM=SQSM+QLMS21(I,IPID,NREPLIC)*QLMS21(I,IPID,NREPLIC)
 9913 TUKEY(I)=QLMS21(I,IPID,NREPLIC)
      CALL SUMMARY(TUKEY,10,T21(IPID,NREPLIC,1),T21(IPID,NREPLIC,2),T21(
     1IPID,NREPLIC,3),T21(IPID,NREPLIC,4),T21(IPID,NREPLIC,5))
      T21(IPID,NREPLIC,6)=SUM/10.
      T21(IPID,NREPLIC,7)=SQRT((SQSM-10.*T21(IPID,NREPLIC,6)*T21(IPID,NR
     1EPLIC,6))/90.)
      T21(IPID,NREPLIC,8)=T21(IPID,NREPLIC,3)
      T21(IPID,NREPLIC,9)=(T21(IPID,NREPLIC,2)+2.*T21(IPID,NREPLIC,3)+T2
     11(IPID,NREPLIC,4))/4.
      T21(IPID,NREPLIC,10)=T21(IPID,NREPLIC,4)-T21(IPID,NREPLIC,2)
      T21(IPID,NREPLIC,11)=T21(IPID,NREPLIC,5)-T21(IPID,NREPLIC,1)
      SUM=0.
      SQSM=0.
      DO 9914 I=1,10
      SUM = SUM+QLMS22(I,IPID,NREPLIC)
      SQSM=SQSM+QLMS22(I,IPID,NREPLIC)*QLMS22(I,IPID,NREPLIC)
 9914 TUKEY(I)=QLMS22(I,IPID,NREPLIC)
      CALL SUMMARY(TUKEY,10,T22(IPID,NREPLIC,1),T22(IPID,NREPLIC,2),T22(
     1IPID,NREPLIC,3),T22(IPID,NREPLIC,4),T22(IPID,NREPLIC,5))
      T22(IPID,NREPLIC,6)=SUM/10.
      T22(IPID,NREPLIC,7)=SQRT((SQSM-10.*T22(IPID,NREPLIC,6)*T22(IPID,NR
     1EPLIC,6))/90.)
      T22(IPID,NREPLIC,8)=T22(IPID,NREPLIC,3)
      T22(IPID,NREPLIC,9)=(T22(IPID,NREPLIC,2)+2.*T22(IPID,NREPLIC,3)+T2
     12(IPID,NREPLIC,4))/4.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      T22(IPID,NREPLIC,10)=T22(IPID,NREPLIC,4)-T22(IPID,NREPLIC,2)
      T22(IPID,NREPLIC,11)=T22(IPID,NREPLIC,5)-T22(IPID,NREPLIC,1)
      SUM=0.
      SQSM=0.
      DO 9915 I=1,10
      SUM = SUM+QLMS31(I,IPID,NREPLIC)
      SQSM=SQSM+QLMS31(I,IPID,NREPLIC)*QLMS31(I,IPID,NREPLIC)
 9915 TUKEY(I)=QLMS31(I,IPID,NREPLIC)
      CALL SUMMARY(TUKEY,10,T31(IPID,NREPLIC,1),T31(IPID,NREPLIC,2),T31(
     1IPID,NREPLIC,3),T31(IPID,NREPLIC,4),T31(IPID,NREPLIC,5))
      T31(IPID,NREPLIC,6)=SUM/10.
      T31(IPID,NREPLIC,7)=SQRT((SQSM-10.*T31(IPID,NREPLIC,6)*T31(IPID,NR
     1EPLIC,6))/90.)
      T31(IPID,NREPLIC,8)=T31(IPID,NREPLIC,3)
      T31(IPID,NREPLIC,9)=(T31(IPID,NREPLIC,2)+2.*T31(IPID,NREPLIC,3)+T3
     11(IPID,NREPLIC,4))/4.
      T31(IPID,NREPLIC,10)=T31(IPID,NREPLIC,4)-T31(IPID,NREPLIC,2)
      T31(IPID,NREPLIC,11)=T31(IPID,NREPLIC,5)-T31(IPID,NREPLIC,1)
      SUM=0.
      SQSM=0.
      DO 9916 I=1,10
      SUM = SUM+QLMS32(I,IPID,NREPLIC)
      SQSM=SQSM+QLMS32(I,IPID,NREPLIC)*QLMS32(I,IPID,NREPLIC)
 9916 TUKEY(I)=QLMS32(I,IPID,NREPLIC)
      CALL SUMMARY(TUKEY,10,T32(IPID,NREPLIC,1),T32(IPID,NREPLIC,2),T32(
     1IPID,NREPLIC,3),T32(IPID,NREPLIC,4),T32(IPID,NREPLIC,5))
      T32(IPID,NREPLIC,6)=SUM/10.
      T32(IPID,NREPLIC,7)=SQRT((SQSM-10.*T32(IPID,NREPLIC,6)*T32(IPID,NR
     1EPLIC,6))/90.)
      T32(IPID,NREPLIC,8)=T32(IPID,NREPLIC,3)
      T32(IPID,NREPLIC,9)=(T32(IPID,NREPLIC,2)+2.*T32(IPID,NREPLIC,3)+T3
     12(IPID,NREPLIC,4))/4.
      T32(IPID,NREPLIC,10)=T32(IPID,NREPLIC,4)-T32(IPID,NREPLIC,2)
      T32(IPID,NREPLIC,11)=T32(IPID,NREPLIC,5)-T32(IPID,NREPLIC,1)
      SUM=0.
      SQSM=0.
      DO 9917 I=1,10
      SUM = SUM+QLMS41(I,IPID,NREPLIC)
      SQSM=SQSM+QLMS41(I,IPID,NREPLIC)*QLMS41(I,IPID,NREPLIC)
 9917 TUKEY(I)=QLMS41(I,IPID,NREPLIC)
      CALL SUMMARY(TUKEY,10,T41(IPID,NREPLIC,1),T41(IPID,NREPLIC,2),T41(
     1IPID,NREPLIC,3),T41(IPID,NREPLIC,4),T41(IPID,NREPLIC,5))
      T41(IPID,NREPLIC,6)=SUM/10.
      T41(IPID,NREPLIC,7)=SQRT((SQSM-10.*T41(IPID,NREPLIC,6)*T41(IPID,NR
     1EPLIC,6))/90.)
```

```
      T41(IPID,NREPLIC,8)=T41(IPID,NREPLIC,3)
      T41(IPID,NREPLIC,9)=(T41(IPID,NREPLIC,2)+2.*T41(IPID,NREPLIC,3)+T4
     11(IPID,NREPLIC,4))/4.
      T41(IPID,NREPLIC,10)=T41(IPID,NREPLIC,4)-T41(IPID,NREPLIC,2)
      T41(IPID,NREPLIC,11)=T41(IPID,NREPLIC,5)-T41(IPID,NREPLIC,1)
      SUM=0.
      SQSM=0.
      DO 9918 I=1,10
      SUM = SUM+QLMS42(I,IPID,NREPLIC)
      SQSM=SQSM+QLMS42(I,IPID,NREPLIC)*QLMS42(I,IPID,NREPLIC)
 9918 TUKEY(I)=QLMS42(I,IPID,NREPLIC)
      CALL SUMMARY(TUKEY,10,T42(IPID,NREPLIC,1),T42(IPID,NREPLIC,2),T42(
     1IPID,NREPLIC,3),T42(IPID,NREPLIC,4),T42(IPID,NREPLIC,5))
      T42(IPID,NREPLIC,6)=SUM/10.
      T42(IPID,NREPLIC,7)=SQRT((SQSM-10.*T42(IPID,NREPLIC,6)*T42(IPID,NR
     1EPLIC,6))/90.)
      T42(IPID,NREPLIC,8)=T42(IPID,NREPLIC,3)
      T42(IPID,NREPLIC,9)=(T42(IPID,NREPLIC,2)+2.*T42(IPID,NREPLIC,3)+T4
     12(IPID,NREPLIC,4))/4.
      T42(IPID,NREPLIC,10)=T42(IPID,NREPLIC,4)-T42(IPID,NREPLIC,2)
      T42(IPID,NREPLIC,11)=T42(IPID,NREPLIC,5)-T42(IPID,NREPLIC,1)
      SUM=0.
      SQSM=0.
      DO 9919 I=1,10
      SUM = SUM+QLMS51(I,IPID,NREPLIC)
      SQSM=SQSM+QLMS51(I,IPID,NREPLIC)*QLMS51(I,IPID,NREPLIC)
 9919 TUKEY(I)=QLMS51(I,IPID,NREPLIC)
      CALL SUMMARY(TUKEY,10,T51(IPID,NREPLIC,1),T51(IPID,NREPLIC,2),T51(
     1IPID,NREPLIC,3),T51(IPID,NREPLIC,4),T51(IPID,NREPLIC,5))
      T51(IPID,NREPLIC,6)=SUM/10.
      T51(IPID,NREPLIC,7)=SQRT((SQSM-10.*T51(IPID,NREPLIC,6)*T51(IPID,NR
     1EPLIC,6))/90.)
      T51(IPID,NREPLIC,8)=T51(IPID,NREPLIC,3)
      T51(IPID,NREPLIC,9)=(T51(IPID,NREPLIC,2)+2.*T51(IPID,NREPLIC,3)+T5
     11(IPID,NREPLIC,4))/4.
      T51(IPID,NREPLIC,10)=T51(IPID,NREPLIC,4)-T51(IPID,NREPLIC,2)
      T51(IPID,NREPLIC,11)=T51(IPID,NREPLIC,5)-T51(IPID,NREPLIC,1)
      SUM=0.
      SQSM=0.
      DO 9920 I=1,10
      SUM = SUM+QLMS52(I,IPID,NREPLIC)
      SQSM=SQSM+QLMS52(I,IPID,NREPLIC)*QLMS52(I,IPID,NREPLIC)
 9920 TUKEY(I)=QLMS52(I,IPID,NREPLIC)
      CALL SUMMARY(TUKEY,10,T52(IPID,NREPLIC,1),T52(IPID,NREPLIC,2),T52(
     1IPID,NREPLIC,3),T52(IPID,NREPLIC,4),T52(IPID,NREPLIC,5))
```

```
      T52(IPID,NREPLIC,6)=SUM/10.
      T52(IPID,NREPLIC,7)=SQRT((SQSM-10.*T52(IPID,NREPLIC,6)*T52(IPID,NR
     1EPLIC,6))/90.)
      T52(IPID,NREPLIC,8)=T52(IPID,NREPLIC,3)
      T52(IPID,NREPLIC,9)=(T52(IPID,NREPLIC,2)+2.*T52(IPID,NREPLIC,3)+T5
     12(IPID,NREPLIC,4))/4.
      T52(IPID,NREPLIC,10)=T52(IPID,NREPLIC,4)-T52(IPID,NREPLIC,2)
      T52(IPID,NREPLIC,11)=T52(IPID,NREPLIC,5)-T52(IPID,NREPLIC,1)
      SUM=0.
      SQSM=0.
      DO 9921 I=1,10
      SUM = SUM+QLMS61(I,IPID,NREPLIC)
      SQSM=SQSM+QLMS61(I,IPID,NREPLIC)*QLMS61(I,IPID,NREPLIC)
 9921 TUKEY(I)=QLMS61(I,IPID,NREPLIC)
      CALL SUMMARY(TUKEY,10,T61(IPID,NREPLIC,1),T61(IPID,NREPLIC,2),T61(
     1IPID,NREPLIC,3),T61(IPID,NREPLIC,4),T61(IPID,NREPLIC,5))
      T61(IPID,NREPLIC,6)=SUM/10.
      T61(IPID,NREPLIC,7)=SQRT((SQSM-10.*T61(IPID,NREPLIC,6)*T61(IPID,NR
     1EPLIC,6))/90.)
      T61(IPID,NREPLIC,8)=T61(IPID,NREPLIC,3)
      T61(IPID,NREPLIC,9)=(T61(IPID,NREPLIC,2)+2.*T61(IPID,NREPLIC,3)+T6
     11(IPID,NREPLIC,4))/4.
      T61(IPID,NREPLIC,10)=T61(IPID,NREPLIC,4)-T61(IPID,NREPLIC,2)
      T61(IPID,NREPLIC,11)=T61(IPID,NREPLIC,5)-T61(IPID,NREPLIC,1)
      SUM=0.
      SQSM=0.
      DO 9922 I=1,10
      SUM = SUM+QLMS62(I,IPID,NREPLIC)
      SQSM=SQSM+QLMS62(I,IPID,NREPLIC)*QLMS62(I,IPID,NREPLIC)
 9922 TUKEY(I)=QLMS62(I,IPID,NREPLIC)
      CALL SUMMARY(TUKEY,10,T62(IPID,NREPLIC,1),T62(IPID,NREPLIC,2),T62(
     1IPID,NREPLIC,3),T62(IPID,NREPLIC,4),T62(IPID,NREPLIC,5))
      T62(IPID,NREPLIC,6)=SUM/10.
      T62(IPID,NREPLIC,7)=SQRT((SQSM-10.*T62(IPID,NREPLIC,6)*T62(IPID,NR
     1EPLIC,6))/90.)
      T62(IPID,NREPLIC,8)=T62(IPID,NREPLIC,3)
      T62(IPID,NREPLIC,9)=(T62(IPID,NREPLIC,2)+2.*T62(IPID,NREPLIC,3)+T6
     12(IPID,NREPLIC,4))/4.
      T62(IPID,NREPLIC,10)=T62(IPID,NREPLIC,4)-T62(IPID,NREPLIC,2)
      T62(IPID,NREPLIC,11)=T62(IPID,NREPLIC,5)-T62(IPID,NREPLIC,1)
 9925 CONTINUE
      NREPLIC=1
      WRITE(5,5100) (ALABEL(I),I=1,3)
 5100 FORMAT(63X*TABLE 7.5*///* DATA SUMMARIES, CENTRAL VALUES, AND SPRE
     1ADS, MULT. BY 10, OVER 10 DIRICHLET SIMULATIONS FOR Q.L. REG-EST M
```

```
     2SE.  FIRST REPLIC.  DESIGN 2.*///57X,3A10/////)
4990 NSS=25
     PID=0.15
     ITABLE=0
     WRITE(5,5105)
5105 FORMAT(19X,53H********************DATA  SUMMARY******************
    1*5X,36H***********CENTRAL VALUES***********4X17H*****SPREADS*****/
    2* ESTIMATOR SS PID  L EXTREME    L HINGE    MEDIAN     U HINGE  U
    3EXTREME*7X*MEAN     (S.E.)     MEDIAN  TRIMEAN    MIDSPREAD  RANGE*
    4/)
4991 WRITE(5,5510) NSS,PID,(T11(IPID,NREPLIC,K),K=1,11),(T21(IPID,NREP
    1LIC,K),K=1,11),(T31(IPID,NREPLIC,K),K=1,11),(T41(IPID,NREPLIC,K),K
    2=1,11),(T51(IPID,NREPLIC,K),K=1,11),(T61(IPID,NREPLIC,K),K=1,11)
5510 FORMAT(3X*APMRO*3X,I2,F4.2,5F11.7,3XF9.6*(*F9.7*)*2F9.6,3X,2F9.6/3
    1X*PMDRO*9X,5F11.7,3X,F9.6*(*F9.7*)*2F9.6,3X,2F9.6/3X*MLE*11X,5F11.
    27,3X,F9.6*(*F9.7*)*2F9.6,3X,2F9.6//3X*APMR1*9X,5F11.7,3X,F9.6*(*F9
    3.7*)*2F9.6,3X,2F9.6//3X*APMR2*9X,5F11.7,3X,F9.6*(*F9.7*)*2F9.6,3X,
    42F9.6/3X*PMDR2*9X,5F11.7,3X,F9.6*(*F9.7*)*2F9.6,3X,2F9.6/)
     ITABLE=ITABLE+1
     GO TO (4992,4993,4994,4995) ITABLE
4992 PID=0.40
     GO TO 4991
4993 NSS=50
     PID=0.15
     GO TO 4991
4994 PID=0.40
     GO TO 4991
4995 WRITE(5,5112)
5112 FORMAT(///3X* NOTE THAT A ZERO BEFORE A DECIMAL DENOTES AN EXACT Z
    1ERO.  OTHERWISE, THE ZERO IS ROUNDED.*)
     IF (NREPLIC-1) 4996,4996,4998
4996 NREPLIC=2
     WRITE(5,4997)
4997 FORMAT(/////)
     WRITE(5,5115) (ALABEL(I),I=1,3)
5115 FORMAT(63X*TABLE 7.6*///* DATA SUMMARIES, CENTRAL VALUES, AND SPRE
    1ADS, MULT. BY 10, OVER 10 DIRICHLET SIMULATIONS FOR Q.L. REG-EST M
    2SE.  SECOND REPLIC.  DESIGN 2.*///57X,3A10/////)
     GO TO 4990
4998 CONTINUE
     END
```

```
      FUNCTION GAMMA(GG)
C
C     GENERATE A GAMMA RANDOM VARIABLE
C     TO DO SO USE ALGORITHM GT FROM AHRENS AND DIETER (1974) "COMPUTER
C     METHODS FOR SAMPLING FROM GAMMA, BETA, POISSON, AND BINOMIAL
C     DISTRIBUTIONS", P229, COMPUTING.  VOL. 12
C
C     NOTE THAT FOR 1/77 SIMULATION STUDY, GG RANGES FROM 0.1 TO 9.8
C
C     OBTAIN INTEGER PART OF GG
C
      K=GG
C
C     OBTAIN FRACTIONAL PART OF GG
C
      F=GG-K
C
C     OBTAIN INTEGER PART OF GAMMA
C
      GI=0.
      IF (K-0) 14,14,8
    8 I=0
      GP=1.
   10 I=I+1
      UU=URAN(0.)
      GP=GP*UU
      IF (I-K) 10,12,12
   12 GI=-ALOG(GP)
C
C     OBTAIN FRACTIONAL PART OF GAMMA
C
   14 GF=0.
      IF (F-0.) 40,40,15
   15 B=(2.7182818284590+F)/2.7182818284590
      OF=1./F
      FMIN1=F-1.
   16 UU=URAN(0.)
      GP=B*UU
C
C     GENERATE NEW UNIFORM RANDOM NUMBER FOR TESTS IN FOLLOWING STEPS
C     18 AND 30
C
      UU=URAN(0.)
      IF (GP-1.) 18,18,30
C
```

```
C       GF IS LESS THAN OR EQUAL TO 1.
C
   18 GF=GP**OF
      TEST=EXP(-GF)
      IF (UU-TEST) 40,40,16
C
C       GF IS GREATER THAN 1.
C
   30 GF=-ALOG((B-GP)*OF)
      TEST=GF**FMIN1
      IF (UU-TEST) 40,40,16
   40 GAMMA=GI+GF
      RETURN
      END
```

```
C
C
      SUBROUTINE GENXZ(UU,NSS)
C
C     GENERATE MULTINOMIAL COMPLETE (X) AND INCOMPLETE (Z) DATA
C     ALSO CALCULATE MAXIMUM-LIKELIHOOD-ESTIMATES FROM COMPLETE DATA
C
      COMMON DEP(3,3),DQL(4,3),E(4,3),IROBUST,NTS,P1,P2,P3,TPID
      COMMON/CALEST/APMC11,APMC12,APMC22,CONVCRI,COVSKIP,DMLC1,DMLC2,DML
     1C3,DPID,EAPMC11,EAPMC12,EAPMC22,EPMC11,EPMC12,EPMC22,ISTOP,N12,N13
     2,N23,PID,PMLC1,PMLC2,PMLC3,SS,SSN,TIMAP,TIMEP,TIM(2),TIM21,TIM31,X
     3NU1,XNU2,XNU3,Z1,Z2,Z3,Z12,Z13,Z23,Z1N,Z2N,Z3N
      COMMON/DATA/XDATA(2),ZDATA(6)
      DIMENSION UU(NSS)
      INTEGER Y1,Y2,W1,W3,V2,V3,COVSKIP
      EQUIVALENCE (E(1,1),PEPM1),(E(1,2),PEPM2),(E(1,3),PEPM3)
      EQUIVALENCE (E(2,1),PMLE1),(E(2,2),PMLE2),(E(2,3),PMLE3)
      EQUIVALENCE (E(3,1),PPMD1),(E(3,2),PPMD2),(E(3,3),PPMD3)
      EQUIVALENCE (E(4,1),PAPM1),(E(4,2),PAPM2),(E(4,3),PAPM3)
      EQUIVALENCE (DEP(1,1),EML1),  (DEP(1,2),EML2),  (DEP(1,3),EML3)
      EQUIVALENCE (DEP(2,1),EPMD1), (DEP(2,2),EPMD2), (DEP(2,3),EPMD3)
      EQUIVALENCE (DEP(3,1),EAPMN1),(DEP(3,2),EAPMN2),(DEP(3,3),EAPMN3)
      EQUIVALENCE (DQL(1,1),DEPMN1),(DQL(1,2),DEPMN2),(DQL(1,3),DEPMN3)
      EQUIVALENCE (DQL(2,1),DML1),  (DQL(2,2),DML2),  (DQL(2,3),DML3)
      EQUIVALENCE (DQL(3,1),DPMD1), (DQL(3,2),DPMD2), (DQL(3,3),DPMD3)
      EQUIVALENCE (DQL(4,1),DAPMN1),(DQL(4,2),DAPMN2),(DQL(4,3),DAPMN3)
C
C     RECALL THAT NSS IS THE INTEGER FORM OF THE SAMPLE SIZE SS
C
C     CALCULATE APPROX AMOUNT OF DATA GOING INTO EACH OF THE 4 GROUPS
C
      H4=PID/2.
      H3=H4
      H2=H4
      H1=1.-3.*H2
      E3=H1+H2
C
C     SET END POINTS
C
      P12=P1+P2
      E1=P1*E3
      E2=P12*E3
      E4=E3+P1*H2
      E5=E3+P12*H2
      E6=H1+2.*H2
```

```
      E7=E6+P1*H2
      E8=E6+P12*H2
C
C     INITIALIZE Z AND DUMMY VARIABLES YI, WI, AND VI
C
      Z1=0.
      Z2=0.
      Z3=0.
      Y1=0
      Y2=0
      W1=0
      W3=0
      V2=0
      V3=0
C
C      GENERATE X, Z DATA
C
C
C     CALL UNIFORM PSEUDO RANDOM-NUMBER GENERATOR
C     X(N+1) = (43490275647445.*X(N)) MOD(2EXP(48))
C     SPECTRAL NUMBERS C(2)=2.839, C(3)=2.095, C(4)=1.819, C(5)=0.978
C
      CALL URANV(0.,NSS,UU)
C
      DO 85 I=1,NSS
      U=UU(I)
      IF (E1-U) 2,2,40
    2 IF (E2-U) 4,4,45
    4 IF (E3-U) 6,6,50
    6 IF (E4-U) 8,8,55
    8 IF (E5-U) 10,10,60
   10 IF (E6-U) 12,12,65
   12 IF (E7-U) 14,14,70
   14 IF (E8-U) 80,80,75
   40 Z1=Z1+1.
      GO TO 85
   45 Z2=Z2+1.
      GO TO 85
   50 Z3=Z3+1.
      GO TO 85
   55 Y1=Y1+1
      GO TO 85
   60 Y2=Y2+1
      GO TO 85
   65 W3=W3+1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      GO TO 85
   70 W1=W1+1
      GO TO 85
   75 V2=V2+1
      GO TO 85
   80 V3=V3+1
   85 CONTINUE
      N12=Y1+Y2
      N13=W1+W3
      N23=V2+V3
C
C     OBTAIN REAL FORM OF SHARED INCOMPLETE DATA
C
      Z12=N12
      Z13=N13
      Z23=N23
C
C     OBTAIN COMPLETE DATA X
C
      X1=Z1+Y1+W1
      X2=Z2+Y2+V2
      X3=Z3+W3+V3
C
C     CALCULATE COMPLETE-DATA MAXIMUM-LIKELIHOOD ESTIMATES
C
      PMLC1=X1/SS
      PMLC2=X2/SS
      PMLC3=1.-PMLC1-PMLC2
      IF (PMLC3-0.) 90,95,95
   90 ISTOP=1
      PRINT 92, XNU1,XNU2,XNU3,P1,P2,P3,PID,SS,X1,X2,X3,Z1,Z2,Z3,Z12,Z13
     1,Z23,PMLC1,PMLC2,PMLC3,NSS,NTS
   92 FORMAT(///* GENXZ. COMPLETE-DATA MLE NEGATIVE P.  XNU=*3F10.4,* GE
     1NERATED P=*3F10.6,* PID=*F6.2,* SS=*F4.0/* X=*3F6.0,* Z=*6F6.0,* P
     2ML=*3F10.6,* NSS=*I3,4X,I3* TH TRINOMIAL SIMUL*)
      RETURN
   95 DMLC1=PMLC1-P1
      DMLC2=PMLC2-P2
      DMLC3=PMLC3-P3
      Z1N=Z1+XNU1
      Z2N=Z2+XNU2
      Z3N=Z3+XNU3
C
C     PUT X AND Z DATA INTO SEPARATE BLOCK COMMON BECAUSE PROGRAM WON"T
C     CORRECTLY RUN IF BLANK COMMON X,Z DATA IS PUT INTO KTITER, COUNTS,
```

```
C      AND BESTEST (PERHAPS PROBLEMS WITH E EQUIVALENCE STATEMENTS)
C
       XDATA(1)=X1
       XDATA(2)=X2
       ZDATA(1)=Z1
       ZDATA(2)=Z2
       ZDATA(3)=Z3
       ZDATA(4)=Z12
       ZDATA(5)=Z13
       ZDATA(6)=Z23
C
C      TRUE PERCENT (PROPORTION) INCOMPLETE DATA
C
       TPID=(Z12+Z13+Z23)/SS
       DPID=PID-TPID
       RETURN
       END
```

```
      SUBROUTINE EPM
C
C     PROGRAM TO CALCULATE EXACT POSTERIOR MEAN AND COVARIANCE MATRICES
C
      COMMON DEP(3,3),DQL(4,3),E(4,3),IROBUST,NTS,P1,P2,P3,TPID
      COMMON/CALEST/APMC11,APMC12,APMC22,CONVCRI,COVSKIP,DMLC1,DMLC2,DML
     1C3,DPID,EAPMC11,EAPMC12,EAPMC22,EPMC11,EPMC12,EPMC22,ISTOP,N12,N13
     2,N23,PID,PMLC1,PMLC2,PMLC3,SS,SSN,TIMAP,TIMEP,TIM(2),TIM21,TIM31,X
     3NU1,XNU2,XNU3,Z1,Z2,Z3,Z12,Z13,Z23,Z1N,Z2N,Z3N
      INTEGER COVSKIP
      EQUIVALENCE (E(1,1),PEPM1),(E(1,2),PEPM2),(E(1,3),PEPM3)
      EQUIVALENCE (E(2,1),PMLE1),(E(2,2),PMLE2),(E(2,3),PMLE3)
      EQUIVALENCE (E(3,1),PPMD1),(E(3,2),PPMD2),(E(3,3),PPMD3)
      EQUIVALENCE (E(4,1),PAPM1),(E(4,2),PAPM2),(E(4,3),PAPM3)
      EQUIVALENCE (DEP(1,1),EML1),  (DEP(1,2),EML2),  (DEP(1,3),EML3)
      EQUIVALENCE (DEP(2,1),EPMD1), (DEP(2,2),EPMD2), (DEP(2,3),EPMD3)
      EQUIVALENCE (DEP(3,1),EAPMN1),(DEP(3,2),EAPMN2),(DEP(3,3),EAPMN3)
      EQUIVALENCE (DQL(1,1),DEPMN1),(DQL(1,2),DEPMN2),(DQL(1,3),DEPMN3)
      EQUIVALENCE (DQL(2,1),DML1),  (DQL(2,2),DML2),  (DQL(2,3),DML3)
      EQUIVALENCE (DQL(3,1),DPMD1), (DQL(3,2),DPMD2), (DQL(3,3),DPMD3)
      EQUIVALENCE (DQL(4,1),DAPMN1),(DQL(4,2),DAPMN2),(DQL(4,3),DAPMN3)
C
      N121=N12+1
      N131=N13+1
      N231=N23+1
    3 Y1=Z1N
      YY1A=GAM(Y1)
      IF (Z12-0.) 4,4,8
    4 IF (Z13-0.) 5,5,8
    5 IF (Z23-0.) 6,6,8
C
C     COMPLETE-DATA CASE, ALL ZIJ=0.
C
    6 SUM12=YY1A*GAM(Z2N)*GAM(Z3N)
      S12P1=Z1N*SUM12
      S12P1SQ=(Z1N+1.)*S12P1
      S12P2=Z2N*SUM12
      S12P2SQ=(Z2N+1.)*S12P2
      S12P1P2=Z1N*Z2N*SUM12
      GO TO 55
    8 Z2N12=Z2N+Z12
      Z3N1323=Z3N+Z13+Z23
C
C     SIJPK DENOTES ZIJ SUM FOR POSTERIOR MEAN P(K) CALCULATIONS.
C     SIMILARLY, TIJPK DENOTES A TERM OF THIS SUM.
```

```
C
      S12P1=0.
      S12P2=0.
      S12P1SQ=0.
      S12P2SQ=0.
      S12P1P2=0.
      SUM12=0.
C
      DO 50 IIA=1,N121
      IA=IIA-1
      IF (IA-0) 9,9,10
    9 CON12=1.
      Y2=Z2N12
      YY2A=GAM(Y2)
      GAMMY3=GAM(Z3N1323)
      GO TO 16
   10 CON12=(Z12-IA+1.)*CON12/IA
      YY1A=Y1*YY1A
      Y1=Z1N+IA
      Y2=Z2N12-IA
      YY2A=YY2A/Y2
   16 SUM13=0.
      S13P1=0.
      S13P2=0.
      S13P1SQ=0.
      S13P2SQ=0.
      S13P1P2=0.
      Z1NIA=Z1N+IA
C
      DO 40 IIB=1,N131
      IB=IIB-1
      IF (IB-0) 17,17,20
   17 CON13=1.
      YY1B=YY1A
      Y3=Z3N1323
      YY3B=GAMMY3
      GO TO 27
   20 BI=IB-1
      CON13=(Z13-BI)*CON13/IB
      Y3=Z3N1323-IB
      YY3B=YY3B/Y3
      YY1B=YY1B*(Y1+BI)
   27 YY2C=YY2A
      SUM23=0.
      S23P2=0.
```

```
      S23P2SQ=0.
C
      DO 35 IIC=1,N231
      IC=IIC-1
      IF (IC-0) 30,30,31
   30 CON23=1.
      YY3C=YY3B
      GO TO 34
   31 CI=IC-1
      CON23=(Z23-CI)*CON23/IC
      YY2C=(Y2+CI)*YY2C
      YY3C=YY3C/(Y3-IC)
   34 T23=CON23*YY2C*YY3C
      F=Y2+IC
      T23P2=T23*F
      T23P2SQ=T23P2*(F+1.)
      SUM23=SUM23+T23
      S23P2=S23P2+T23P2
      S23P2SQ=S23P2SQ+T23P2SQ
   35 CONTINUE
C
      G=CON13*YY1B
      GG=Z1NIA+IB
      T13=G*SUM23
      T13P1=T13*GG
      T13P2=G*S23P2
      T13P1SQ=T13P1*(GG+1.)
      T13P2SQ=G*S23P2SQ
      T13P1P2=T13P2*GG
      SUM13=SUM13+T13
      S13P1=S13P1+T13P1
      S13P2=S13P2+T13P2
      S13P1SQ=S13P1SQ+T13P1SQ
      S13P2SQ=S13P2SQ+T13P2SQ
      S13P1P2=S13P1P2+T13P1P2
   40 CONTINUE
C
      T12=CON12*SUM13
      T12P1=CON12*S13P1
      T12P2=CON12*S13P2
      T12P1SQ=CON12*S13P1SQ
      T12P2SQ=CON12*S13P2SQ
      T12P1P2=CON12*S13P1P2
      SUM12=T12+SUM12
      S12P1=S12P1+T12P1
```

```
      S12P2=S12P2+T12P2
      S12P1SQ=S12P1SQ+T12P1SQ
      S12P2SQ=S12P2SQ+T12P2SQ
      S12P1P2=S12P1P2+T12P1P2
   50 CONTINUE
C
C     ELEMENTS OF POSTERIOR MEAN OF P GIVEN Z
C
   55 D1=SUM12*SSN
      D2=D1*(SSN+1.)
      PEPM1=S12P1/D1
      PEPM2=S12P2/D1
      PEPM3=1.-PEPM1-PEPM2
      IF (PEPM3-0.) 230,201,201
  201 CALL SECOND(TIM2)
C
C     ELEMENTS OF POSTERIOR COVARIANCE MATRIX OF P GIVEN Z
C
      X1X2=S12P1P2/D2
      EPMC12=X1X2-PEPM1*PEPM2
      X1SQ=S12P1SQ/D2
      EPMC11=X1SQ-PEPM1*PEPM1
      X2SQ=S12P2SQ/D2
      EPMC22=X2SQ-PEPM2*PEPM2
      DEPMN1=PEPM1-P1
      DEPMN2=PEPM2-P2
      DEPMN3=PEPM3-P3
      RETURN
  230 PRINT 231
  231 FORMAT(////* EXACT POSTERIOR MEAN.  NEGATIVE P.*//)
      PRINT 240, PEPM1,EPMC11,X1SQ,S12P1,S12P1SQ,PEPM2,EPMC22,X2SQ,S12P2
     1,S12P2SQ,PEPM3,EPMC12,X1X2,SUM12,S12P1P2,NTS
  240 FORMAT(//7H PEPM1=E15.8,4X,8H EPMC11=E15.8,4X,6H X1SQ=E15.8,4X,7H
     1S12P1=E15.8,4X,9H S12P1SQ=E15.8/7H PEPM2=E15.8,4X,8H EPMC22=E15.8,
     24X,6H X2SQ=E15.8,4X,7H S12P2=E15.8,4X,9H S12P2SQ=E15.8/7H PEPM3=E1
     35.8,4X,8H EPMC12=E15.8,4X,6H X1X2=E14.7,4X,7H SUM12=E14.7,4X,9H S1
     42P1P2=E14.7,* NTS=*I3/)
      ISTOP=1
      RETURN
      END
```

```
      FUNCTION GAM(X)
C
C
C     EVALUATE GAM(X) FOR CASES FROM PRIOR (.1,.1,9.8)
C
      IF (X-9.2) 1,35,35
C
C     ARGUMENT IS LESS THAN OR EQUAL TO 9.1.  TO INSURE 11 SIGNIFICANT
C     PLACES OF ACCURACY IN GAMMA, USE 11 SIGNIFICANT-FIGURE VALUE
C     CALCULATED FROM LOG(GAM(X)) FROM ABRAMOWITZ AND STEGUN OR DAVIS.
C     CALCULATE GAM(0.1), GAM(1.1), AND GAM(2.1) BY HAND FROM GAM(3.1)
C     AND RELATION GAM(X+1)= X GAM(X)
C
    1 IF (ABS(X-9.1)-1.E-13) 2,2,4
C     GAM(9.1)
    2 GAM=49973.708949629
      RETURN
    4 IF (ABS(X-8.1)-1.E-13) 6,6,8
C     GAM(8.1)
    6 GAM=6169.5936974851
      RETURN
    8 IF (ABS(X-7.1)-1.E-13) 10,10,12
C     GAM(7.1)
   10 GAM=868.95685880072
      RETURN
   12 IF (ABS(X-6.1)-1.E-13) 14,14,16
C     GAM(6.1)
   14 GAM=142.45194406569
      RETURN
   16 IF (ABS(X-5.1)-1.E-13) 18,18,20
C     GAM(5.1)
   18 GAM=27.931753738371
      RETURN
   20 IF (ABS(X-4.1)-1.E-13) 22,22,24
C     GAM(4.1)
   22 GAM=6.8126228630175
      RETURN
   24 IF (ABS(X-3.1)-1.E-13) 26,26,28
C     GAM(3.1)
   26 GAM=2.1976202783927
      RETURN
   28 IF (ABS(X-2.1)-1.E-13) 30,30,32
C     GAM(2.1)
   30 GAM=1.046485846854
      RETURN
```

```
   32 IF (ABS(X-1.1)-1.E-13) 34,34,36
C     GAM(1.1)
   34 GAM=0.95135076987
      RETURN
C     GAM(0.1)
   36 GAM=9.5135076987
      RETURN
C
C     ARGUMENT X IS GREATER THAN OR EQUAL TO 10.
C     USE STIRLING"S FORMULA TO OBTAIN AN APPROXIMATION TO GAMMA
C     THAT IS ACCURATE TO 11 SIGNIFICANT FIGURES
C
   35 XSQ=X*X
      XCU=XSQ*X
      XFIFTH=XSQ*XCU
C
C     Y IS THE APPROXIMATED NATURAL LOG(BASE E) OF GAMMA(X)
C
      Y=(X-0.5)*ALOG(X)-X+0.91893853320467+1./(12.*X)-1./(360.*XCU)+1./(
     11260.*XFIFTH)
      IF (X-22.) 40,45,45
   40 Y=Y-1./(1680.*XFIFTH*XSQ)
   45 GAM=EXP(Y)
      RETURN
      END
```

```
      SUBROUTINE METHODS
C
C     FOR INCOMPLETE DATA CALCULATE MY APPROXIMATION, POSTERIOR MODE,
C     AND MAXIMUM-LIKELIHOOD ESTIMATE
C
      INTEGER COVSKIP
      DIMENSION A(3,3),B(3,1),IPIVOT(3),IWK(6)
      COMMON DEP(3,3),DQL(4,3),E(4,3),IROBUST,NTS,P1,P2,P3,TPID
      COMMON/CALEST/APMC11,APMC12,APMC22,CONVCRI,COVSKIP,DMLC1,DMLC2,DML
     1C3,DPID,EAPMC11,EAPMC12,EAPMC22,EPMC11,EPMC12,EPMC22,ISTOP,N12,N13
     2,N23,PID,PMLC1,PMLC2,PMLC3,SS,SSN,TIMAP,TIMEP,TIM(2),TIM21,TIM31,X
     3NU1,XNU2,XNU3,Z1,Z2,Z3,Z12,Z13,Z23,Z1N,Z2N,Z3N
      COMMON/ITKT/AVNUMIT(6),CTNUMIT(6,7)
      EQUIVALENCE (E(1,1),PEPM1),(E(1,2),PEPM2),(E(1,3),PEPM3)
      EQUIVALENCE (E(2,1),PMLE1),(E(2,2),PMLE2),(E(2,3),PMLE3)
      EQUIVALENCE (E(3,1),PPMD1),(E(3,2),PPMD2),(E(3,3),PPMD3)
      EQUIVALENCE (E(4,1),PAPM1),(E(4,2),PAPM2),(E(4,3),PAPM3)
      EQUIVALENCE (DEP(1,1),EML1),  (DEP(1,2),EML2),  (DEP(1,3),EML3)
      EQUIVALENCE (DEP(2,1),EPMD1), (DEP(2,2),EPMD2), (DEP(2,3),EPMD3)
      EQUIVALENCE (DEP(3,1),EAPMN1),(DEP(3,2),EAPMN2),(DEP(3,3),EAPMN3)
      EQUIVALENCE (DQL(1,1),DEPMN1),(DQL(1,2),DEPMN2),(DQL(1,3),DEPMN3)
      EQUIVALENCE (DQL(2,1),DML1),  (DQL(2,2),DML2),  (DQL(2,3),DML3)
      EQUIVALENCE (DQL(3,1),DPMD1), (DQL(3,2),DPMD2), (DQL(3,3),DPMD3)
      EQUIVALENCE (DQL(4,1),DAPMN1),(DQL(4,2),DAPMN2),(DQL(4,3),DAPMN3)
C
C     CHECK TO INSURE THAT CONVERGENCE CRITERION IS NOT TOO STRICT
C
      IF (IROBUST-1) 1,100,50
C
C     INCOMPLETE-DATA MAXIMUM-LIKELIHOOD ESTIMATE
C
    1 PMLE1=PEPM1
      PMLE2=PEPM2
      PMLE3=PEPM3
      K=1
    5 PL1=PMLE1
      PL2=PMLE2
      PMLE12=PMLE1+PMLE2
      PMLE13=PMLE1+PMLE3
      PMLE23=PMLE2+PMLE3
      IF (PMLE12-1.E-14) 7,7,8
    7 TEMP=0.
      GO TO 9
    8 TEMP=Z12/PMLE12
    9 PMLE1=(Z1+PMLE1*(TEMP+Z13/PMLE13))/SS
```

```
      PMLE2=(Z2+PMLE2*(TEMP+Z23/PMLE23))/SS
      PMLE3=1.-PMLE1-PMLE2
      IF (PMLE3-0.) 21,14,14
C
   14 IF (PMLE1-0.1) 15,15,16
   15 IF (ABS(PMLE1-PL1)-0.00001) 17,20,20
   16 IF (ABS(PMLE1-PL1)/PMLE1-CONVCRI) 17,20,20
   17 IF (PMLE2-0.1) 18,18,19
   18 IF (ABS(PMLE2-PL2)-0.00001) 25,20,20
   19 IF (ABS(PMLE2-PL2)/PMLE2-CONVCRI) 25,20,20
   20 K=K+1
      IF (K-1000) 5,5,21
   21 PRINT 22, K,XNU1,P1,PL1,PL2,CONVCRI,IROBUST,NTS,TPID,PMLE3,XNU2,P2
     1,PMLE1,PMLE2
   22 FORMAT(//  * EXCESS NUMBER OF ITERATIONS FOR INCOMPLETE-DATA M.L.E
     1. IS*I3,* XNU1=*F9.4,* P1=*F9.6,3X*PL1=*F11.8,3X*PL2=*F11.8/* CONV
     2(RD)=*F7.5* IROBUST=*I2* NTS=*I3* TPID=*F6.2* PMLE3=*F6.4* XNU2=*F
     39.4* P2=*F9.6* PMLE1=*F11.8* PMLE2=*F11.8)
      IF (K-1030) 25,25,250
C
C     CONVERGENCE FOR MAXIMUM-LIKELIHOOD ESTIMATE INCOMPLETE DATA
C
   25 EML1=PMLE1-PEPM1
      EML2=PMLE2-PEPM2
      EML3=PMLE3-PEPM3
      DML1=PMLE1-P1
      DML2=PMLE2-P2
      DML3=PMLE3-P3
C
C     INCREMENT COUNTER FOR NUMBER OF ITERATIONS FOR CONVERGENCE
C
      CALL KTITER(K,1)
C
C
C     POSTERIOR MODE
C
   50 T1=Z1N-1.
      T2=Z2N-1.
      K=1
      D=SSN-3.
      PPMD1=PEPM1
      PPMD2=PEPM2
      PPMD3=PEPM3
   55 PL1=PPMD1
      PL2=PPMD2
```

```
      PPMD12=PPMD1+PPMD2
      PPMD13=PPMD1+PPMD3
      PPMD23=PPMD2+PPMD3
      IF (PPMD12-1.E-14) 57,57,58
   57 TEMP=0.
      GO TO 59
   58 TEMP=Z12/PPMD12
   59 PPMD1=(T1+PPMD1*(TEMP+Z13/PPMD13))/D
      IF (PPMD1.LT.0.) PPMD1=0.
      PPMD2=(T2+PPMD2*(TEMP+Z23/PPMD23))/D
      IF (PPMD2.LT.0.) PPMD2=0.
      PPMD3=1.-PPMD1-PPMD2
      IF (PPMD3-0.) 71,64,64
   64 IF (PPMD1-0.1) 65,65,66
   65 IF (ABS(PPMD1-PL1)-0.00001) 67,70,70
   66 IF (ABS(PPMD1-PL1)/PPMD1-CONVCRI) 67,70,70
   67 IF (PPMD2-0.1) 68,68,69
   68 IF (ABS(PPMD2-PL2)-0.00001) 75,70,70
   69 IF (ABS(PPMD2-PL2)/PPMD2-CONVCRI) 75,70,70
   70 K=K+1
      IF (K-1000) 55,55,71
   71 PRINT 72, K,XNU1,P1,PL1,PL2,CONVCRI,IROBUST,NTS,TPID,PPMD3,XNU2,P2
     1,PPMD1,PPMD2
   72 FORMAT(//  * EXCESS NUMBER OF ITERATIONS FOR INCOMPLETE-DATA PPMD
     1. IS*I3,* XNU1=*F9.4,* P1=*F9.6,3X*PL1=*F11.8,3X*PL2=*F11.8/* CONV
     2(RD)=*F7.5* IROBUST=*I2* NTS=*I3* TPID=*F6.2* PPMD3=*F6.4* XNU2=*F
     39.4* P2=*F9.6* PPMD1=*F11.8* PPMD2=*F11.8)
      IF (K-1030) 75,75,250
C
C     CONVERGENCE FOR POSTERIOR MODE INCOMPLETE DATA
C
   75 EPMD1=PPMD1-PEPM1
      EPMD2=PPMD2-PEPM2
      EPMD3=PPMD3-PEPM3
      DPMD1=PPMD1-P1
      DPMD2=PPMD2-P2
      DPMD3=PPMD3-P3
C
C     INCREMENT COUNTER FOR NUMBER OF ITERATIONS FOR CONVERGENCE
C
      J=2
      IF (IROBUST.EQ.2) J=5
      CALL KTITER(K,J)
C
C
```

```
C     MY TAYLOR-SERIES APPROXIMATED POSTERIOR MEAN AND COVARIANCE
C     MATRICES
C
  100 PAPM1=PEPM1
      PAPM2=PEPM2
      PAPM3=PEPM3
      K=1
  105 PL1=PAPM1
      PL2=PAPM2
      PL3=PAPM3
      PAPM12=PAPM1+PAPM2
      PAPM13=PAPM1+PAPM3
      PAPM23=PAPM2+PAPM3
      IF (PAPM12-1.E-14) 107,107,108
  107 TEMP=0.
      GO TO 109
  108 TEMP=Z12/PAPM12
  109 PAPM1=(Z1N+PAPM1*(TEMP+Z13/PAPM13))/SSN
      PAPM2=(Z2N+PAPM2*(TEMP+Z23/PAPM23))/SSN
      PAPM3=1.-PAPM1-PAPM2
      IF (PAPM3-0.) 121,114,114
C
  114 IF (PAPM1-0.1) 115,115,116
  115 IF (ABS(PAPM1-PL1)-0.00001) 117,120,120
  116 IF (ABS(PAPM1-PL1)/PAPM1-CONVCRI) 117,120,120
  117 IF (PAPM2-0.1) 118,118,119
  118 IF (ABS(PAPM2-PL2)-0.00001) 125,120,120
  119 IF (ABS(PAPM2-PL2)/PAPM2-CONVCRI) 125,120,120
  120 K=K+1
      IF (K-1000) 105,105,121
  121 PRINT 122, K,XNU1,P1,PL1,PL2,CONVCRI,IROBUST,NTS,TPIO,PAPM3,XNU2,P
     12,PAPM1,PAPM2
  122 FORMAT(//  * EXCESS NUMBER OF ITERATIONS FOR INCOMPLETE-DATA PAPM
     1. IS*I3,* XNU1=*F9.4,* P1=*F9.6,3X*PL1=*F11.8,3X*PL2=*F11.8/* CONV
     2(RO)=*F7.5* IROBUST=*I2* NTS=*I3* TPIO=*F6.2* PAPM3=*F6.4* XNU2=*F
     39.4* P2=*F9.6* PAPM1=*F11.8* PAPM2=*F11.8)
      IF (K-1030) 125,125,250
C
C     CONVERGENCE FOR T.S. APPROX. POSTERIOR MEAN INCOMPLETE DATA
C
  125 EAPMN1=PAPM1-PEPM1
      EAPMN2=PAPM2-PEPM2
      EAPMN3=PAPM3-PEPM3
      DAPMN1=PAPM1-P1
      DAPMN2=PAPM2-P2
```

```
      DAPMN3=PAPM3-P3
C
C     INCREMENT COUNTER FOR NUMBER OF ITERATIONS FOR CONVERGENCE
C
      IF (IROBUST-1) 127,128,129
  127 J=3
      GO TO 130
  128 J=4
      GO TO 130
  129 J=6
  130 CALL KTITER(K,J)
      IF (IROBUST.GT.0) RETURN
C
C     APPROXIMATED POSTERIOR VAR/COV MATRIX.   NONITERATIVE METHOD.
C
  150 P12=PAPM1+PAPM2
      P13=PAPM1+PAPM3
      P23=PAPM2+PAPM3
C
C     CAUTION.  INSURE THAT P12, P13, AND P23 ARE NOT IN COMMON FROM
C     GENERATED P1, P2, AND P3
C
      R12=PAPM1/P12
      R13=PAPM1/P13
      R21=PAPM2/P12
      R23=PAPM2/P23
C
C     SSN = SUM OF DATA PLUS SUM OF PRIOR PARAMETERS XNUI
C
      T=SSN*(SSN+1.)
      P12SQ=P12*P12
      P13SQ=P13*P13
      P23SQ=P23*P23
      ZRP21=Z12*R21/P12
      ZRP13=Z13*R13/P13
      ZRP12=Z12*R12/P12
      ZRP23=Z23*R23/P23
C CALCULATE A(1,1)
      A112=(ZRP21+Z13/P13)**2/T
      A113=(ZRP21*R21)/(P12*T)
      A114=Z13/(P13SQ*T)
      A(1,1)=-1.+A112-A113-A114
C CALCULATE A(1,2)
      A121=(ZRP13-ZRP12)*(ZRP21+Z13/P13)
      A122=ZRP12*R21/P12
```

```
      A123=ZRP13/P13
      A(1,2)=2.*(A121+A122-A123)/T
      TEMP=Z12/P12SQ
C CALCULATE A(1,3)
      A131=(ZRP12-ZRP13)**2
      A132=ZRP12*R12/P12+ZRP13*R13/P13
      A(1,3)=(A131-A132)/T
C CALCULATE B(1,1)
      B(1,1)=-(SSN*PAPM1*P23+ZRP12*PAPM2+ZRP13*PAPM3)/T
C
C CALCULATE A(2,1)
      A211=(ZRP21+Z13/P13)*(ZRP23-ZRP21)
      A212=TEMP*R21**2
      A(2,1)=(A211+A212)/T
C CALCULATE A(2,2)
      A2221=TEMP*PAPM1+Z23*(1.-2.*PAPM1)/P23SQ
      A2222=TEMP*PAPM2+Z13*(1.-2.*PAPM2)/P13SQ
      A222=A2221*A2222
      A223=TEMP*(Z12-2.)*PAPM1*PAPM2/P12SQ
      A224=Z13*Z23*(P12-2.*PAPM1*PAPM2)/(P13SQ*P23SQ)
      A(2,2)=(-T+A222+A223+A224)/T
C CALCULATE A(2,3)
      A231=(ZRP12+Z23/P23)*(ZRP13-ZRP12)
      A232=TEMP*R12**2
      A(2,3)=(A231+A232)/T
C CALCULATE B(2,1)
      B(2,1)=PAPM1*PAPM2*(SSN+TEMP)/T
C
C CALCULATE A(3,1)
      A311=(ZRP21-ZRP23)**2
      A312=ZRP21*R21/P12
      A313=ZRP23*R23/P23
      A(3,1)=(A311-A312-A313)/T
C CALCULATE A(3,2)
      A321=(-ZRP12-Z23/P23)*(ZRP21-ZRP23)
      A322=ZRP12*R21/P12
      A323=ZRP23/P23
      A(3,2)=2.*(A321+A322-A323)/T
C CALCULATE A(3,3)
      A332=(ZRP12+Z23/P23)**2
      A333=ZRP12*R12/P12
      A334=Z23/P23SQ
      A(3,3)=(-T+A332-A333-A334)/T
C CALCULATE B(3,1)
      B(3,1)=-(SSN*PAPM2*P13+ZRP21*PAPM1+ZRP23*PAPM3)/T
```

```
C
C     SOLVE SYSTEM A*X=B FOR X.  X IS VECTOR OF COVARIANCES C11,C12,C22
C
      CALL MATINV(3,3,A,1,B,1,DETERM,ISCALE,IPIVOT,IWK)
      IF (ABS(DETERM)-5.0E-14) 212,212,220
  212 PRINT 214, DETERM,K,XNU1,XNU2,P1,P2,PAPM1,PAPM2,((A(I,J),J=1,3),B(
     1I,1),I=1,3)
  214 FORMAT(///* SINGULAR SYSTEM. DETERM=*F18.14,* NUMBER OF ITERATIONS
     1 WAS*I2,* XNU1=*F6.1,* XNU2=*F6.1,* P1=*F9.6,* P2=*F9.6,* PAPM1=*F
     29.6/* PAPM2=*F9.6,* AX=B  MID-CALC IS*3E20.8,5X,*X1*,3X,*=*,E23.8/
     334X,3E20.8,5X,*X2*,3X,*=*,E23.8/34X,3E20.8,5X,*X3*,3X,*=*,E23.8/)
      GO TO 230
C
C     DIFFERENCE OF APPROXIMATED POSTERIOR COVARIANCES FROM EXACT
C     POSTERIOR COVARIANCES.   C11, C12, AND C22
C
  220 APMC11=B(1,1)
      APMC12=B(2,1)
      APMC22=B(3,1)
      IF (APMC11-0.) 225,225,221
  221 IF (APMC22-0.) 225,225,222
  222 EAPMC11=APMC11-EPMC11
      EAPMC12=APMC12-EPMC12
      EAPMC22=APMC22-EPMC22
      RETURN
  225 PRINT 226, APMC11,APMC12,APMC22,EPMC11,EPMC12,EPMC22,XNU1,XNU2,PAP
     1M1,PAPM2,NTS
  226 FORMAT(//* APPROXIMATED VARIANCE IS NEGATIVE.  APMC11=*E21.14,* AP
     1MC12=*E18.11,* APMC22=*E18.11/* EPMC11=*E18.11,* EPMC12=*E18.11,*
     2EPMC22=*E20.13,* XNU1=*F4.0,* XNU2=*F4.0,* PAPM1=*F5.3,* PAPM2=*F5
     4.3,* NTS=*I3/)
  230 COVSKIP=1
      RETURN
  250 ISTOP=1
      RETURN
      END
```

```
      SUBROUTINE COUNTS(BIAS,RELDIFF,J)
C
C     COUNT NUMBER OF NUMXZ UNTERMINATED TRIALS THAT HAVE NEGETAIVE,
C     ZERO, AND POSITIVE BIAS AND THAT HAVE ABSOLUTE RELATIVE
C     DIFFERENCES LESS THAN CERTAIN PERCENTAGES.
C
C     BIAS = (APPROX-EXACT) OR (APPROX-GENERATED P)
C     RELDIFF = ABS(BIAS/EXACT) OR ABS(BIAS/GENERATED P)
C     (RECALL THAT COV IS NEG SO WANT DENOMINATOR INCLUDED IN ABS VALUE)
C
C     J DENOTES, IN SUBSEQUENT ORDER, ONE OF EAPMC11, EAPMC12, EAPMC22,
C     (BIAS OF APPROX T.S. EXPANSION FOR EXACT POSTERIOR COV)
C     DMLC1 AND DMLC2 (COMPLETE-DATA MLE BIAS FROM GENERATED OR GIVEN P)
C     (THUS, J=3 REFERS TO EAPMC22)
C
      COMMON DEP(3,3),DQL(4,3),E(4,3),IROBUST,NTS,P1,P2,P3,TPID
      COMMON/BIASRD/COUNTB(3,8),COUNTRD(8,8)
      AB=ABS(BIAS)
      IF (AB-1.E-15) 3,3,1
    1 IF (BIAS-0.) 2,2,4
C     NEGATIVE BIAS
    2 COUNTB(1,J)=COUNTB(1,J)+1.
      GO TO 5
C     ZERO BIAS (CDC 6600 COMPUTER ACCURACY IS 14 SIGN FIGURES BUT
C     CONSIDER ONLY 15 DECIMAL PLACES FOR ZERO BIAS
    3 COUNTB(2,J)=COUNTB(2,J)+1.
      GO TO 5
C     POSITIVE BIAS
    4 COUNTB(3,J)=COUNTB(3,J)+1.
C
C     25 PERCENT RELATIVE DIFFERENCE
    5 IF (RELDIFF-0.25)   8, 8,30
    8 COUNTRD(8,J)=COUNTRD(8,J)+1.
C     20 PERCENT RELATIVE DIFFERENCE
      IF (RELDIFF-0.20)  10,10,30
   10 COUNTRD(7,J)=COUNTRD(7,J)+1.
C     15 PERCENT RELATIVE DIFFERENCE
      IF (RELDIFF-0.15)  12,12,30
   12 COUNTRD(6,J)=COUNTRD(6,J)+1.
C     10 PERCENT RELATIVE DIFFERENCE
      IF (RELDIFF-0.10)  14,14,30
   14 COUNTRD(5,J)=COUNTRD(5,J)+1.
C      5 PERCENT RELATIVE DIFFERENCE
      IF (RELDIFF-0.05)  16,16,30
   16 COUNTRD(4,J)=COUNTRD(4,J)+1.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C        1 PERCENT RELATIVE DIFFERENCE
      IF (RELDIFF-0.01)  18,18,30
   18 COUNTRD(3,J)=COUNTRD(3,J)+1.
C       .1 PERCENT RELATIVE DIFFERENCE
      IF (RELDIFF-0.001) 20,20,30
   20 COUNTRD(2,J)=COUNTRD(2,J)+1.
C       .01 PERCENT RELATIVE DIFFERENCE
      IF (RELDIFF-.0001) 22,22,30
   22 COUNTRD(1,J)=COUNTRD(1,J)+1.
   30 CONTINUE
      IF (RELDIFF-0.15) 40,31,31
   31 IF (J-4) 33,40,32
   32 IF (J-5) 40,40,33
   33 PRINT 35, J,NTS,BIAS,RELDIFF,TPID,IROBUST,E(1,1),E(1,2),E(4,1),E(4
     1,2)
   35 FORMAT(* SUBR COUNTS.  J=*I2*  NTS=*I3*  BIAS=*F9.7*  RELDIFF=*F5.
     12*  TPID=*F6.2*  IROBUST=*I2*  PEPM1,2=*2F7.4*  PAPM1,2=*2F7.4)
   40 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ESTMSE(Y,Y2,XY,X,X2,TXMSE,N,MSE)
C
C     CALCULATE ESTIMATES OF MSE AND SAMPLE VARIANCE OF THESE ESTIMATES
C
      REAL MSE(6),MSECV
C
C     FOR TERM=(PE1-P1)**2+(PE2-P2)**2+(PE3-P3)**2 AND CONTROL-VARIATE
C         TERMCV=(PMLECD1-P1)**2+(PMLECD2-P2)**2+(PMLECD3-P3)**2
C     FOR PE- DENOTING ONE OF ESTIMATORS EPM, APM, PMD, AND MLE AND
C     PMLECD- DENOTING COMPLETE-DATA MLE
C
C     Y  = SUM OF N TERM
C     Y2 = SUM OF N TERM*TERM
C     XY = SUM OF N TERM*TERMCV
C     X  = SUM OF N TERMCV
C     X2 = SUM OF N TERMCV*TERMCV
C     N  = NUMBER OF TERMS
C     TXMSE = TRUE MEAN SQ ERROR OF CONTROL VARIATE
C     MSECV = USUAL SAMPLE MSE FOR THE CONTROL VARIATE
C
C     MSE(1) IS USUAL MSE (B=0 IN MSE REGRESSION ESTIMATE)
C     MSE(2) IS VAR OF MSE(1)
C     MSE(3) IS USUAL CONTROL-VARIATE MSE (B=1 IN MSE REGRESSION EST)
C     MSE(4) IS VAR OF MSE(3)
C     MSE(5) IS LEAST-SQUARES REGRESSION ESTIMATE MSE (B=LEAST-SQS EST)
C     MSE(6) IS VAR OF MSE(5)
C
C     NOTE THAT MSE(5) SHOULD HAVE SMALLEST VARIANCE.  HOWEVER, IT WILL
C     BE A BIASED ESTIMATE.  HENCE, USE IT IN ANALYSES ONLY IF IT
C     DIFFERS FROM EITHER ONE OF TWO UNBIASED ESTIMATES BY NO MORE
C     THAN 1%.  (IE, IT CAN DIFFER BY MORE THAN 1% FROM EITHER MSE(1) OR
C     MSE(3) BUT NOT BOTH.)
C
C     GENERAL FORM OF ESTIMATED MSE IS
C
C                       MSE=MSE(1)+B*(TXMSE-MSECV)
C
C     USUAL MSE (B=0)
C
      XN=N*(N-1)
      MSE(1)=Y/N
      MSE(2)=(Y2-N*MSE(1)*MSE(1))/XN
C
C     USUAL CONTROL-VARIATE MSE (B=1)
C
```

```
      XN=N*(N-2)
      T0=2.*XY
      MSECV=X/N
      T1=N*MSECV
      T2=2.*N*TXMSE
      T3=T2*TXMSE/2.
      D=TXMSE-MSECV
      MSE(3)=MSE(1)+D
      MSE(4)=(Y2-T0+X2+T2*(MSE(1)-MSECV)+T3-N*MSE(3)*MSE(3))/XN
C
C     LEAST-SQUARES REGRESSION-ESTIMATE MSE (B=LEAST-SQUARES ESTIMATE)
C
      B=(XY-T1*MSE(1))/(X2-T1*MSECV)
      B2=B*B
      MSE(5)=MSE(1)+B*D
      MSE(6)=(Y2-B*T0+B2*X2+B*T2*(MSE(1)-B*MSECV)+B2*T3-N*MSE(5)*MSE(5))
     1/XN
      RETURN
      END
```

```
      SUBROUTINE KTITER(K,J)
C
C     INCREMENT COUNTERS FOR (1) AVERAGING NUMBER OF ITERATIONS AN
C     ESTIMATOR REQUIRED AND (2) DETERMINING HOW MANY CASES IN A
C     REPLICATION TOOK A SPECIFIED NUMBER OF ITERATIONS
C
C     K IS NUMBER OF ITERATIONS REQUIRED TO MEET CONVERGENCE CRITERION
C     J DENOTES, IN SUBSEQUENT ORDER, ONE OF ESTIMATORS MLE, PMDRO,
C     APMRO, APMR1, PMDR2, APMR2 (THUS, J=4 REFERS TO APMR1)
C
      COMMON DEP(3,3),DQL(4,3),E(4,3),IROBUST,NTS,P1,P2,P3,TPID
      COMMON/DATA/XDATA(2),ZDATA(6)
      COMMON/ITKT/AVNUMIT(6),CTNUMIT(6,10)
C
C     FOR AVERAGING NUMBER OF ITERATIONS FOR JTH ESTIMATOR
C
      AVNUMIT(J)=AVNUMIT(J)+K
C
C     INCREMENT COUNTER FOR NUMBER OF ITERATIONS
C
      I=1
      IF (K-1)  20,20,2
    2 I=2
      IF (K-2)  20,20,3
    3 I=3
      IF (K-3)  20,20,4
    4 I=4
      IF (K-4)  20,20,5
    5 I=5
      IF (K-5)  20,20,6
    6 I=6
      IF (K-6)  20,20,7
    7 I=7
      IF (K-7)  20,20,8
    8 I=8
      IF (K-10) 20,20,9
    9 I=9
      IF (K-15) 20,20,10
   10 I=10
   20 CTNUMIT(J,I)=CTNUMIT(J,I)+1
      IF (K-25) 30,25,25
   25 PRINT 27, NTS,K,J,TPID,XDATA(1),XDATA(2),IROBUST,((E(II,JJ),JJ=1,2
     2),II=1,4),(ZDATA(II),II=1,6)
   27 FORMAT(* SUBR KTITER.  NTS=*I3*  NUMBER OF ITER IS*I4*  FOR METHOD
     1 J=*I1* (MLE,PMDRO,APMRO,APMR1,PMDR2,APMR2).  TPID=*F4.2*  X(C.D.)
```

```
  2=*2F7.2/5X* IROBUST=*I1*  PEPM1,2=*2F6.4*  PMLE1,2=*2F6.4*  PPMD1,
  32=*2F6.4*  PAPM1,2=*2F6.4*  Z=*6F4.0)
30 CONTINUE
   RETURN
   END
```

```
      SUBROUTINE SUMMARY(X,N,LE,LH,M,UH,UE)
C
C     TUKEY"S FIVE-POINT DATA SUMMARY.  ROUTINE SORTS INPUT VECTOR X OF
C     LENGTH N AND THEN CALCULATES LOWER EXTREME LE, LOWER HINGE LH,
C     MEDIAN M, UPPER HINGE UH, AND UPPER EXTREME UE.
C     REFERENCE "EXPLORATORY DATA ANALYSIS" BY JOHN W. TUKEY
C     FORTRAN EXTENDED VERSION 4.6, CDC 6600 COMPUTER (14 SIGN FIG S.P.)
C     PROGRAMER IS KAREN R CREDEUR, NASA, LANGLEY RESEARCH CENTER
C
      DIMENSION X(N)
      REAL LE,LH,M
C
C     SORT DATA IN ASCENDING ORDER
C
      CALL ASORT(X,1,N)
C
C     MEDIAN
C
      XN=(N+1.)/2.
      K=XN+1.E-12
      M=X(K)
      IF (ABS(XN-K)-1.E-8) 5,5,1
    1 M=(M+X(K+1))/2.
C
C     HINGES
C
    5 XN=(K+1)/2.
      K=XN+1.E-12
      LH=X(K)
      UH=X(N+1-K)
      IF (ABS(XN-K)-1.E-8) 15,15,10
   10 LH=(LH+X(K+1))/2.
      UH=(X(N-K)+UH)/2.
C
C     EXTREMES
C
   15 LE=X(1)
      UE=X(N)
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE BESTEST(BESTQL)
C
C     BY TWO DIFFERENT CRITERION (SUMMED ABSOLUTE RELATIVE DIFFERENCE
C     RELD-- AND SUMMED SQUARED ERROR SE-- FOR SUM BEING OVER THE THREE
C     COMPONENTS OF AN ESTIMATOR) DETERMINE WHICH ESTIMATOR IS BEST FOR
C     A GIVEN ONE OF THE TRINOMIAL SIMULATION TRIALS
C
C     TIES.  SCORE AS BEST EACH ESTIMATOR THAT TIES FOR BEST.
C
C
C     THE FOUR ESTIMATORS IN E SHOULD BE IN THE FOLLOWING ORDER PEPM,
C     PMLE, PPMD, PAPM.   BIASES SHOULD BE IN CORRESPONDING ORDER.
C
      DIMENSION BESTQL(4,2),RDEP(3),RDQL(4),RELDEP(3,3),RELDQL(4,3)
      DIMENSION SEEP(3),SEQL(4),V(4),W(4),X(3),Y(3)
      COMMON DEP(3,3),DQL(4,3),E(4,3),IROBUST,NTS,P1,P2,P3,TPID
      COMMON/BEST/BESTEP(3,2),CTRDEP,CTRDQL(3),PRDEP(9,3),PRDQL(9,7),SBI
     1ASEP(3,3),SBIASQL(3,7)
      COMMON/DATA/XDATA(2),ZDATA(6)
C
      IR=IROBUST+1
      IF (IROBUST-1) 1,100,150
C
C     FOR EPM COMPARISONS
C
    1 DO 10 I=1,3
      SEEP(I)=0.
      DO 2 J=1,3
      X(I)=SEEP(I)=SEEP(I)+DEP(I,J)*DEP(I,J).
    2 CONTINUE
C
C     INCORPORATE SUBROUTINE COUNTS TWICE (ONCE FOR EACH OF EP AND QL
C     COMPARISONS) IN THIS SUBROUTINE TO SAVE PROGRAM EXECUTION COST OF
C     MANY SUBROUTINE CALLS AND INDEX RESETTINGS.
C
C     DETERMINE SIGN OF FIRST COMPONENT OF ESTIMATOR
C
      IF (ABS(DEP(I,1))-1.E-13) 5,5,3
    3 IF (DEP(I,1)-0.) 4,4,6
C     NEGATIVE BIAS
    4 SBIASEP(1,I)=SBIASEP(1,I)+1.
      GO TO 10
C     ZERO BIAS (CONSIDER ONLY 13 DECIMAL PLACES)
    5 SBIASEP(2,I)=SBIASEP(2,I)+1.
      GO TO 10
```

```
C     POSITIVE BIAS
    6 SBIASEP(3,I)=SBIASEP(3,I)+1.
   10 CONTINUE
      CALL ASORT(X,1,3)
      DO 20 I=1,3
      IF (SEEP(I).EQ.X(1)) BESTEP(I,1)=BESTEP(I,1)+1.
   20 CONTINUE
C
C     DETERMINE WHETHER ANY PEPM COMPONENT IS ZERO
C
      ICK=0
      IF (E(1,1)-1.E-10) 30,25,25
   25 IF (E(1,2)-1.E-10) 30,26,26
   26 IF (E(1,3)-1.E-10) 30,32,32
   30 ICK=1
   32 IF (ICK-0) 33,33,56
   33 CTRDEP=CTRDEP+1.
      DO 35 I=1,3
      RDEP(I)=0.
      DO 34 J=1,3
      RELDEP(I,J)=ABS(DEP(I,J))/E(1,J)
      Y(I)=RDEP(I)=RDEP(I)+RELDEP(I,J)
   34 CONTINUE
   35 CONTINUE
      CALL ASORT(Y,1,3)
      DO 40 I=1,3
      IF (RDEP(I).EQ.Y(1)) BESTEP(I,2)=BESTEP(I,2)+1.
   40 CONTINUE
C
C     FOR DETERMINING PROPORTION OF CASES FOR WHICH % ABSOLUTE RELATIVE
C     DIFFERENCE FOR EACH OF ALL THREE ESTIMATOR COMPONENTS IS LESS THAN
C     SPECIFIED AMOUNTS (INCORPORATED IN PART FROM SUBROUTINE COUNTS)
C
      DO 55 I=1,3
      II=1
      DO 53 J=1,3
      GO TO (41,43,45,47,49,51,520,528,53) II
   41 IF (RELDEP(I,J)-0.0001) 53,42,42
   42 II=2
   43 IF (RELDEP(I,J)-0.001)  53,44,44
   44 II=3
   45 IF (RELDEP(I,J)-0.01)   53,46,46
   46 II=4
   47 IF (RELDEP(I,J)-0.05)   53,48,48
   48 II=5
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
   49 IF (RELDEP(I,J)-0.10)   53,50,50
   50 II=6
   51 IF (RELDEP(I,J)-0.15)   53,52,52
   52 II=7
  520 IF (RELDEP(I,J)-0.20)   53,525,525
  525 II=8
  528 IF (RELDEP(I,J)-0.25)   53,530,530
  530 II=9
   53 CONTINUE
      PRDEP(II,I)=PRDEP(II,I)+1.
   55 CONTINUE
      IF (RELDEP(3,1)-0.15) 541,543,543
  541 IF (RELDEP(3,2)-0.15) 56,543,543
  543 PRINT 544, NTS,IROBUST,TPID,RELDEP(3,1),RELDEP(3,2),((E(I,J),J=1,2
     1),I=1,4,3),((E(I,J),J=1,2),I=2,3),((DEP(I,J),J=1,2),I=1,3),XDATA(1
     1),XDATA(2),(ZDATA(I),I=1,6)
  544 FORMAT(* SUBR BESTEST.  NTS=*I3*  IROBUST=*I1*  TPID=*F4.2*  RELDE
     1P(3,1-2)=*2F5.2*  PEPM1,2=*2F6.4*  PAPM1,2=*2F6.4*  PMLE1,2=*2F5.3
     2/15X*PPMD1,2=*2F7.4*  DEP=*3(2F7.4,3X),* X,Z=*2F4.0,2X,6F4.0)
C
C     FOR QL COMPARISONS, ROUTINE IS CALLED FOR EACH OF THREE
C     ROBUSTNESS SETS
C
C
C     ROBUSTNESS SET 0.  ORIGINAL PRIOR.
C
   56 I1=1
      L=2
      ICK=0
C
C     DETERMINE WHETHER ANY P COMPONENT IS ZERO
C
      IF (P1-1.E-10) 59,57,57
   57 IF (P2-1.E-10) 59,58,58
   58 IF (P3-1.E-10) 59,61,61
   59 ICK=1
   61 DO 75 I=I1,4
      SEQL(I)=0.
      DO 610 J=1,3
      V(I)=SEQL(I)=SEQL(I)+DQL(I,J)*DQL(I,J)
  610 CONTINUE
      IF (IR-2) 62,615,64
  615 IF (I-3) 75,75,63
   62 K=I
      GO TO 65
```

```
   63 K=5
      GO TO 65
   64 K=I+3
   65 IF (ABS(DQL(I,1))-1.E-13) 69,69,67
   67 IF (DQL(I,1)-0.) 68,68,70
C     NEGATIVE BIAS
   68 SBIASQL(1,K)=SBIASQL(1,K)+1.
      GO TO 75
C     ZERO BIAS (CONSIDER ONLY 13 DECIMAL PLACES)
   69 SBIASQL(2,K)=SBIASQL(2,K)+1.
      GO TO 75
C     POSITIVE BIAS
   70 SBIASQL(3,K)=SBIASQL(3,K)+1.
   75 CONTINUE
      CALL ASORT(V,L,4)
      DO 76 I=L,4
      IF (SEQL(I).EQ.V(L)) BESTQL(I,1)=BESTQL(I,1)+1.
   76 CONTINUE
C
C     UNSORT V FOR ROBUSTNESS SETS
C
      DO 770 I=L,4
      V(I)=SEQL(I)
  770 CONTINUE
C
C     IF ANY ESTIMATOR IS ZERO, SKIP DIV FOR RELATIVE DIFF FOR ALL EST.
C
      IF (ICK.GT.0) RETURN
      CTRDQL(IR)=CTRDQL(IR)+1.
      DO 78 I=I1,4
      RELDQL(I,1)=ABS(DQL(I,1))/P1
      RELDQL(I,2)=ABS(DQL(I,2))/P2
      RELDQL(I,3)=ABS(DQL(I,3))/P3
      RDQL(I)=0.
      DO 77 J=1,3
      W(I)=RDQL(I)=RDQL(I)+RELDQL(I,J)
   77 CONTINUE
   78 CONTINUE
      CALL ASORT(W,L,4)
      DO 79 I=L,4
      IF (RDQL(I).EQ.W(L)) BESTQL(I,2)=BESTQL(I,2)+1.
   79 CONTINUE
C
C     UNSORT W FOR ROBUSTNESS SETS
C
```

```
      DO 790 I=L,4
      W(I)=RDQL(I)
  790 CONTINUE
      IF (IROBUST.EQ.1) I1=4
      DO 98 I=I1,4
      II=1
      IF (IR-2) 80,81,82
   80 K=I
      GO TO 83
   81 K=5
      GO TO 83
   82 K=I+3
   83 DO 96 J=1,3
      GO TO (84,86,88,90,92,94,950,958,96) II
   84 IF (RELDQL(I,J)-0.0001) 96,85,85
   85 II=2
   86 IF (RELDQL(I,J)-0.001)  96,87,87
   87 II=3
   88 IF (RELDQL(I,J)-0.01)   96,89,89
   89 II=4
   90 IF (RELDQL(I,J)-0.05)   96,91,91
   91 II=5
   92 IF (RELDQL(I,J)-0.10)   96,93,93
   93 II=6
   94 IF (RELDQL(I,J)-0.15)   96,95,95
   95 II=7
  950 IF (RELDQL(I,J)-0.20)   96,955,955
  955 II=8
  958 IF (RELDQL(I,J)-0.25)   96,960,960
  960 II=9
   96 CONTINUE
      PRDQL(II,K)=PRDQL(II,K)+1.
   98 CONTINUE
      RETURN
C
C     ROBUSTNESS SET 1.  UNIFORM PRIOR.
C
  100 I1=3
      L=2
C
C     SET POSTERIOR MODE EQUAL TO M.L.E.
C
      DO 102 I=1,3
      DQL(3,I)=DQL(2,I)
  102 CONTINUE
```

```
      GO TO 61
C
C     ROBUSTNESS SET 2.  PERTURBED PRIOR
C
  150 I1=3
      GO TO 61
      END
```

APPENDIX A

LANGLEY LIBRARY FUNCTION URAN

Language: COMPASS

Purpose: URAN generates uniformly distributed random numbers over the interval (0,1).

Use: Y = URAN(X)

X An input real number on which three conditions exist:

X = 0, The next random number is generated and returned. If no previous call was made, a default seed of 17171274321477413155B is provided.

X < 0, A random number is not generated but the last previously generated random number (or the seed if no random number has been generated) is returned.

X > 0, The exponent part of X is set to 1717B and the low order bit is set to one. This result is returned as the seed of a new sequence, and any additional calls to URAN will be based on a sequence using this seed.

Method: This pseudorandom-number generator is multiplicative with algorithm

$$X_{i+1} = 43490275647445\ X_i \bmod(2^{48}).$$

Each random number is generated from the previous one by taking the lower order 48 bits of the 96 bit product produced by $X_{i+1} = 43490275647445\ X_i$. The exponent of the product is such that X_{i+1} is constrained to lie between 0 and 1.

Accuracy: The generator has a full period of 2^{46}. Extensive statistical testing for randomness and distribution were performed to establish its validity as a reliable random number generator.

SPECTRAL NUMBERS:	C_2	C_3	C_4	C_5
	2.839	2.095	1.819	0.978

References: (a) Ahrens, J. H. and Dieter, U.: "Computer Methods for Sampling from Gamma, Beta, Poisson, and Binomial Distributions," Computing 12, 1974, p 224.

(b) Ahrens, J. H., Dieter, U., and Grube: "Pseudo-Random Numbers: A New Proposal for the Choice of Multiplicators," Computing 6, 1970, pp 121-138.

(c) Knuth, Donald E.: The Art of Computer Programming, Vol. 2 (Seminumerical Algorithms). Addison-Wesley, Reading, Mass. 1969.

Storage: 13 octal locations

Subroutine date: March 1, 1977

APPENDIX B

LANGLEY LIBRARY SUBROUTINE URANV

Language: COMPASS

Purpose: URANV generates uniformly distributed random numbers over the interval (0,1).

Use: CALL URANV(X,N,V)

X An input real number on which three conditions exist:

X = 0, A vector of random numbers is generated using the last random number generated on the previous call as a seed. If no previous call was made, a default seed of 17171274321477413155B is provided.

X < 0, The last random number calculated by the routine, or the default seed if no previous call was made, is returned in V(1). V(2), ..., V(N) are not altered.

X > 0, The first random number is found by packing an exponent of 1717B and the coefficient part of X into V(1), and setting the low order bit to one. Random numbers V(2), ..., V(N) are then calculated using the algorithm given under METHOD.

N Input integer specifying the number of random numbers to be returned in V.

$N \leq 1$, V(1) is calculated and returned.

$N > 1$, V(1), ..., V(N) are calculated and returned.

V An output one-dimensional real array dimensioned at least N. On output, V will contain the N calculated random numbers.

Method: This pseudorandom-number generator is multiplicative with algorithm

$$X_{i+1} = 43490275647445\ X_i \bmod(2^{48}).$$

Each random number is generated from the previous one by taking the lower order 48 bits of the 96 bit product produced by $X_{i+1} = 43490275647445\ X_i$. The exponent of the product is such that X_{i+1} is constrained to lie between 0 and 1.

Accuracy: The generator has a full period of 2^{46}. Extensive statistical testing for randomness and distribution were performed to establish its validity as a reliable random number generator.

SPECTRAL NUMBERS:

C_2	C_3	C_4	C_5
2.839	2.095	1.819	0.978

References: (a) Ahrens, J. H. and Dieter, U.: "Computer Methods for Sampling from Gamma, Beta, Poisson, and Binomial Distributions," Computing 12, 1974, p 224.

(b) Ahrens, J. H., Dieter, U., and Grube: "Pseudo-Random Numbers: A New Proposal for the Choice of Multiplicators," Computing 6, 1970, pp 121-138.

(c) Knuth, Donald E.: The Art of Computer Programming, Vol. 2 (Seminumerical Algorithms). Addison-Wesley, Reading, Mass., 1969.

Storage: 25 octal locations

Subroutine date: March 1, 1977

APPENDIX C

LANGLEY LIBRARY SUBROUTINE MATINV

Language: FORTRAN

Purpose: MATINV solves the matrix equation $AX = B$, where A is a square coefficient matrix and B is a matrix of constant vectors. The solution to a set of simultaneous equations, the matrix inverse, and the determinant may be obtained.

Use: CALL MATINV(MAX,N,A,M,B,IOP,DETERM,ISCALE,IPIVOT,IWK)

MAX — The maximum order of A as stated in the DIMENSION statement of the calling program

N — The order of A; $1 \leq N \leq MAX$

A — A two-dimensional array of coefficients. On return to the calling program, A^{-1} is stored in A.

M — The number of column vectors in B. On return to the calling program, X is stored in B if $M > 0$; for $M = 0$, the subroutine is used only for inversion.

B — A two-dimensional array of the constant vectors B. On return to the calling program, X is stored in B.

IOP — Option to compute the determinant:
- 0 Compute the determinant.
- 1 Do not compute the determinant.

DETERM — Gives the value of the determinant by the formula $\mathrm{Det}(a) = 10^{(100 \times ISCALE)}(DETERM)$ when $IOP = 0$. For $IOP = 1$, the determinant is set to 1. For a singular matrix and $IOP = 0$ or $IOP = 1$, the determinant is set to zero.

ISCALE — A scale factor computed by the subroutine to keep the results of computation within the floating-point word size of the computer

IPIVOT — A one-dimensional array of temporary storage used by the subroutine

IWK — A two-dimensional array of temporary storage used by the subroutine

Restrictions: Arrays A, B, IPIVOT, and INDEX have variable dimensions in the subroutine. The maximum size of these arrays must be specified in a DIMENSION statement of the calling program as A(MAX,MAX), B(MAX,M), IPIVOT(MAX), and IWK(MAX,2). The original matrices A and B are destroyed. They must be saved

APPENDIX C

by the user if there is further need for them. The determinant is set to zero for a singular matrix.

Method: Jordan's method is used to reduce a matrix A to the identity matrix I through a succession of elementary transformations $\ell_n, \ell_{n-1}, \ldots, \ell_1$. If these transformations are simultaneously applied to I and to a matrix B of constant vectors, the results are A^{-1} and X where $AX = B$. Each transformation is selected so that the largest element is used in the pivotal position. (See ref. (a).)

Accuracy: Total pivotal strategy is used to minimize the rounding errors; however, the accuracy of the final results depends upon how well-conditioned the original matrix is. A return with DETERM $\neq 0$ does not guarantee accuracy in the solutions of inverse.

Reference: (a) Fox, L.: An Introduction to Numerical Linear Algebra. Oxford Univ. Press, 1965.

Storage: 516 octal locations

Subroutine date: January 1, 1975

REFERENCES

1. Credeur, Karen Rackley: Estimation From Incomplete Multinomial Data. Ph.D. Thesis, Harvard University, 1978. (Available as NASA TM 78703, 1978)

2. Dempster, A. P., Laird, N. M., and Rubin, D. B.: Maximum Likelihood from Incomplete Data Via the EM Algorithm. J. Royal Statist. Soc., Ser. B, Vol. 39, 1977, pp. 1-22.

3. CYBER 170 Series Computer Systems: CONTROL DATA MODIFY Reference Manual, Publ. No. 60281700, Arden Hills, Minn., 1974.

4. International Mathematical and Statistical Libraries, Inc.: Edition 5 Reference Manual, Houston, Texas, 1976.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM 78704		

4. Title and Subtitle	5. Report Date
A Computer Program For Estimation From Incomplete Multinomial Data	5/25/78
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Karen R. Credeur	
9. Performing Organization Name and Address	10. Work Unit No.
NASA Langley Research Center Hampton, Va. 23665	11. Contract or Grant No.
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
National Aeronautics and Space Administration Washington, D.C. 20546	Technical Memorandum
	14. Sponsoring Agency Code

15. Supplementary Notes

Describes the main computer program used for NASA TM 78703

16. Abstract

This paper describes and lists the main computer program written for results given in NASA TM 78703. Coding is given for maximum likelihood and Bayesian estimation of the vector $\underset{\sim}{p}$ of multinomial cell probabilities from incomplete data. Also included is coding to calculate and approximate elements of the posterior mean and covariance matrices. The program is written in FORTRAN IV language for the Control Data CYBER 170 series digital computer system with network operating system (NOS) 1.1. The program requires approximately 44000 octal locations of core storage. A typical case requires from 72 seconds to 92 seconds on CYBER 175 depending on the value of the prior parameter.

17. Key Words (Suggested by Author(s))

computer program,
estimation,
multinomial,
incomplete data,
simulation

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
unclassified	unclassified	73	

[redacted]