A COMPUTER PROGRAM FOR ESTIMATION FROM INCOMPLETE MULTINOMIAL DATA

Karen R. Credeur

May 1978

Because of its significant early commercial potential, this information, which has been developed under a U. S. Government program, is being disseminated within the United States in advance of general publication. This information may be duplicated and used by the recipient with the express limitation that it not be published. Release of this information to other domestic parties by the recipient shall be made subject to these limitations. Foreign release may be made only with prior NASA approval and appropriate export licenses. This legend shall be marked on any reproduction of this information in whole or in part.

Date for general release May 1980.
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 \( \mathbf{p} \) 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

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>1</td>
</tr>
<tr>
<td>INDEX OF COMPUTER PROGRAM</td>
<td>5</td>
</tr>
<tr>
<td>LISTING OF COMPUTER PROGRAM</td>
<td>6</td>
</tr>
<tr>
<td>APPENDIX A: URAN</td>
<td>67</td>
</tr>
<tr>
<td>APPENDIX B: URANV</td>
<td>69</td>
</tr>
<tr>
<td>APPENDIX C: MATINV</td>
<td>71</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>73</td>
</tr>
</tbody>
</table>
DESCRIPTION

This paper describes the main computer program used in reference 1 for estimating the vector 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 \( L(p-\hat{p}) = (p-\hat{p})'(p-\hat{p}) \) for \( \hat{p} \) an estimator of \( p \).

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 \( p \);

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 programming 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 \((4\times2\times3)\) having four levels* and two replications per cell. DESIGN 2 has a mixed-effects model constituting nested factorials \((4\times10\times2\times3)\) having four levels and two replications per cell within each of four variations of the prior parameter \(\nu\). The ten generations of the Dirichlet probability \(p\) in DESIGN 2 are considered random; the remaining factors are considered fixed.

---

*level 1: four Dirichlet probabilities \(p\)
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

\[ \begin{align*}
XNU1 &= \\
XNU2 &= \\
IP &= \\
SEED &= 
\end{align*} \]

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

\[ \text{DATA ALABEL/IOHA. NU = (,10HO.1,0.1,9.,10H8). /} \]
\[ \text{DATA TLABEL/IOHTABLE 7.1 /} \]

are changed as needed. For values of \( y \) that are less than 10, \( \text{GAM}(y+1) \) is calculated from the relationship \( \text{GAM}(y+1) = y \text{GAM}(y) \) and a starting value. For \( y \) an integer, the starting value is \( \text{GAM}(2) = 1 \) and for \( y = x \, \frac{1}{3} \), for \( x \) an integer, the starting value is \( \text{GAM}(3 \, \frac{1}{3}) = 2.7781584804296 \). 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 URANY 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.
<table>
<thead>
<tr>
<th>NAME</th>
<th>USAGE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN</td>
<td>main program</td>
<td>6</td>
</tr>
<tr>
<td>GAMMA</td>
<td>subroutine to generate a gamma random variable</td>
<td>35</td>
</tr>
<tr>
<td>GENXZ</td>
<td>subroutine to generate trinomial complete (x) and incomplete (z) data; also calculates complete-data maximum-likelihood estimate</td>
<td>37</td>
</tr>
<tr>
<td>EPM</td>
<td>subroutine to calculate exact posterior mean and covariance matrices</td>
<td>41</td>
</tr>
<tr>
<td>GAM</td>
<td>subroutine to evaluate the Gamma function by exact values or Stirling's approximation; note that this subroutine differs for each of the four different values of the prior parameter ( \nu=NU ) (only the subroutine for ( \nu=(0.1,0.1,9.8) ) is shown)</td>
<td>45</td>
</tr>
<tr>
<td>METHODS</td>
<td>subroutine to calculate the Taylor-series approximations APM and APC for the posterior mean and covariance matrices; also calculates the posterior mode PMD and the maximum-likelihood estimate MLE from incomplete data</td>
<td>47</td>
</tr>
<tr>
<td>COUNTS</td>
<td>subroutine for covariance approximations and complete-data maximum-likelihood estimate to count the number of the 200 trinomial simulation trials that have negative, zero, and positive error and that have absolute relative difference less than certain percentages</td>
<td>54</td>
</tr>
<tr>
<td>ESTMSE</td>
<td>subroutine to calculate the usual, control-variate, and regression estimates of mean squared error and their sample variances</td>
<td>56</td>
</tr>
<tr>
<td>KTITER</td>
<td>subroutine to increment counters for averaging the number of iterations an estimator requires and for determining how many of the 200 trinomial simulations an estimator requires a specified number of iterations</td>
<td>58</td>
</tr>
<tr>
<td>SUMMARY</td>
<td>subroutine to calculate Tukey's five-point data summary: median, hinges, and extremes</td>
<td>60</td>
</tr>
<tr>
<td>BESTEST</td>
<td>by two different criteria (summed absolute relative difference and summed squared error), subroutine determines which estimator is best for a given one of the 200 trinomial simulations; also includes a section corresponding to COUNTS for the estimators APM, PMD, and MLE</td>
<td>61</td>
</tr>
</tbody>
</table>
PROGRAM MAIN(INPUT, OUTPUT, TAPE5, TAPE12)

PROGAMED BY KAREN RACKLEY CREOPRE, SPRING 1977, CDC 6600,
FORTRAN EXTENDED VERSION 4.6, NASA, LANGLEY RESEARCH CENTER

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,PI,P2,P3,TPID
COMMON/BEST/BESTEP(3,2),CTROEP,CTROQL(3),PRDEP(19,3),PRDQL(9,7),SBI
1ASEP(3,3),SBIAQL(3,7)
COMMON/BIASRD/COUNTB(3,8),COUNTRO(8,8)
COMMON/CALEST/APMC11,APMC12,APMC22,CONVCR,COVSKIP,DMLC1,DMLC2,DML
1C3,DPID,EAPMC11,EAPMC12,EAPMC22,EPMC11,EPMC12,EPMC22,ISTOP,N12,N13
2,N23,PI,PMLC1,PMLC2,PMLC3,SS,SSN,TIMAP,TIMEP,TIM(2),TIM21,TIM31,X
3NU,XXNU,XXNU2,Z1,Z2,Z3,Z12,Z13,Z23,Z1N,Z2N,Z3N
COMMON/IKIK/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),EM1),(DEP(1,2),EM2),(DEP(1,3),EM3)
EQUIVALENCE (DEP(2,1),EPM1),(DEP(2,2),EPM2),(DEP(2,3),EPM3)
EQUIVALENCE (DEP(3,1),EPMN1),(DEP(3,2),EPMN2),(DEP(3,3),EPMN3)
EQUIVALENCE (DQL(1,1),DEPN1),(DQL(1,2),DEPN2),(DQL(1,3),DEPN3)
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)

DATA ALABEL/10H4. NU = (10H0,1,0,1,9,10H8)
DATA TLABEL/10HTABLE 7,1 /
CONVCRl RELATIVE-ERROR CONVERGENCE CRITERION (USUALLY 0.0001)
IP   DENOTES, IN THE FOLLOWING ORDER, ONE OF THE EXPECTED P'S
      (.01,.01,.98), (.1,.1,.8), (.2,.3,.5), AND (1/3,1/3,1/3)
NSS  INTEGER SS
NXZ  NUMBER OF TRINOMIAL (X AND Z DATA) SIMULATIONS
PAPMI I-TH T.S. APPROXIMATED POSTERIOR MEAN
PEPMI I-TH EXACT POSTERIOR MEAN
PI   I-TH GENERATED P
PID  PERCENTAGE OF INCOMPLETE DATA
PMCI I-TH COMPLETE M.L.E.
PMLEI I-TH INCOMPLETE-DATA M.L.E.
PPMOI I-TH POSTERIOR MODE
SSN  SS * SUM OF PRIOR PARAMETERS XNUI
XNU  VECTOR OF PRIOR PARAMETERS XNU=(XNU1,XNU2,XNU3)
ZI   NUMBER OF OBSERVATIONS FALLING IN CATEGORY I
ZIJ  NUMBER OF OBSERVATIONS SUCH THAT EACH OBSERVATION IS
     KNOWN TO FALL IN ONE OF CATEGORIES I AND J BUT IS NOT
     FURTHER CLASSIFIED
ZIN  ZI+XNU1

XNU1=0.1
XNU2=0.1
IP=2
SEED=24158739.
GSEED=SEED+100.

INITIALIZE ONE-DIMENSIONAL FORM OF UNIFORM RANDOM-NUMBER
GENERATOR FOR GENERATING GAMMA RANDOM VARIABLES

UN=URAN(GSEED)
PRINT 4, GSEED, UN
XNU3=10.-XNU1-XNU2
XNU=xnu1+xnu2+xnu3

GENERATE A 3-COMPONENT (2-DIM) VECTOR OF DIRICHLET PROBABILITIES

DO 9910 IGEN=1,10
   G1=gamma(XNU1)
   G2=gamma(XNU2)
   G3=gamma(XNU3)
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
   XNU2=G2/G
   XNU3=G3/G
   G=G1+G2+G3
   XNU1=G1/G
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, 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
PRDQL(I, J)=0.
27 CONTINUE
SBIASEP(I)=0.
28 CONTINUE
   DO 29 I=1,21
      SBIASOL(I)=0.
29 CONTINUE
   DO 30 I=1,27
      PRODEP(I)=0.
30 CONTINUE
   CTRDEP=CTRDEP(1)=CTRDEP(2)=CTRDEP(3)=0.

C INITIALIZE COUNTERS FOR NUMBER OF ITERATIONS FOR CONVERGENCE
   DO 40 II=1,6
      AVNUMIT(II)=0.
   DO 39 II=1,7
      CNUMIT(II,II)=0.
      CNUMIT(II,8)=CNUMIT(II,9)=CNUMIT(II,10)=0.
40 CONTINUE
   TIM(1)=TIM(2)=TIMEP=TIMAP=0.

C DO 42 J=1,2
   BESTEP(J,J)=BESTOL(J,J,1)=BESTOL(J,J,2)=BESTOL(J,J,3)=0.
   BESTEP(2,J)=BESTOL(2,J,1)=BESTOL(2,J,2)=BESTOL(2,J,3)=0.
   BESTEP(3,J)=BESTOL(3,J,1)=BESTOL(3,J,2)=BESTOL(3,J,3)=0.
   BESTOL(4,J,1)=BESTOL(4,J,2)=BESTOL(4,J,3)=0.
42 CONTINUE

C INITIALIZE COUNTERS FOR BIAS AND RELATIVE DIFFERENCES
   DO 45 K2=1,8
   DO 44 K1=1,8
      COUNTRO(K1,K2)=0.
      IF (K1-3) 43,43,44
43 COUNTB(K1,K2)=0.
44 CONTINUE
45 CONTINUE

C INITIALIZE AVERAGES AND ERROR SUMMARIES TO ZERO

C AVERAGE ESTIMATORS AND THEIR STANDARD ERRORS (S.E.)
   APMLE1= APPEPM1= APAPM1= APPMD1= APMLC1= AP1=0.
   XAPMLE1=XAPEPM1=XAPAPM1=XAPMD1=XAPMLC1=XAP1=0.
   APMLE2= APPEPM2= APAPM2= APPMD2= APMLC2= AP2=0.
   XAPMLE2=XAPEPM2=XAPAPM2=XAPMD2=XAPMLC2=XAP2=0.
AVPMN1 = AVPMN2 = AQPMN1 = AQPMN2 = AQPM0 = AQPM02 = 0.
APPMN1 = APPMN2 = APGPN1 = APGPN2 = APGPN0 = APGPN02 = 0.

AV BIAS, MSE, & VAR(MSE) OF ESTIMATORS FROM EXACT POSTERIOR MEAN

XAPMN1 = XAPMN2 = XAPMN4 = XAPMN0 = 0.
XPM01 = XPM02 = XPM04 = XPM00 = 0.
XML1 = XML2 = XML4 = XML0 = 0.

AV BIAS, MSE, & VAR(MSE) OF ESTIMATORS FROM GIVEN OR GENERATED P

YEPMN1 = YAPMN1 = YPM01 = YML1 = YMLC1 = VAPMN1 = QAPMN1 = QPM01 = 0.
YEPMN2 = YAPMN2 = YPM02 = YML2 = YMLC2 = VAPMN2 = QAPMN2 = QPM02 = 0.
YEPMN4 = YAPMN4 = YPM04 = YML4 = VAPMN4 = QAPMN4 = QPM04 = 0.
YEPMSQ = YAPMSQ = YPM0SQ = YMLSQ = VAPMSQ = QAPMSQ = QPM0SQ = 0.
REGAPM0 = REGPMD0 = REGAPM0 = REGMLC0 = YMLC20 = 0.
REGAPM1 = REGPMD1 = REGAPM1 = REGMLC1 = YMLC21 = 0.
REGAPM2 = REGPMD2 = REGAPM2 = REGMLC2 = YMLC22 = 0.

INITIALIZE FOR S.E. OF AVERAGE BIAS

WAPMN1 = WPM01 = WML1 = WAPMN2 = WPM02 = WML2 = 0.
UMLC1 = UEPMN1 = UAPMN1 = UPM01 = UML1 = FAPMN1 = GAPMN1 = GPM01 = 0.
UMLC2 = UEPMN2 = UAPMN2 = UPM02 = UML2 = FAPMN2 = GAPMN2 = GPM02 = 0.

EST AVERAGE, BIAS, S.E., AV % REL DIFF, & MSE FOR COV EST FROM EP

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 = VARK4 = 0.

GENERATE NUMXZ SETS OF TRINOMIAL DATA, CALCULATE ESTIMATORS, AND
COMPARE THEM AS APPROXIMATIONS FOR THE EXACT POSTERIOR MEAN.
ALSO ASSESS HOW WELL THE T.S. APPROXIMATION DOES FOR THE EXACT
POSTERIOR COVARIANCE MATRIX. COMPARE ESTIMATORS WITH GENERATED
P AND DO EXPLORATORY ROBUSTNESS STUDY

DO 65 NT=1, NXZ
NTS=NT
IROBUST=0

GENERATE MULTINOMIAL COMPLETE DATA X AND INCOMPLETE DATA Z
CALCULATE COMPLETE-DATA MAXIMUM LIKELIHOOD ESTIMATE

CALL GENXZ(W0,NSS)
IF (ISTOP=1) 46,58,58

CALCULATE EXACT POSTERIOR MEAN

46 CALL EPM

FOR INCOMPLETE DATA Z, CALCULATE MAXIMUM-LIKELIHOOD ESTIMATE,
POSTERIOR MODE, AND TAYLOR-SERIES APPROXIMATED POSTERIOR MEAN
AND TAYLOR-SERIES APPROXIMATED POSTERIOR COVARIANCE MATRICES

CALL METHODS
IF (ISTOP=1) 49,58,58

AVERAGE ESTIMATORS

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

TO CALCULATE S.E. OF AVERAGE ESTIMATORS (WANT S.E. SMALL RELATIVE
TO DIFFERENCE BETWEEN AVERAGE BIAS)

XAPMLE1=XAPMLE1+PMLE1
XAPMLE2=XAPMLE2+PMLE2
XAPEPM1=XAPEPM1+PEPM1
XAPEPM2=XAPEPM2+PEPM2
XAPAPM1=XAPAPM1+PAPM1
XAPAPM2=XAPAPM2+PAPM2
XAPPMD1=XAPPMD1+PPMD1+PPMD2
Differences of estimators from exact posterior mean

Bias of T.S. approx. posterior mean from exact posterior mean

Mean square error of T.S. APM from exact posterior mean

For variance of mean square error

Bias of posterior mode from exact posterior mean

Mean square error of posterior mode from exact posterior mean

For variance of mean square error

Bias of MLE from exact posterior mean (incomplete data)

Mean square error of IC-D MLE from exact posterior mean
FOR VARIANCE OF MSE

DIFFERENCE OF APPROXIMATED POSTERIOR COVARIANCES FROM EXACT POSTERIOR COVARIANCES

IF (COVSKIP=0) 51,51,50
50 NCOV=NCOV-1
COVSKIP=0
GO TO 53

CUMULATE VALUES TO AVERAGE EXACT POSTERIOR VARIANCES

51 AEPHC11=AEPHC11+EPMC11
AEPHC12=AEPHC12+EPMC12
AEPHC22=AEPHC22+EPMC22

CUMULATE VALUES TO AVERAGE APPROXIMATE POSTERIOR VARIANCES

AAPMC11=AAPMC11+APMC11
AAPMC12= AAPMC12+APMC12
AAPMC22= AAPMC22+APMC22

FOR S.E. OF COVARIANCE AVERAGES

A2EPC11=A2EPC11+EPMC11*EPMC11
A2EPC12=A2EPC12+EPMC12*EPMC12
A2EPC22=A2EPC22+EPMC22*EPMC22
A2APC11=A2APC11+APMC11*APMC11
A2APC12=A2APC12+APMC12*APMC12
A2APC22=A2APC22+APMC22*APMC22

CUMULATE DIFFERENCES BETWEEN EXACT AND APPROXIMATE POSTERIOR VARIANCES

BAPMC11=BAPMC11+EAPMC11
BAPMC12=BAPMC12+EAPMC12
BAPMC22=BAPMC22+EAPMC22

FOR AVERAGE PERCENT RELATIVE DIFFERENCE

PRDC11=ABS(EAPMC11)/EPMC11
PRDC12=ABS(EAPMC12/EPMC12)
PRDC22 = ABS(EAPMC22) / EPMC22
APRC11 = APRDC11 + PROC11
APRC12 = APRDC12 + PROC12
APRC22 = APRDC22 + PROC22

FOR MSE OF ELEMENTS OF APPROXIMATED POSTERIOR COVARIANCE MATRIX

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

FOR VARIANCE OF MSE

VC11MSE = VC11MSE + EAP2C11 * EAP2C11
VC12MSE = VC12MSE + EAP2C12 * EAP2C12
VC22MSE = VC22MSE + EAP2C22 * EAP2C22
VARMSE = VARMSE + VVV

ADD TO BIAS-SIGN AND %-RELATIVE-DIFFERENCE COUNTS

CALL COUNTS(EAPMC11, PROC11, 1)
CALL COUNTS(EAPMC12, PROC12, 2)
CALL COUNTS(EAPMC22, PROC22, 3)

ALSO CHECK DIRECTION OF BIAS OF DEPENDENT COVARIANCES

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
PROC13 = ABS(BC13 / EC13)
PRDC23 = ABS(BC23 / EC23)
PRDC33 = ABS(BC33 / EC33)
CALL COUNTS(BC13, PROC13, 6)
CALL COUNTS(BC23, PRDC23, 7)
CALL COUNTS(BC33,PROC33,8)

DIFFERENCE OF ESTIMATORS FROM GENERATED P

\[ YMLC1 - YMLC1 + DMLC1 \]
\[ YMLC2 - YMLC2 + DMLC2 \]
\[ YYMLCD = DMLC1 \ast DMLC1 + DMLC2 \ast DMLC2 + DMLC3 \ast DMLC3 \]
\[ YMLC20 = YMLC2 + YYMLCD \]
\[ SQMLCD = YYMLCD \ast YYMLCD \]
\[ REGMLCO = REGMLCO + SQMLCD \]

NOTE THAT FOR REMAINING COMPARISONS WITH GENERATED P WE ARE MAINLY INTERESTED IN MSE BECAUSE MAIN CONCERN IS DETERMINING WHICH ESTIMATOR BEST MINIMIZES QUADRATIC LOSS. HOWEVER, WE WILL ALSO CALCULATE BIAS FROM GENERATED (OR GIVEN) P.

BIAS OF EXACT POSTERIOR MEAN FROM GENERATED P

\[ YEPMN1 = YEPMN1 + DEPMN1 \]
\[ YEPMN2 = YEPMN2 + DEPMN2 \]
\[ YYEPMN = DEPMN1 \ast DEPMN1 + DEPMN2 \ast DEPMN2 + DEPMN3 \ast DEPMN3 \]

FOR USUAL MEAN SQUARE ERROR OF EPM FROM GENERATED P

\[ YEPMSQ = YEPMN1 \ast YYEPMN \]

FOR REGRESSION ESTIMATION AND CONTROL VARIATE MSE EPM

\[ REGEMNO = REGEMMO + YYEPMN + YYMLCD \]

FOR VARIANCE OF ALL 3 MEAN SQUARE ERRORS

\[ YEPMN4 = YEPMN4 + YYEPMN \]

BIAS OF APPROX POSTERIOR MEAN FROM GENERATED P

\[ YAPMN1 = YAPMN1 + DAPMN1 \]
\[ YAPMN2 = YAPMN2 + DAPMN2 \]
\[ YYAPMN = DAPMN1 \ast DAPMN1 + DAPMN2 \ast DAPMN2 + DAPMN3 \ast DAPMN3 \]

USUAL MEAN SQUARE ERROR FOR APMN FROM GENERATED P

\[ YAPMSQ = YAPMSQ + YYAPMN \]
FOR REGRESSION ESTIMATION AND CONTROL VARIATE MSE

\[ \text{REGAPM0} = \text{REGAPM0} + YYAPM1 \times YYM1CD \]

FOR VARIANCE OF MEAN SQUARE ERRORS

\[ YYAPM1 = YYAPM1 + YYAPM1 \times YYAPM1 \]

FOR BIAS OF POSTERIOR MODE FROM GENERATED P

\[ YPMD1 = YPMD1 + DPM1 \]
\[ YPMD2 = YPMD2 + DPM2 \]
\[ YPPM1 = YPPM1 \times DPM1 \times DPM2 \times DPM3 \times DPM3 \]
\[ YPPM0 = YPPM0 + YPPM1 \times YPPM1 \]
\[ YPMD0 = YPMD0 \times YPPM0 \times YYM1CD \]
\[ YPMD4 = YPMD4 \times YPPM0 \times YYM1CD \]

BIAS OF MLE FROM GENERATED P

\[ YML1 = YML1 + DML1 \]
\[ YML2 = YML2 + DML2 \]
\[ YYML = YYML = DML1 \times DML2 \times DML2 \times DML3 \]
\[ YYM0 = YYM0 + YYML \times YYML \]
\[ YYML0 = YYML0 \times YYML \times YYM1CD \]
\[ YYML4 = YYML4 \times YYML \times YYM1CD \]

BY TWO DIFF CRITERION CALC BEST EST FOR THIS TRINOMIAL TRIAL

CALL BESTEST(BESTOL(1))

FOR S.E. OF AVERAGE BIAS

\[ WAPM1 = WAPM1 + EAPM1 \times EAPM1 \]
\[ WAPM2 = WAPM2 + EAPM2 \times EAPM2 \]
\[ WPM1 = WPM1 + EPM1 \times EPM1 \]
\[ WPM2 = WPM2 + EPM2 \times EPM2 \]
\[ WML1 = WML1 \times EML1 \times EML1 \]
\[ WML2 = WML2 \times EML2 \times EML2 \]
\[ UM11 = UM11 \times DML1 \times DML1 \times DML1 \]
\[ UM12 = UM12 \times DML2 \times DML2 \times DML2 \]
\[ UEPM1 = UEPM1 + DEPM1 \times DEPM1 \]
\[ UEPM2 = UEPM2 + DEPM2 \times DEPM2 \]
\[ UAPM1 = UAPM1 + DAAPM1 \times DAAPM1 \]
\[ UAPM2 = UAPM2 + DAAPM2 \times DAAPM2 \]
UPMD1=UPMD1+DPMD1+DPMD1
UPMD2=UPMD2+DPMD2+DPMD2
UML1=UML1+DML1+DML1
UML2=UML2+DML2+DML2

CALCULATE ROBUSTNESS ESTIMATORS (FOR QUADRATIC-LOSS COMPARISON ONLY). USE WRONG PRIOR IN ESTIMATES.
EXCLUDE EPM BECAUSE OF EXPENSE.

ROBUSTNESS SET 1. UNIFORM PRIOR. RECULATE ONLY APPROXIMATE POSTERIOR MEAN BECAUSE FOR A UNIFORM PRIOR THE POSTERIOR MODE WILL EQUAL THE ALREADY CALCULATED M.L.E. (FOR INCOMPLETE DATA)

Z1N=Z1+1.
Z2N=Z2+1.
Z3N=Z3+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
  APVMN1=VAPMN1+PAPM1
  APVMN2=VAPMN2+PAPM2
  FAPMN1=FAPMN1+DAPMN1*PAPM1
  FAPMN2=FAPMN2+DAPMN2*PAPM2
  AFPMN1=AFPMN1+DAPMN1*PAPM1
  AFPMN2=AFPMN2+DAPMN2*PAPM2
  VAPMSQ=VAPMSQ+VVAPMN
  REGAPM1=REGAPM1+VVAPMN*YYMLCD
  VAPMN4=VAPMN4+VVAPMN*VVAPMN
  YMLC21=YMLC21+YYMLCD
  REGMLC1=REGMLC1+SOHMLCD

BY TWO DIFF CRITERION CALC BEST EST FOR THIS TRINOMIAL TRIAL
CALL BESTEST(BESTEST(9))

ROBUSTNESS SET 2. IN BAYESIAN ESTIMATORS APMN AND PMD, USE PRIOR PARAMETERS 10. (NU/10.+{0.09, 0.05, -0.14})

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
QAPMN1=QAPMN1+DAPMN1
QAPMN2=QAPMN2+DAPMN2
AQPM1=AQPM1+PAPM1
AQPM2=AQPM2+PAPM2
QAPMN=QAPMN1*DAPMN1+DAPMN2*DAPMN2+DAPMN3*DAPMN3
QAPMSO=QAPMSO+QOAPMN
REGAPM2=REGAPM2+QQAPMN*YYMLCO
QAPMN4=QAPMN4+QQAPMN*QQAPMN
YMLC22=YMLC22+YYMLCO
REGMLC2=REGMLC2+SQMLCD

C
QPMD1=QPMD1+DPMD1
QPMD2=QPMD2+DPMD2
AQPMD1=QPMD1+PPMD1
AQPMD2=QPMD2+PPMD2
QPMD=QPMD1+QPMD2+DPMD2*DPMD2*DPMD3*DPMD3
QPMDSO=QPMDSO+QQPMD
REGPMD2=REGPMD2+QQPMD*YYMLCO
QPMD4=QPMD4+QQPMD*QQPMD

C
GAPMN1=GAPMN1+DAPMN1+DAPMN1
GAPMN2=GAPMN2+DAPMN2+DAPMN2
GPM11=GPM11+DPM11+DPM11
GPM21=GPM21+DPM21+DPM21
AGPM1=AGPM1+PAPM1+PAPM1
AGPM2=AGPM2+PAPM2+PAPM2
PGMD1=PGMD1+PPMD1+PPMD1
PGMD2=PGMD2+PPMD2+PPMD2

C
BY TWO DIFF CRITERION CALC BEST EST FOR THIS TRINOMIAL TRIAL
CALL BESTEST(BESTQL(17))

C
GO TO 65
58 ISTOP=0
IF (IROBUST-1) 60,61,62
NUMXZ=NUMXZ-1
NCOV=NCOV-1
NXZR1=NXZR1-1
NXZR2=NXZR2-1
GO TO 65
61 NXZR1=NXZR1-1
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 ESTIMATOR MEANS (ESTIMATORS AVERAGED OVER NUMXZ TRIALS)

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
AQPMO1=AQPMO1/NXZR2
AQPMO2=AQPMO2/NXZR2

C ND=NUMXZ*(NUMXZ-1)
NC=NCOV*(NCOV-1)
NDR1=NXZR1*(NXZR1-1)
NDR2=NXZR2*(NXZR2-1)

C STANDARD ERRORS OF ESTIMATOR MEANS

SE1=SORT((XAPMLE1-NUMXZ*APMLE1*APMLE1)/ND)
SE2=SORT((XAPMLE2-NUMXZ*APMLE2*APMLE2)/ND)
SE3=SORT((XAPEPM1-NUMXZ*APEPM1*APEPM1)/ND)
SE4=SORT((XAPEPM2-NUMXZ*APEPM2*APEPM2)/ND)
SE5=SORT((XAPAPM1-NUMXZ*APAPM1*APAPM1)/ND)
SE6=SORT((XAPAPM2-NUMXZ*APAPM2*APAPM2)/ND)
SE7=SORT((XAPPMD1-NUMXZ*APPMD1*APPMD1)/ND)
SE8=SORT((XAPPMD2-NUMXZ*APPMD2*APPMD2)/ND)
SE9=SORT((XAPMLC1-NUMXZ*APl-APl)/ND)
SE10=SORT((XAPMLC2-NUMXZ*APl-APl)/ND)
SE11=SORT((XAP2-NUMXZ*AP2-AP2)/ND)
SE12=SORT((XAP2-NUMXZ*AP2-AP2)/ND)
SE13=SORT((AFPMN1-NXZ1*AVPMN1/AVPMN1)/NDR1)
SE14=SORT((AFPMN2-NXZ1*AVPMN2/AVPMN2)/NDR1)
SE15=SORT((APGMN1-NXZ2*APGMN1/AGPMN1)/NDR2)
SE16=SORT((APGMN2-NXZ2*APGMN2/AGPMN2)/NDR2)
SE17=SQRT((PGMD1-NXZ2*APMD1/AGPMN1)/NDR2)
SE18=SQRT((PGMD2-NXZ2*APMD2/AGPMN2)/NDR2)
PRINT 1079, NUMXZ
PRINT 1080, APMLE1, SE1, APEPM1, SE3, APAPM1, SE5, APMD1, SE7, APMLC1, SE9
AP1, SE11, APMLE2, SE2, APEPM2, SE4, APAPM2, SE6, APMD2, SE8, APMLC2, SE10,
2AP2, SE12
1079 FORMAT(10X, *AVERAGE ESTIMATORS (MEANS) AND THEIR STANDARD ERRORS OR
1ER*, I4, *TRIALS*)
2PI 12F11.7/* P2 12F11.7/)

C AVERAGE BIASES AND COVARIANCE ESTIMATORS
C
XAPMN1=XAPMN1/NUMXZ
XAPMN2=XAPMN2/NUMXZ
XPM1=XPM1/NUMXZ
XPM2=XPM2/NUMXZ
XML1=XML1/NUMXZ
XML2=XML2/NUMXZ
AAPMC11=AAPMC11/NCOV
AAPMC12=AAPMC12/NCOV
AAPMC21=AAPMC21/NCOV
AAPMC22=AAPMC22/NCOV
BAPMC11=BAPMC11/NCOV
BAPMC12=BAPMC12/NCOV
BAPMC21=BAPMC21/NCOV
BAPMC22=BAPMC22/NCOV
YMLC1=YMLC1/NUMXZ
YMLC2=YMLC2/NUMXZ
YEPMN1=YEPMN1/NUMXZ
YEPMN2=YEPMN2/NUMXZ
YAPMN1=YAPMN1/NUMXZ
YAPMN2=YAPMN2/NUMXZ
YPM1=YPM1/NUMXZ
YPM2=YPM2/NUMXZ
YML1-YML1/NUMXZ
YHL2-YHL2/NUMXZ
VAPMN1=VAPMN1/NXZR1
VAPMN2=VAPMN2/NXZR1
QAPMN1=QAPMN1/NXZR1
QAPMN2=QAPMN2/NXZR2
QPM1=QPM2/NXZR2
QPM2=QPM2/NXZR2
EBIAS1(NREPLIC, ISS, IPID) = XAPMN1
EBIAS2(NREPLIC, ISS, IPID) = XPM01
EBIAS3(NREPLIC, ISS, IPID) = XML1

CALCULATE S.E. OF BIAS. WANT THIS TO BE SMALL RELATIVE TO THE DIFFERENCE BETWEEN THE BIASES

SE21 = SQRT((WAPMN1-NUMXZ*YAPMN1*YAPMN1)/ND)
SE22 = SQRT((WAPMN2-NUMXZ*YAPMN2*YAPMN2)/ND)
SE23 = SQRT((WPM1-NUMXZ*YPM01*YPM01)/ND)
SE24 = SQRT((WPM2-NUMXZ*YPM02*YPM02)/ND)
SE25 = SQRT((WML1-NUMXZ*YML1*YML1)/ND)
SE26 = SQRT((WML2-NUMXZ*YML2*YML2)/ND)
SE27 = SQRT((A2EPC11-NCOV*AEMC11*AEMC11)/NC)
SE28 = SQRT((A2EPC12-NCOV*AEMC12*AEMC12)/NC)
SE29 = SQRT((A2EPC22-NCOV*AEMC22*AEMC22)/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*YEMPNI*YEMPNI)/ND)
SE39 = SQRT((UEPMN2-NUMXZ*YEMPNI*YEMPNI)/ND)
SE40 = SQRT((UAPMN1-NUMXZ*YAPMN1*YAPMN1)/ND)
SE41 = SQRT((UAPMN2-NUMXZ*YAPMN2*YAPMN2)/ND)
SE42 = SQRT((UPM1-NUMXZ*YPM01*YPM01)/ND)
SE43 = SQRT((UPM2-NUMXZ*YPM02*YPM02)/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((QPM1-NXZR2*QPM2*QPM2)/NDR2)
SE51=SQR((GPMD-NXZR2*OPMD2*QPM2)/NDR2)

PRINT 1150, NUMXZ, XAPMN1, XPM1, XAPMN2, XPM2, XML1, XML2, XML3, XAPMN1, XPM1, XAPMN2, XPM2, XML1, XML2, XML3

1150 FORMAT(//* AVERAGE BIASES AND THEIR STANDARD ERRORS OVER TRIALS */14X* TRI 1ALS// FROM EPM1*XAPM1*XPM1*S.E.16X* XML1*XPM1*S.E.17X* XML1*S.E.10X,P1 *4(F12.8,F18.14)*/ P2 *4(F12.8,F18.14)//)

PRINT 1154, AVPMN1, SE13, APMLE1, SE1, AQPNMN1, SE15, AQPMN2, SE17, AVPMN2, SE19

1154 FORMAT(// ROBUST-ESTIMATOR AVERAGES AND S.E. */10X* S.E. 1RELATIVE TO DIFFERENCE BETWEEN ESTIMATORS. */9X*APRM1*11X*S.E.*9X*S.E.2MD1*ML1*9X*S.E.*11X*APMR2*10X*S.E.*11X*PMDR2*10X*S.E.*11X*QPM1*14(F12.8,F18.14)//)

PRINT 1155, VAPMN1, SE46, YML1, SE44, VAPMN2, SE48, QPMN1, SE50, QPMN2, SE52

1155 FORMAT(// ROBUST-ESTIMATORS BIASES AND STANDARD ERRORS. */10X* 1VAP1*11X*S.E.*9X*VPM1*9X*S.E.*11X*QPM1*11X*S.E.*11X*QPM1

21X*S.E.*10X* P1 *4(F12.8,F18.14)*/ P2 *4(F12.8,F18.14)//)

C MEAN SQUARE ERRORS AND THEIR STANDARD ERRORS

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=SQR((XAPM4-NUMXZ*XAPMNSQ*XAPMNSQ)/ND)
SD2=SQR((XPM4-NUMXZ*XPMDSQ*XPMDSQ)/ND)
SD3=SQR((XML4-NUMXZ*XMLSQ*XMLSQ)/ND)
SD4=SQR((REGNLCO-NUMXZ*YMLC20*YMLC20)/ND)
SD5=SQR((YEPMN4-NUMXZ*YEPMSQ*YEPMSQ)/ND)
SD6=SQR((YAPMN4-NUMXZ*YAPMSQ*YAPMSQ)/ND)
SD7=SQR((YPMD4-NUMXZ*YPMDSQ*YPMDSQ)/ND)
SD8=SQR((YML4-NUMXZ*YMLSQ*YMLSQ)/ND)

C DIFFERENCES BETWEEN MEAN SQUARE ERRORS
DM1-XAPMNSO-XPMDSQ
DM2-XAPMNSO-XAPMNSO
DM3-XPMDSQ-XMLSQ
DM4=YMLC20-YEPMSO
DM5=YMLC20-YAPMNSO
DM6=YMLC20-YPMDSO
DM7=YMLC20-YMLSO
DM8=YEPMSQ-YAPMNSO
DM9=YEPMSQ-YPMDSO
DM10=YEPMSQ-YMLSQ
DM11=YAPMNSQ-YPMDSO
DM12=YAPMNSQ-YMLSQ
DM13=YPMDSQ-YMLSQ
PRINT 1200, NUMXZ, XAPMNSQ, SD1, YEPMSQ, SD2, YAP
1NSQ, SD6, YMLSQ, SD8, XMLSQ, SD9, YPMDSQ, SD7
1200 FORMAT(//, 'AVERAGE MEAN SQUARE ERRORS AND THEIR STANDARD ERRORS OV
1ER TRIALS',// 17X, 'MSE*12X*S.E.*25X*MSE*12X*S.E.*25X*MSE*12X*S.E
3F18.13/2X*PMO-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*PMO-EPM*F14.9*F18.13/)
PRINT 1300, DM1, DM3, DM4, DM6, DM9, DM12, DM6, DM3, DM10, DM13, DM7
1300 FORMAT(//, 'DIFFERENCE BETWEEN AVERAGE MSE'S. WANT DIFFERENCE LARG
1E RELATIVE TO SE(MSE)',// 17X, 'EAPM-EPM*F14.9,12X*DEPM-DEPM*F14.9,12X
2*DEPM-DEPM*F14.9,12X*DMLC-DEPM*F14.9/* EAPM-EMLE*F14.9,12X*DEPM-DE
3PM*F14.9,12X*DMLE-DEPM*F14.9,12X*DMLE-DEPM*F14.9,12X*DMLE-DEPM*F14
4.9,12X*DEPM-DEPM*F14.9,12X*DMLC-DEPM*F14.9,12X*DMLE-DEPM*F14.9)
XN=NUMXZ
DON=7I=1,6
IF (I=4) 69,67,68
67 XN=NXZR1
68 XN=NXZR2
69 AVNUMIT(I)=AVNUMIT(I)/XN
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=*212/
2 NUMXZ=*I3, NXZR1=*I3, NXZR2=*I3, AV NUM ITER FOR MLE=*F7
43//)
PRINT 2100, ((CTNUMIT(I,J), J=1,10), I=1,4), ((CTNUMIT(I,J), J=1,10), I=2,5),((CTNUMIT(I,J), J=1,10), I=3,6)
2010 FORMAT(* PROPORTION OF DATA SETS FOR WHICH NUMBER OF ITERATIONS WAS
OF SPECIFIED AMOUNTS/10*1  2  3  4  5  6  7  8
2  8-10  11-15 GT 15*/10X*1  2  3  4  5  6  7  8
3-10  11-15 GT 15/*/10X*1  2  3  4  5  6  7  8
4  PM2R2 *10F6.3* APMRO *10F6.3* APMR2 *10F6.3/**
YEPMSQ*NUMXZ*YEPMSQ
YAPMSQ*NUMXZ*YAPMSQ
YPMDSO*NUMXZ*YPMDSO
YMLSQ*NUMXZ*YMLSQ
YMLC20*NUMXZ*YMLC20
T=(1-(P1+P2+P3*P3))/SS
C
CALL ESTMSE(YEPMSQ,YEPMN4,REGEPMO,YMLC20,REGMLC0,T,NUMXZ,MSE(1))
CALL ESTMSE(YAPMSQ,YAPMN4,REGAPMO,YMLC21,REGMLC1,T,NUMXZ,MSE(1))
CALL ESTMSE(YPMDSO,YPM4,REGPMDO,YMLC22,REGMLC2,T,NUMXZ,MSE(1))
CALL ESTMSE(YMLSQ,YML4,REGMLQ,YMLC23,REGMLC3,T,NUMXZ,MSE(1))
CALL ESTMSE(YMLC20,YMLC20,REGMLC4,T,NUMXZ,MSE(1))
CALL ESTMSE(YMLC20,YMLC20,REGMLC5,T,NUMXZ,MSE(1))
CALL ESTMSE(YMLC20,YMLC20,REGMLC6,T,NUMXZ,MSE(1))
CALL ESTMSE(YMLC20,YMLC20,REGMLC7,T,NUMXZ,MSE(1))
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
ISS COMPARISONS. IP=I2, PID=F4.2, SS=F3.0, NREPLIC=I2*/18X*
2EPN*15X*APM*15X*PMO*15X*MLE*14X*APMR1*13X*APMR2*13X*PMO2/** 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, 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
C

2045 CTSEQOL(1) = NUMXZ
    CTSEQOL(2) = NXZR1
    CTSEQOL(3) = NXZR2
    DO 75 I = 1, 4
    DO 72 IR = 1, 3
        BESTQL(I, 1, IR) = BESTQL(I, 1, IR) / CTSEQOL(IR)
        BESTQL(I, 2, IR) = BESTQL(I, 2, IR) / CTSEQOL(IR)
        SBIASQL(IR, I) = SBIASQL(IR, I) / NUMXZ
    72 CONTINUE
    IF (IR = 1) 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
    76 CONTINUE

C

CALCULATE % ABS REL DIFF LESS THAN (INSTEAD OF BETWEEN) SPECIFIED AMOUNTS

C

DO 80 I = 1, 7
    DO 79 II = 2, 8
        PRDQL(II, I) = PRDQL(II, I) + PRDQL(II-1, I)
        IF (II = 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
    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
2050 FORMAT(* PROPORTION OF CASES THAT AN ESTIMATOR IS BEST. FIRST 3 COLUMNS ARE RESULTS FOR ESTIMATING EPM. REMAINING COLUMNS ARE FOR MIN QUALITY CRITERIA. PRIOR ROBUST SET = UNIFORM PRIOR ROBUST SET */2D PMD APM*11X, *MLE PMD APM*14X*PMLE PMD APM*17X*PMLE PMD APM*/* SQUARED ERROR CRIT */3F6.2,8X3F6.2,3X,2(9X,3F6.2)/* RELATIVE DIFF CRIT */3F6.2,8X,3F6.2,3X,2(9X,3F6.2)/*)

2070 FORMAT(* PROPORTION OF CASES IN WHICH DIFFERENCE BETWEEN FIRST COMPONENT OF ESTIMATOR AND THAT OF ESTIMATED IS OF A CERTAIN SIGN*/15X*EMLE*6X*EPMID*6X*EAPM*16X*QEM*6X*QMLE*5X*QMDR*4X*QAPM*9X*QA*/3F6.2,8X,3F6.2,3X,2(9X,3F6.2)/* POS */3F10.4,10X,4F10.4,5X,2F10.4/* NEG */3F10.4,10X,4F10.4,5X,2F10.4/*)

2080 FORMAT(* PROPORTION OF CASES FOR WHICH THE ABSOLUTE RELATIVE DIFFERENCE FOR EACH OF THE THREE ESTIMATOR COMPONENTS IS LESS THAN SPECIFIED AMOUNTS*/15X*EMLE*6X*EPMID*6X*EAPM*16X*QEM*6X*QMLE*5X*QMDR*4X*QAPM*9X*QA*/3F6.2,8X,3F6.2,3X,2(9X,3F6.2)/* POS */5X*F10.4,5X,2F10.4/* NEG */3F10.4,10X,4F10.4,5X,2F10.4/*)

C PERCENT AVERAGE RELATIVE DIFFERENCE FOR COVARIANCE ESTIMATES

PARDC11=100.*((AEPMC11-AAPMC11)/AEPMC11)
PARDC12=100.*((AEPMC12-AAPMC12)/AEPMC12)
PARDC22=100.*((AEPMC22-AAPMC22)/AEPMC22)

C AVERAGE PERCENT RELATIVE DIFFERENCE

APRDC11=100.*APRDC11/NCOV
APRDC12=100.*APRDC12/NCOV
APRDC22=100.*APRDC22/NCOV

C SQUARE ROOT MSE DIVIDED BY AVERAGE EPV

C11MSE=C11MSE/NCOV
C12MSE = C12MSE / NCOV
C22MSE = C22MSE / NCOV
C11RT1 = SQRT(C11MSE) / AEPMC11
C12RT1 = SQRT(C12MSE) / AEPMC12
C22RT1 = SQRT(C22MSE) / AEPMC22

C

C SE1MSE1 / MSE
C
SE11MSE = SORT((VC11MSE - NCOV*C11MSE*C11MSE) / NC)
SE12MSE = SORT((VC12MSE - NCOV*C12MSE*C12MSE) / NC)
SE22MSE = SORT((VC22MSE - NCOV*C22MSE*C22MSE) / NC)
C11RT2 = SE11MSE / NCOV
C12RT2 = SE12MSE / NCOV
C22RT2 = SE22MSE / NCOV
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, C11RT2,
AEPMC12, SE28, AAPMC12, SE31, PARDC12, APRDC12, C12RT1, C12RT2, AEPMC22,
SE29, 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 SORT(MSE) / (1) SE(MSE) / MSE /// 3X * C11
2 * 8E16.7/3X * C12 * 8E16.7/3X * C22 * 8E16.7)
PRINT 3005, NCOV, IP, PID, SS, NREPLI, (COUNTRD(I, 1), I = 1, 6), (COUNTRD(I,
1), 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(/// PROPORION OF 14 CASES IN WHICH PERCENT REL DIFF WAS
1 LESS THAN VARYING AMOUNTS. IP = 12 PID = 42 SS = 4.0 NREPLI
2C = 12/8X * 0.1 0.1 1.0 5.0 10.0 15.0 12X * 0.1 0.1 1.0 35.0 10.0 15.0 12X * 0.1
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(/// PROPORION OF 14 CASES IN WHICH BIAS IS OF A CERTAIN SIGN
1//7X*NEG ZER 0 POS * 7X*NEG ZER 0 POS * 7X*NEG ZER 0 POS * 7X*NEG
2 ZER 0 POS * 7X*NEG ZER 0 POS * 7X*NEG ZER 0 POS /* C11 * 3F6.3 * C2
32 * 3F6.3 * C33 * 3F6.3 * C12 * 3F6.3 * C13 * 3F6.3 * C23 * 3F6.3)
AVERAGE TRUE PERCENT (PROPORTION) INCOMPLETE DATA

AVTPID = AVTPID + PID
AVDPID = AVDPID + PID
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//////)

95 CONTINUE

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
ISS = 2
GO TO 1

9904 PID = 0.40
IPID = 2
ISS = 2
GO TO 1

9905 CONTINUE

WRITE(12,8000) TLABEL(1), (ALABEL(I), I=1,3)

8000 FORMAT(63X,10///7X*REGRESSION-ESTIMATE MSE DATA OVER 200 TRINOMIA
1L SIMULATIONS. TWO REPLICATIONS PER CELL. DESIGN 2. *2A10,3/*)
WRITE(12,8001)

8001 FORMAT(47X*A. ORIGINAL PRIOR IN BAYESIAN ESTIMATORS.)*/*/*
WRITE(12,8010)

8010 FORMAT(12X,27H*************** , *SS=25*,27H ************
1*****************,X,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
6ION2 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(* APHR0 *12,4(2E15.6,2X)/(9X,4(2E15.6,2X)))
WRITE(12, 8012) (I, (QLMS21(I, J, K), K = 1, 2), J = 1, 2), (QLMS22(I, J, K), K = 1, 2), J = 1, 2), I = 1, 10)
8012 FORMAT(* PMDR *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, 2), J = 1, 2), I = 1, 10)
8013 FORMAT(* MLE *I2, *4(2E15,6,2X)/9X, 4(2E15,6,2X))
WRITE(12, 8000) TLABEL(1), (ALABEL(I), I = 1, 3)
WRITE(12, 8020)
8020 FORMAT(* MDR *I2, *4(2E15,6,2X)/9X, 4(2E15,6,2X))
WRITE(12, 8010) (I, (QLMS41(I, J, K), K = 1, 2), J = 1, 2), (QLMS42(I, J, K), K = 1, 2), J = 1, 2), I = 1, 10)
8021 FORMAT(* MPK *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, 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, 2), J = 1, 2), I = 1, 10)
8023 FORMAT(* MDR *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.
SOSM = 0.
DO 9911 I = 1, 10
SUM = SUM + QLMS11(I, IPID, NREPLIC)
SOSM = SOSM + 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(IPID, NREPLIC, 3), T11(IPID, NREPLIC, 4), T11(IPID, NREPLIC, 5), T11(IPID, NREPLIC, 6) = SUM/10.
T11(IPID, NREPLIC, 7) = SQRT((SOSM - 10. * T11(IPID, NREPLIC, 6) * T11(IPID, NREPLIC, 6))/90.)
T11(IPID, NREPLIC, 8) = T11(IPID, NREPLIC, 3)
9910 CONTINUE
C
SUM = 0.
SOSM=0.
DO 9912 I=1,10
SUM = SUM+QLMS12(I,IPID,NREPLIC)
SOSM=SOSM+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(IPID,NREPLIC,3),T12(IPID,NREPLIC,4),T12(IPID,NREPLIC,5))
T12(IPID,NREPLIC,6)=SUM/10.
T12(IPID,NREPLIC,7)=SORT((SOSM-10.*T12(IPID,NREPLIC,6)*T12(IPID,NREPLIC,7))
T12(IPID,NREPLIC,8)=T12(IPID,NREPLIC,3)
T12(IPID,NREPLIC,9)=T12(IPID,NREPLIC,2)+2.*T12(IPID,NREPLIC,3)+T12(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.
SOSM=0.
DO 9913 I=1,10
SUM = SUM+QLMS21(I,IPID,NREPLIC)
SOSM=SOSM+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(IPID,NREPLIC,3),T21(IPID,NREPLIC,4),T21(IPID,NREPLIC,5))
T21(IPID,NREPLIC,6)=SUM/10.
T21(IPID,NREPLIC,7)=SORT((SOSM-10.*T21(IPID,NREPLIC,6)*T21(IPID,NREPLIC,7))
T21(IPID,NREPLIC,8)=T21(IPID,NREPLIC,3)
T21(IPID,NREPLIC,9)=T21(IPID,NREPLIC,2)+2.*T21(IPID,NREPLIC,3)+T21(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.
SOSM=0.
DO 9914 I=1,10
SUM = SUM+QLMS22(I,IPID,NREPLIC)
SOSM=SOSM+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(IPID,NREPLIC,3),T22(IPID,NREPLIC,4),T22(IPID,NREPLIC,5))
T22(IPID,NREPLIC,6)=SUM/10.
T22(IPID,NREPLIC,7)=SORT((SOSM-10.*T22(IPID,NREPLIC,6)*T22(IPID,NREPLIC,7))
T22(IPID,NREPLIC,8)=T22(IPID,NREPLIC,3)
T22(IPID,NREPLIC,9)=T22(IPID,NREPLIC,2)+2.*T22(IPID,NREPLIC,3)+T22(IPID,NREPLIC,4)/4.
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(IPID,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,NREPLIC,6))/90.)
T31(IPID,NREPLIC,8) = T31(IPID,NREPLIC,3)
T31(IPID,NREPLIC,9) = (T31(IPID,NREPLIC,2) + 2. * T31(IPID,NREPLIC,3) * T31(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(IPID,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,NREPLIC,6))/90.)
T32(IPID,NREPLIC,8) = T32(IPID,NREPLIC,3)
T32(IPID,NREPLIC,9) = (T32(IPID,NREPLIC,2) + 2. * T32(IPID,NREPLIC,3) * T32(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(IPID,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,NREPLIC,6))/90.)
T41(IPID,NREPLIC,6)=T41(IPID,NREPLIC,3)
T41(IPID,NREPLIC,9)=(T41(IPID,NREPLIC,2)+2)*T41(IPID,NREPLIC,3)+T4
T11(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.
SOSH=0.
DO 9918 I=1,10
SUM = SUM+QLMS42(I,IPID,NREPLIC)
SOSH=SOSH+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
(IPID,NREPLIC,3),T42(IPID,NREPLIC,4),T42(IPID,NREPLIC,5))
T42(IPID,NREPLIC,6)=SUM/10.
T42(IPID,NREPLIC,7)=SQRT((SOSH-10.*T42(IPID,NREPLIC,6)*T42(IPID,NR
EPLIC,6))/90.)
T42(IPID,NREPLIC,8)=T42(IPID,NREPLIC,3)
T42(IPID,NREPLIC,9)=(T42(IPID,NREPLIC,2)+2)*T42(IPID,NREPLIC,3)+T4
T12(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.
SOSH=0.
DO 9919 I=1,10
SUM = SUM+QLMS51(I,IPID,NREPLIC)
SOSH=SOSH+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
(IPID,NREPLIC,3),T51(IPID,NREPLIC,4),T51(IPID,NREPLIC,5))
T51(IPID,NREPLIC,6)=SUM/10.
T51(IPID,NREPLIC,7)=SQRT((SOSH-10.*T51(IPID,NREPLIC,6)*T51(IPID,NR
EPLIC,6))/90.)
T51(IPID,NREPLIC,8)=T51(IPID,NREPLIC,3)
T51(IPID,NREPLIC,9)=(T51(IPID,NREPLIC,2)+2)*T51(IPID,NREPLIC,3)+T5
T11(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.
SOSH=0.
DO 9920 I=1,10
SUM = SUM+QLMS52(I,IPID,NREPLIC)
SOSH=SOSH+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
(IPID,NREPLIC,3),T52(IPID,NREPLIC,4),T52(IPID,NREPLIC,5))
DO 9921 I=1,10
   SUM = SUM*QLSM61(I,IPID,NREPLIC)
   SOSM=SOSM*QLSM61(I,IPID,NREPLIC)*QLMS61(I,IPID,NREPLIC)
   CALL SUMMARY(TUKEY(I),T61(IPID,NREPLIC,1),T61(IPID,NREPLIC,2),T61(IPID,NREPLIC,3),T61(IPID,NREPLIC,4),T61(IPID,NREPLIC,5))
   T61(IPID,NREPLIC,6)=SUM/10.
   T61(IPID,NREPLIC,7)=SQRT((SOSM-10.*T61(IPID,NREPLIC,6)*T61(IPID,NREPLIC,6))/90.)
   T61(IPID,NREPLIC,8)=T61(IPID,NREPLIC,3)
   T61(IPID,NREPLIC,9)=(T61(IPID,NREPLIC,2)*2.*T61(IPID,NREPLIC,3))*T61(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.
   SOSM=0.
   DO 9922 I=1,10
   SUM = SUM*QLSM62(I,IPID,NREPLIC)
   SOSM=SOSM*QLSM62(I,IPID,NREPLIC)*QLMS62(I,IPID,NREPLIC)
   CALL SUMMARY(TUKEY(I),T62(IPID,NREPLIC,1),T62(IPID,NREPLIC,2),T62(IPID,NREPLIC,3),T62(IPID,NREPLIC,4),T62(IPID,NREPLIC,5),T62(IPID,NREPLIC,6))
   T62(IPID,NREPLIC,7)=SUM/10.
   T62(IPID,NREPLIC,8)=T62(IPID,NREPLIC,3)
   T62(IPID,NREPLIC,9)=(T62(IPID,NREPLIC,2)*2.*T62(IPID,NREPLIC,3))*T62(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)
   SUM=0.
   SOSM=0.
9925 CONTINUE
NREPLIC=1
WRITE(5,5100) (ALABEL(I),I=1,3)
5100 FORMAT(63X*TABLE 7.5*//// DATA SUMMARIES, CENTRAL VALUES, AND SPREADS, MULT. BY 10, OVER 10 DIRICHLET SIMULATIONS FOR Q.L. REG-EST M
FIRST REPLIC. DESIGN 2.*/57X3A10////////

SS=25
PI=0.15
ITABLE=0
WRITE(5,510)

DATA SUMMARY

CENTRAL VALUES

ESTIMATOR SS PID L EXTREME L HINGE MEDIAN U HINGE U
EXTREME MEAN (S.E.) MEDIAN TRIMEAN MIDSPREAD RANGE

WRITE(5,512)

NOTE THAT A ZERO BEFORE A DECIMAL DENOTES AN EXACT ZERO. OTHERWISE, THE ZERO IS ROUNDED.*

CONTINUE
END
FUNCTION GAMMA(GG)

GENERATE A GAMMA RANDOM VARIABLE
TO DO SO USE ALGORITHM GT FROM AHRENS AND DIETER (1974) "COMPUTER
METHODS FOR SAMPLING FROM GAMMA, BETA, POISSON, AND BINOMIAL
DISTRIBUTIONS", P229, COMPUTING. VOL. 12

NOTE THAT FOR 1/77 SIMULATION STUDY, GG RANGES FROM 0.1 TO 9.8

OBTAIN INTEGER PART OF GG

K=GG

OBTAIN FRACTIONAL PART OF GG

F=GG-K

OBTAIN INTEGER PART OF GAMMA

GI=0.
IF (K-0) 14,14,8
8  I=0.
10  GP=1.
    I=I+1.
    UU=URAN(0.)
    GP=GP*UU
    IF (I-K) 10,12,12
12  GI=-ALOG(GP).

OBTAIN FRACTIONAL PART OF GAMMA

14  GF=0.
15  IF (F-0.) 40,40,15
16  B=(2.7182818284590+F)/2.7182818284590
    DF=1./F
    FMIN=F-1.
    UU=URAN(0.)
    GP=B*UU

GENERATE NEW UNIFORM RANDOM NUMBER FOR TESTS IN FOLLOWING STEPS
18 AND 30

UU=URAN(0.)
IF (GP-1.) 18,18,30
GF IS LESS THAN OR EQUAL TO 1.

10 GF=GP**OF
   TEST=EXP(-GF)
   IF (UU-TEST) 40,40,16

GF IS GREATER THAN 1.

30 GF=-ALOG((B-GP)**OF)
   TEST=GF**FMIN1
   IF (UU-TEST) 40,40,16
40 GAMMA=GI+GF
RETURN
END
SUBROUTINE GENXZ(UU,NSS)

GENERATE MULTINOMIAL COMPLETE (X) AND INCOMPLETE (Z) DATA
ALSO CALCULATE MAXIMUM-LIKELIHOOD-ESTIMATES FROM COMPLETE DATA

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,DMLC3
1C3,DP10,EAPMC11,EAPMC12,EAPMC22,EAPMC11,EAPMC12,EAPMC22,ISTOP,N12,N13
2N23,P10,PLMC1,PLMC2,PLMC3,SS,SN,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,W2,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),EPM01),(DEP(2,2),EPM02),(DEP(2,3),EPM03)
EQUIVALENCE (DEP(3,1),EAPM1),(DEP(3,2),EAPM2),(DEP(3,3),EAPM3)
EQUIVALENCE (DQL(1,1),DEPM1),(DQL(1,2),DEPM2),(DQL(1,3),DEPM3)
EQUIVALENCE (DQL(2,1),DEPM1),(DQL(2,2),DEPM2),(DQL(2,3),DEPM3)
EQUIVALENCE (DQL(3,1),DEPM1),(DQL(3,2),DEPM2),(DQL(3,3),DEPM3)
EQUIVALENCE (DQL(4,1),DEPM1),(DQL(4,2),DEPM2),(DQL(4,3),DEPM3)

RECALL THAT NSS IS THE INTEGER FORM OF THE SAMPLE SIZE SS

CALCULATE APPROX AMOUNT OF DATA GOING INTO EACH OF THE 4 GROUPS

H4=PID/2,
H3=H4
H2=H4
H1=1.3*H2
E3=H1+H2

SET END POINTS

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

INITIALIZE Z AND DUMMY VARIABLES Y1, W1, AND V1

Z1 = 0,
Z2 = 0,
Z3 = 0,
Y1 = 0,
Y2 = 0,
W1 = 0,
W2 = 0,
V2 = 0,
V3 = 0

GENERATE X, Z DATA

CALL UNIFORM PSEUDO RANDOM-NUMBER GENERATOR
X(N+1) = (43490275647445 * X(N)) MOD (2EXP(48))
SPECTRAL NUMBERS C(2) = 2.839, C(3) = 2.095, C(4) = 1.819, C(5) = 0.978

CALL URANV(0., NSS, UU)

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
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 OBTAIN REAL FORM OF SHARED INCOMPLETE DATA
C
Z12=N12
Z13=N13
Z23=N23

C OBTAIN COMPLETE DATA X
C
X1=Z1+W1
X2=Z2+V2
X3=Z3+V3

C CALCULATE COMPLETE-DATA MAXIMUM-LIKELIHOOD ESTIMATES
C
PMLC1=X1/SS
PMLC2=X2/SS
PMLC3=(-PMLC1-PMLC2)
IF (PMLC3<0.) 90, 95, 95
90 ISTOP=1
PRINT 92, XNU1, XNU2, XNU3, PI, 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 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,
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

TRUE PERCENT (PROPORTION) INCOMPLETE DATA

TPID = (Z12 + Z13 + Z23) / SS
DPI0 = PIO - TPID
RETURN
END
SUBROUTINE EPM

C PROGRAM TO CALCULATE EXACT POSTERIOR MEAN AND COVARIANCE MATRICES

COMMON DEP(3,3), DQL(4,3), E(4,3), IROBUST, NTS, P1, P2, P3, T PID
COMMON/CALEST/ APMC11, APMC12, APMC22, CONVCRI, COVSKIP, DMLC1, DMLC2, DML
IC3, OPIE, EAPMC12, EAPMC22, EPMC11, EPMC12, EPMC22, ISTOP, N12, N13
2, N23, PID, PMLC1, PMLC2, PMLC3, SS, SSS, TIMAP, TIMEP, TIM(2), TIM21, TIM31, X
3 NU1, XNU2, XNU3, Z1, Z2, Z3, Z12, Z13, Z23, ZIN, 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), EPM01), (DEP(2,2), EPM02), (DEP(2,3), EPM03)
EQUIVALENCE (DEP(3,1), EAPMN1), (DEP(3,2), EAPMN2), (DEP(3,3), EAPMN3)
EQUIVALENCE (DQL(1,1), DPM01), (DQL(1,2), DPM02), (DQL(1,3), DPM03)
EQUIVALENCE (DQL(2,1), DML1), (DQL(2,2), DML2), (DQL(2,3), DML3)
EQUIVALENCE (DQL(3,1), DPM01), (DQL(3,2), DPM02), (DQL(3,3), DPM03)
EQUIVALENCE (DQL(4,1), DAPMN1), (DQL(4,2), DAPMN2), (DQL(4,3), DAPMN3)

N12 = N12 + 1
N13 = N13 + 1
N23 = 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 COMPLETE-DATA CASE, ALL ZIJ = 0.

6 SUM12 = YY1A * GAM(Z2N) * GAM(Z3N)
S12P1 = Z1N * SUM12
S12P1S = (ZIN + 1.) * S12P1
S12P2 = Z2N * SUM12
S12P2S = (Z2N + 1.) * S12P2
S12P1P2 = ZIN * Z2N * SUM12
GO TO 55
8 Z2N12 = Z2N + Z12
Z3N1323 = Z3N + Z13 + Z23

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=IA.
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.
Z1N1A=Z1N+IA
C
DO 40 IIB=1,N131
IB=IIB-1
IF (IB=0) 17,17,20
17 CON13=IA.
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+B1)
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 CONTINUE

CON23=1.
YY3C=YY3B
GO TO 34

31 CI=IC-1
CON23=(T23-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=ZINIA+IB
T13=G*SUM13
T13P1=T13*GG
T13P2=G*S23P2
T13P1SO=T13P1*(GG+1.)
T13P2SO=G*S23P2SO
T13P1P2=T13P2*GG
SUM13=SUM13+T13
S13P1=S13P1+T13P1
S13P2=S13P2+T13P2
S13P1SO=S13P1SO+T13P1SO
S13P2SO=S13P2SO+T13P2SO
S13P1P2=S13P1P2+T13P1P2

40 CONTINUE

C
T12=CON12*SUM13
T12P1=CON12*S13P1
T12P2=CON12*S13P2
T12P1SO=CON12*S13P1SO
T12P2SQ=CON12*S13P2SQ
T12P1P2=CON12*S13P1P2
SUM12=T12+SUM12
S12P1=S12P1+T12P1
S12P2*S12P2+T12P2
S12P1SQ*S12P1SQ+T12P1SQ
S12P2SQ*S12P2SQ+T12P2SQ
S12P1P2*S12P1P2+T12P1P2
90 CONTINUE

C ELEMENTS OF POSTERIOR MEAN OF P GIVEN Z

50 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 SECONDITIM2

C ELEMENTS OF POSTERIOR COVARIANCE MATRIX OF P GIVEN Z

X1X2=S12P1P2/D2
EPMC12=X1X2-PEPM1*PEPM2
X1SQ=S12P1SQ/D2
EPMC11=X1SQ-PEPM1*PEPM1
X2SQ=S12P2SQ/D2
EPMC22=X2SQ-PEPM2*PEPM2
DEPM1=PEPM1-P1
DEPM2=PEPM2-P2
DEPM3=PEPM3-P3
RETURN

230 PRINT 231
231 FORMAT(/// ' EXACT POSTERIOR MEAN. NEGATIVE P.'///)
PRINT 240, PEPM1, EPMC11, X1S0, S12P1, S12P1SO, EPMC22, X2SQ, S12P2
1, S12P2SQ, PEPM3, EPMC12, X1X2, SUM12, S12P1P2, NTS
240 FORMAT(// ' PEPH1=E15.8, X=8H EPMC11=E15.8, X=8H X1S0=E15.8, X=8H S12P1=E15.8, X=8H S12P1SO=E15.8, X=8H PEPM2=E15.8, X=8H PEPM3=E15.8, X=8H EPMC22=E15.8, X=8H EPMC12=E15.8, X=8H X1X2=E15.8, X=8H SUM12=E15.8, X=8H NTS=E15.8, X=8H X=8H')
ISTOP=1
RETURN
END
FUNCTION GAM(X)

EVALUATE GAM(X) FOR CASES FROM PRIOR (0.1, 1.9, 8)

IF (X < 9.2) 1, 35, 35

ARGUMENT IS LESS THAN OR EQUAL TO 9.1. TO INSURE 11 SIGNIFICANT
PLACES OF ACCURACY IN GAMMA, USE 11 SIGNIFICANT-FIGURE VALUE
CALCULATED FROM LOG(GAM(X)) FROM ABRAMOWITZ AND STEGUN OR DAVIS.
CALCULATE GAM(0.1), GAM(1.1), AND GAM(2.1) BY HAND FROM GAM(3.1)
AND RELATION GAM(X+1) = X GAM(X)

1 IF (ABS(X-9.1)-1.E-13) 2, 2, 4
GAM(9.1)
2 GAM=4.9973708194629
RETURN
4 IF (ABS(X-8.1)-1.E-13) 6, 6, 8
GAM(8.1)
6 GAM=6.1695936974051
RETURN
8 IF (ABS(X-7.1)-1.E-13) 10, 10, 12
GAM(7.1)
10 GAM=8.68995685880072
RETURN
12 IF (ABS(X-6.1)-1.E-13) 14, 14, 16
GAM(6.1)
14 GAM=14.45194406569
RETURN
16 IF (ABS(X-5.1)-1.E-13) 18, 18, 20
GAM(5.1)
18 GAM=27.931753738371
RETURN
20 IF (ABS(X-4.1)-1.E-13) 22, 22, 24
GAM(4.1)
22 GAM=8.126228630175
RETURN
24 IF (ABS(X-3.1)-1.E-13) 26, 26, 28
GAM(3.1)
26 GAM=2.1976202783927
RETURN
28 IF (ABS(X-2.1)-1.E-13) 30, 30, 32
GAM(2.1)
30 GAM=1.04648584685
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 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
Y IS THE APPROXIMATED NATURAL LOG(BASE E) OF GAMMA(X)
C
Y=(X-0.5)*ALOG(X)-X+0.91893853320467+(12.*X)-1./(360.*XCU)+1./((
11260.*XFIFTH)
IF (X-22.) 40,45,45
40 Y=Y-1./(1660.*XFIFTH*XSQ)
45 GAM=EXP(Y)
RETURN
END
SUBROUTINE METHODS

FOR INCOMPLETE DATA CALCULATE MY APPROXIMATION, POSTERIOR MODE, AND MAXIMUM-LIKELIHOOD ESTIMATE

INTEGER COVSKIP
DIMENSION A(3,3), B(3,1), IPIVOT(3), IWK(6)
COMMON DEP(3,3), DOL(3,1), E(4,3), IROBUST, NT, P1, P2, P3, TPID
COMMON/CALEST/APMC11, APMC12, APMC22, EP, COVSKIP, DMLC1, DMLC2, DMLC3,
I3, DOL, EAPMC11, EAPMC12, EAPMC22, EAPMC32, EAPMC33, ISTOP, N1, N2, N3,
XNU1, 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), EPM1), (DEP(2,2), EPM2), (DEP(2,3), EPM3)
EQUIVALENCE (DEP(3,1), EAPM1), (DEP(3,2), EAPM2), (DEP(3,3), EAPM3)
EQUIVALENCE (DOL(1,1), DOLM1), (DOL(1,2), DOLM2), (DOL(1,3), DOLM3)
EQUIVALENCE (DOL(2,1), DOLM1), (DOL(2,2), DOLM2), (DOL(2,3), DOLM3)
EQUIVALENCE (DOL(3,1), DOLM1), (DOL(3,2), DOLM2), (DOL(3,3), DOLM3)
EQUIVALENCE (DOL(4,1), DOLM1), (DOL(4,2), DOLM2), (DOL(4,3), DOLM3)
EQUIVALENCE (DEP(1,1), EML1), (DEP(1,2), EML2), (DEP(1,3), EML3)
EQUIVALENCE (DEP(2,1), EPM1), (DEP(2,2), EPM2), (DEP(2,3), EPM3)
EQUIVALENCE (DEP(3,1), EAPM1), (DEP(3,2), EAPM2), (DEP(3,3), EAPM3)
EQUIVALENCE (DOL(1,1), DOLM1), (DOL(1,2), DOLM2), (DOL(1,3), DOLM3)
EQUIVALENCE (DOL(2,1), DOLM1), (DOL(2,2), DOLM2), (DOL(2,3), DOLM3)
EQUIVALENCE (DOL(3,1), DOLM1), (DOL(3,2), DOLM2), (DOL(3,3), DOLM3)
EQUIVALENCE (DOL(4,1), DOLM1), (DOL(4,2), DOLM2), (DOL(4,3), DOLM3)

IF (IROBUST - 1) > 100, 50

INCOMPLETE-DATA MAXIMUM-LIKELIHOOD ESTIMATE

PMLE1 = PEPM1
PMLE2 = PEPM2
PMLE3 = PEPM3
K=1
5 PL1 = PMLE1
PL2 = PMLE2
PMLE1 = PMLE1 + PMLE2
PMLE3 = PMLE1 + PMLE3
PMLE2 = PMLE2 + PMLE3
IF (PMLE1 - 1) < 1E-10
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)-0.000001) 17,20,20
16 IF (ABS(PMLE1-PL1)/PMLE1-CONVCR1) 17,20,20
17 IF (PMLE2-0.1) 18,18,19
18 IF (ABS(PMLE2-PL2)-0.000001) 25,20,20
19 IF (ABS(PMLE2-PL2)/PMLE2-CONVCR2) 25,20,20
20 K=K+1
IF (K-1000) 5,5,21
21 PRINT 22, K,XNU1,P1,PL1,PL2,CONVCR1,IRONBUST,NTS,TPID,PMLE3,XNU2,P2
1,PMLE1,PMLE2
22 FORMAT(15,13,1,F9.4,3X,F9.6,3X,F11.8,3X,F11.8,3X,F7.5,12,1*I3,F6.2,1,F6.2,1,F6.2,1,F6.2)
IF (K-1030) 25,25,250

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

INCREMENT COUNTER FOR NUMBER OF ITERATIONS FOR CONVERGENCE

CALL KTITER(K,1)

C

POSTERIOR MODE

C

50 T1=Z1N-1.
T2=Z2N-1.
K=1
D=SSN-3.
PPMD1=PEPM1
PPMD2=PEPM2
PPMD3=PEPM3
53 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.) PPM01=0.
PPMD2=(T2*PPMD2*(TEMP+Z23/PPMD23))/D
IF (PPMD2.LT.0.) PPM02=0.
PPMD3=1.-PPMD1-PPMD2
IF (PPMD3-0.) 71,64,64
64 IF (PPMD01-0.1) 65,65,66
65 IF (ABS(PPMD1-PL1)-0.00001) 67,70,70
66 IF (ABS(PPMD1-PL1)/PPMD1-CONVCR1) 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-CONVCR2) 75,70,70
70 K=K+1
IF (K-1000) 55,55,71
71 PRINT 72, K, XNU1, P1, PL1, PL2, CONVCR1, IROBUST, NTS, TPI0, PPM03, 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* PPM03*F6.4* XNU2*F
39.4* P2*F9.6* PPMD01*F11.8* PPMD2*F11.8)
IF (K-1030) 75,75,250

CONVERGENCE FOR POSTERIOR MODE INCOMPLETE DATA

75 EPMD1=PPMD1-PEPM1
EPMD2=PPMD2-PEPM2
EPMD3=PPMD3-PEPM3
DPMD1=PPMD1-P1
DPMD2=PPMD2-P2
DPMD3=PPMD3-P3

INCREMENT COUNTER FOR NUMBER OF ITERATIONS FOR CONVERGENCE

J=2
IF (IROBUST.EQ.2) J=5
CALL KTITER(K,J)
MY TAYLOR-SERIES APPROXIMATED POSTERIOR MEAN AND COVARIANCE MATRICES

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<0.0) 121*114*114

114 IF (PAPM1<0.1) 115*115*116
115 IF (ABS(PAPM1-PL1)-0.00001) 117*120*120
116 IF (ABS(PAPM2-PL1)-0.00001) 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-CONVCR1) 125*120*120
120 K=K+1
IF (K>1000) 105*105*121
121 PRINT 122* K*XNU1*P1*PL1*PL2*CONVCR1*IROBUSt*NTS*TPID*PAPM3*XNU2*P
122 FORMAT(* EXCESS NUMBER OF ITERATIONS FOR INCOMPLETE-DATA PAPM
1. IS*I3* XNU1**F9.4* P1**F9.6* PL1**F11.8* PL2**F11.8* CONV
2(R0)*F7.5* IROBUSt**I2* NTS**I3* TPID**F6.2* PAPM3**F6.4* XNU2**F
39.4* P2**F9.6* PAPM1**F11.8* PAPM2**F11.8)
IF (K>1030) 125*125*250

CONVERGENCE FOR T.S. APPROX. POSTERIOR MEAN INCOMPLETE DATA

125 EAPMN1=PAPM1-PEPM1
EAPMN2=PAPM2-PEPM2
EAPMN3=PAPM3-PEPM3
DAPMN1=PAPM1-P1
DAPMN2=PAPM2-P2
DAPM3 = PAPM3 - P3

INCREMENT COUNTER FOR NUMBER OF ITERATIONS FOR CONVERGENCE

IF (IROBUST = 1) 127, 128, 129

J = 3
GO TO 130

J = 4
GO TO 130

J = 6

CALL KTITER(K, J)
IF (IROBUST .GT. 0) RETURN

APPROXIMATED POSTERIOR VAR/COV MATRIX. NONITERATIVE METHOD.

P12 = PAPM1 + PAPM2
P13 = PAPM1 + PAPM3
P23 = PAPM2 + PAPM3

CAUTION. INSURE THAT P12, P13, AND P23 ARE NOT IN COMMON FROM
GENERATED P1, P2, AND P3

R12 = PAPM1 / P12
R13 = PAPM1 / P13
R21 = PAPM2 / P12
R23 = PAPM2 / P23

SSN = SUM OF DATA PLUS SUM OF PRIOR PARAMETERS XNU1

T = SSN * (SSN + 1.)
P12SQ = P12 * P12
P13SQ = P13 * P13
P23SQ = P23 * P23
ZRPI21 = Z12 * R21 / P12
ZRPI3 = Z13 * R13 / P13
ZRPI2 = Z12 * R12 / P12
ZRPI23 = Z23 * R23 / P23

CALCULATE A(1, 1)
A112 = (ZRPI21 + Z13 * P13)**2 / T
A113 = (ZRPI21 * R21 / (P12 * T)
A114 = Z13 / (P13SQ * T)
A(1, 1) = -1. + A112 - A113 - A114

CALCULATE A(1, 2)
A121 = (ZRPI2 - ZRPI2) * (ZRPI21 + Z13 / P13)
A122 = ZRPI2 * R21 / P12
\[ A_{123} = \frac{ZRP_{13}/P_{13}}{A(1,2) = 2.0(AI_{21} + AI_{22} - AI_{23})/T} \]
\[ \text{TEMP} = Z_{12}/P_{12S} \]

C CALCULATE A(1,3)
\[ A_{131} = (ZRP_{12} - ZRP_{13})^{.2} \]
\[ A_{132} = ZRP_{12}R_{12}/P_{12} + ZRP_{13}R_{13}/P_{13} \]
\[ A(1,3) = (A_{131} - A_{132})/T \]

C CALCULATE B(1,1)
\[ B(1,1) = -(SSN*PAPM_{1}*P_{23} + ZRP_{12}PAPH_{2} + ZRP_{13}PAPH_{3})/T \]

C CALCULATE A(2,1)
\[ A_{211} = (ZRP_{21} + Z_{13}/P_{13})*(ZRP_{23} - ZRP_{21}) \]
\[ A_{212} = \text{TEMP}R_{21}^{.2} \]
\[ A(2,1) = (A_{211} + A_{212})/T \]

C CALCULATE A(2,2)
\[ A_{221} = \text{TEMP}PAPM_{1} + Z_{23}(1.0 - 2.0PAPM_{1})/P_{23S} \]
\[ A_{222} = \text{TEMP}PAPM_{2} + Z_{13}(1.0 - 2.0PAPM_{2})/P_{13S} \]
\[ A_{223} = \text{TEMP}(Z_{12} - 2.0PAPM_{1}PAPM_{2})/P_{12S} \]
\[ A_{224} = Z_{13}Z_{23}(P_{12} - 2.0PAPM_{1}PAPM_{2})/(P_{13S}P_{23S}) \]
\[ A(2,2) = (T_{1} - A_{221} + A_{222} + A_{223} + A_{224})/T \]

C CALCULATE A(2,3)
\[ A_{231} = (ZRP_{12} + Z_{23}/P_{23})*(ZRP_{13} - ZRP_{12}) \]
\[ A_{232} = \text{TEMP}R_{12}^{.2} \]
\[ A(2,3) = (A_{231} + A_{232})/T \]

C CALCULATE B(2,1)
\[ B(2,1) = PAPM_{1}PAPM_{2}(SSN + TEMP)/T \]

C CALCULATE A(3,1)
\[ A_{311} = (ZRP_{21} - ZRP_{23})^{.2} \]
\[ A_{312} = ZRP_{21}R_{21}/P_{12} \]
\[ A_{313} = ZRP_{23}R_{23}/P_{23} \]
\[ A(3,1) = (A_{311} - A_{312} - A_{313})/T \]

C CALCULATE A(3,2)
\[ A_{321} = (-ZRP_{12} - Z_{23}/P_{23})*(ZRP_{21} - ZRP_{23}) \]
\[ A_{322} = ZRP_{12}R_{21}/P_{12} \]
\[ A_{323} = ZRP_{23}/P_{23} \]
\[ A(3,2) = 2.0(A_{321} + A_{322} - A_{323})/T \]

C CALCULATE A(3,3)
\[ A_{331} = (ZRP_{12} + Z_{23}/P_{23})^{.2} \]
\[ A_{332} = ZRP_{12}R_{12}/P_{12} \]
\[ A_{333} = Z_{23}/P_{23S} \]
\[ A(3,3) = (T - A_{331} - A_{332} - A_{333})/T \]

C CALCULATE B(3,1)
\[ B(3,1) = -(SSN*PAPM_{2}P_{13} + ZRP_{21}PAPH_{1} + ZRP_{23}PAPH_{3})/T \]
SOLVE SYSTEM A*X=B FOR X. X IS VECTOR OF COVARIANCES C11, C12, C22

CALL MATINV(3, 3, A, 1, 8, 1, DETERM, ISCALE, IPIVOT, IWK)
IF (ABS(DETERM) - 5.0E-14) 212, 212, 220
212 PRINT 214, DETERM, XNU1, XNU2, P1, P2, PAPM1, PAPM2, (A(I,J), J=1,3), B
11, 1, I=1, 3)
214 FORMAT(/ /* SINGULAR SYSTEM. DETERM=*F18.14, NUMBER OF ITERATIONS
29, 6/* AX=B MID-CALC IS*3E20.8, 5X*XI*, 3X*==*E23.8/
33X*3E20.8, 5X*X2*, 3X*, ==*E23.8/34X*3E20.8, 5X*X3*, 3X*, ==*E23.8/
60 TO 230

DIFFERENCE OF APPROXIMATED POSTERIOR COVARIANCES FROM EXACT
POSTERIOR COVARIANCES. C11, C12, AND C22

220 APMC11 = B(1, 1)
APMC12 = B(2, 1)
APMC22 = B(3, 1)
IF (APMC11 - 0.) 223, 225, 221
221 IF (APMC22 - 0.) 223, 225, 222
222 EAPMC11 = APMC11 - EPMC11
EAPMC12 = APMC12 - EPMC12
EAPMC22 = APMC22 - EPMC22
RETURN
223 PRINT 226, APMC11, APMC12, APMC22, EPMC11, EPMC12, EPMC22, XNU1, XNU2, PAP
IM1, PAPM2, NTS
226 FORMAT(/ /* APPROXIMATED VARIANCE IS NEGATIVE. APMC11=*E21.14, APM
C12=*E18.11, APMC22=*E18.11, EPMC11=*E18.11, EPMC12=*E18.11, EPMC22=*E18.11
2ZEPMC22=*E20.13, XNU1=*F4.0, XNU2=*F4.0, PAPM1=*F5.3, PAPM2=*F5
4.3, NTS=*13/)"
SUBROUTINE COUNTS(BIAS, RELDIFF, J)

COUNT NUMBER OF NUMXZ UNTERMINATED TRIALS THAT HAVE NEGATIVE,
ZERO, AND POSITIVE BIAS AND THAT HAVE ABSOLUTE RELATIVE
DIFFERENCES LESS THAN CERTAIN PERCENTAGES.

BIAS = (APPROX-EXACT) OR (APPROX-GENERATED P)
RELDIFF = ABS(BIAS/EXACT) OR ABS(BIAS/GENERATED P)
(RECALL THAT COV IS NEG SO WANT DENOMINATOR INCLUDED IN ABS VALUE)

J DENOTES, IN SUBSEQUENT ORDER, ONE OF EAPMC11, EAPMC12, EAPMC22,
(BIAS OF APPROX T.S. EXPANSION FOR EXACT POSTERIOR COV)
DMLC1 AND DMLC2 (COMPLETE-DATA MLE BIAS FROM GENERATED OR GIVEN P)
(THUS, J=3 REFERS TO EAPMC22)

COMMON DEP(3,3), DQL(4,3), E(4,3), IROBUST, NTS, P1, P2, P3, TPID
COMMON/BIASRD/COUNT(3,8), COUNTRD(8,8)

AB=ABS(BIAS)
IF (AB-1.E-15) 3,3,1
1 IF (BIAS-0.) 2,2,4
NEGATIVE BIAS
2 COUNTB(1,J)=COUNTB(1,J)+1.
GO TO 5
ZERO BIAS (CDC 6600 COMPUTER ACCURACY IS 14 SIGN FIGURES BUT
CONSIDER ONLY 15 DECIMAL PLACES FOR ZERO BIAS
3 COUNTB(2,J)=COUNTB(2,J)+1.
GO TO 5
POSITIVE BIAS
4 COUNTB(3,J)=COUNTB(3,J)+1.

25 PERCENT RELATIVE DIFFERENCE
5 IF (RELDIFF-0.25) 8, 8, 30
8 COUNTRD(8,J)=COUNTRD(8,J)+1.
20 PERCENT RELATIVE DIFFERENCE
10 IF (RELDIFF-0.20) 10, 10, 30
10 COUNTRD(7,J)=COUNTRD(7,J)+1.
15 PERCENT RELATIVE DIFFERENCE
12 COUNTRD(6,J)=COUNTRD(6,J)+1.
15 IF (RELDIFF-0.15) 12,12,30
10 COUNTRD(5,J)=COUNTRD(5,J)+1.
10 PERCENT RELATIVE DIFFERENCE
14 COUNTRD(4,J)=COUNTRD(4,J)+1.
5 PERCENT RELATIVE DIFFERENCE
16 COUNTRD(4,J)=COUNTRD(4,J)+1.
C 1 PERCENT RELATIVE DIFFERENCE
   IF (RELDIFF-0.01) 18,18,30
   18 COUNTR(3,J)=COUNTR(3,J)+1.
C 0.1 PERCENT RELATIVE DIFFERENCE
   IF (RELDIFF-0.001) 20,20,30
   20 COUNTR(2,J)=COUNTR(2,J)+1.
C 0.01 PERCENT RELATIVE DIFFERENCE
   IF (RELDIFF-0.0001) 22,22,30
   22 COUNTR(1,J)=COUNTR(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(4,2)
   40 CONTINUE
      RETURN
   END
SUBROUTINE ESTMSE(Y, Y2, XY, X, X2, TXMSE, MSE)

CALCULATE ESTIMATES OF MSE AND SAMPLE VARIANCE OF THESE ESTIMATES.

REAL MSE(6), MSECV

FOR TERM = (PE1 - P1)^2 + (PE2 - P2)^2 + (PE3 - P3)^2 AND CONTROL-VARIATE
TERMVC = (PMLECD1 - P1)^2 + (PMLECD2 - P2)^2 + (PMLECD3 - P3)^2
FOR PE- DENOTING ONE OF ESTIMATORS EPM, APM, PMO, AND MLE AND
PMLECD- DENOTING COMPLETE-DATA MLE

Y = SUM OF N TERM
Y2 = SUM OF N TERM^2 TERM
XY = SUM OF N TERM*TERMVC
X = SUM OF N TERMVC
X2 = SUM OF N TERMVC^2 TERMCV
N = NUMBER OF TERMS
TXMSE = TRUE MEAN SQ ERROR OF CONTROL VARIATE
MSECV = USUAL SAMPLE MSE FOR THE CONTROL VARIATE

MSE(1) IS USUAL MSE (B=0 IN MSE REGRESSION ESTIMATE)
MSE(2) IS VAR OF MSE(1)
MSE(3) IS USUAL CONTROL-VARIATE MSE (B=1 IN MSE REGRESSION EST)
MSE(4) IS VAR OF MSE(3)
MSE(5) IS LEAST-SQUARES REGRESSION ESTIMATE MSE (B=LEAST-SQS EST)
MSE(6) IS VAR OF MSE(5)

NOTE THAT MSE(5) SHOULD HAVE SMALLEST VARIANCE. HOWEVER, IT WILL
BE A BIASED ESTIMATE. HENCE, USE IT IN ANALYSES ONLY IF IT
DIFFERS FROM EITHER ONE OF TWO UNBIASED ESTIMATES BY NO MORE
THAN 1%. (IE, IT CAN DIFFER BY MORE THAN 1% FROM EITHER MSE(1) OR
MSE(3) BUT NOT BOTH.)

GENERAL FORM OF ESTIMATED MSE IS

MSE = MSE(1)*B*(TXMSE - MSECV)

USUAL MSE (B=0)

XN = N*(N-1)
MSE(1) = Y/N
MSE(2) = (Y2 - N*MSE(1)*MSE(1))/XN

USUAL CONTROL-VARIATE MSE (B=1)
\begin{align*}
XN &= N \cdot (N - 2) \\
T0 &= 2 \cdot X \cdot Y \\
MSECV &= X / N \\
T1 &= N \cdot MSECV \\
T2 &= 2 \cdot N \cdot TXMSE \\
T3 &= T2 \cdot TXMSE / 2 \\
D &= TXMSE - MSECV \\
MSE(3) &= MSE(1) + D \\
MSE(4) &= (Y2 - T0 + X2 + T2 \cdot (MSE(1) - MSECV) + T3 - N \cdot MSE(3) \cdot MSE(3)) / XN \\
C &= \text{LEAST-SQUARES REGRESSION-ESTIMATE MSE \ (B = LEAST-SQUARES ESTIMATE)} \\
B &= (XY - T1 \cdot MSE(1)) / (X2 - T1 \cdot MSECV) \\
B2 &= B \cdot B \\
MSE(5) &= MSE(1) + B \cdot D \\
MSE(6) &= (Y2 - B \cdot T0 + B2 \cdot X2 + B \cdot T2 \cdot (MSE(1) - B \cdot MSECV) + B2 \cdot T3 - N \cdot MSE(5) \cdot MSE(5)) / XN \\
C &= \text{RETURN} \\
\text{END}
\end{align*}
SUBROUTINE KTITER(K, J)

INCREMENT COUNTERS FOR (1) AVERAGING NUMBER OF ITERATIONS AN
ESTIMATOR REQUIRED AND (2) DETERMINING HOW MANY CASES IN A
REPLICATION TOOK A SPECIFIED NUMBER OF ITERATIONS

K IS NUMBER OF ITERATIONS REQUIRED TO MEET CONVERGENCE CRITERION
J DENOTES, IN SUBSEQUENT ORDER, ONE OF ESTIMATORS MLE, PMDRO,
APMRO, APMR1, PMDR2, APMR2 (THUS, J=4 REFERS TO APMR1)

COMMON OEP(3,3), DQLU(4,3), EU(4,3), IROBUST, NTS, P1, P2, P3, TPID
COMMON/XDATA/XDATA(2), IDATA(6)
COMMON/ITKT/AVNUMIT(6), CTNUMIT(6,10)

FOR AVERAGING NUMBER OF ITERATIONS FOR JTH ESTIMATOR

AVNUMIT(J) = AVNUMIT(J) + K

INCREMENT COUNTER FOR NUMBER OF ITERATIONS

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, (EII, JJ), JJ = 1, 2
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.)

1
Z=2F7.Z/5X* IROBUST=11* PEPM1,2=2F6.4* PMLE1,2=2F6.4* PPMD1,2=2F6.4* PAPM1,2=2F6.4* Z=6F4.01
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
SORT DATA IN ASCENDING ORDER
CALL ASORT(X,1,N)
C
MEDIAN
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
HINGES
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
EXTREMES
15 LE=X(1)
UE=X(N)
RETURN
END
SUBROUTINE BESTEST(BESTQL)

BY TWO DIFFERENT CRITERION (SUMMED ABSOLUTE RELATIVE DIFFERENCE
RELO-- AND SUMMED SQUARED ERROR SE-- FOR SUM BEING OVER THE THREE
COMPONENTS OF AN ESTIMATOR) DETERMINE WHICH ESTIMATOR IS BEST FOR
A GIVEN ONE OF THE TRINOMIAL SIMULATION TRIALS

TIES. SCORE AS BEST EACH ESTIMATOR THAT TIES FOR BEST.

THE FOUR ESTIMATORS IN E SHOULD BE IN THE FOLLOWING ORDER PEPM,
PMLE, PPMD, PAPM. BIASES SHOULD BE IN CORRESPONDING ORDER.

DIMENSION BESTQL(4,2),RDEP(3),REQL(4),RELP(3,3),RELDQL(4,3)
DIMENSION SEEP(3),SEQL(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,CTRGLQ(3),PRDEP(9,3),PRQL(9,7),SBI
ASEP(3,3),SBIASEQL(3,7)
COMMON/DATA/XDATA(2),ZDATA(6)

C
IR=IROBUST+1
IF (IROBUST=1) 1,100,150

C FOR EPH COMPARISONS

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 INCORPORATE SUBROUTINE COUNTS TWICE (ONCE FOR EACH OF EP AND QL
COMPARISONS) IN THIS SUBROUTINE TO SAVE PROGRAM EXECUTION COST OF
MANY SUBROUTINE CALLS AND INDEX RESETTINGS.

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(I,I)=SBIASEP(I,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

SBIASEP3,1)-SBIASEP3,1:11

CONTINUE - CALL ASORT(X*1,3)
DO 20
1-1.3
IF (SEEPU).EQ.Xm) BESTEP(I,1)-BESTEP(I,3).

C

C DETERMINE WHETHER ANY PEPH COMPONENT IS ZERO
C
ICK-0
IF (E(1,1)-1.E-10) 30*25*25
25 IF CE(1*2)-1.E-10) 30*26*26
26 IF <E<1*3I-1.E-10I 30*32*32
30
ICK-1.
32 IF (ICK-0) 33*33*56
33 CTRDEP-CTRDEP*1.
DO 35 I-1»3
34 CONTINUE
35 CONTINUE
CALL ASORT(Y*1*3>
00 40 I-1»3
IF (ROEP(I).EO.Y(D) BESTEP (I*2 )-BESTEP( I, 2) »1.
40 CONTINUE
C

C FOR DETERMINING PROPORTION OF CASES FOR WHICH X ABSOLUTE RELATIVE
C DIFFERENCE FOR EACH OF ALL THREE ESTIMATOR COMPONENTS IS LESS THAN
C SPECIFIED AMOUNTS (INTEGRATED IN PART FROM SUBROUTINE COUNTS)
C
DO 55 I-1»3
II-1
53 TO <41*43»45*47**9*51*520*528»530*528
41 IF (RELOEP(I»J»-O.OOO1) 53*42*42
42
II-2
43 IF (RELDEP(I»J)-0.001) 53*44*44
44
II-3
45 IF (RELOEP(I»J)-0.01) 53*46*46
46
II-4
47 IF (RELOEP(I»J)-0.05) 53*48*48
48 II-5
49 IF (RELOEP(I,J) - 0.10) 53, 50, 50
50 II = 6
51 IF (RELOEP(I,J) - 0.15) 53, 52, 52
52 II = 7
520 IF (RELOEP(I,J) - 0.20) 53, 525, 525
525 II = 8
528 IF (RELOEP(I,J) - 0.25) 53, 530, 530
530 II = 9
53 CONTINUE
PRDEP(II,I) = PRDEP(II,1) + 1.
55 CONTINUE
IF (RELOEP(3,1) - 0.15) 541, 543, 543
541 IF (RELDEP(3,1) - 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,J), I = 1, 2, Z(J), J = 1, 2, I = 1, 3, XDATA(I, J, I = 1, 3)
1XDATA(1, J, I = 1, 3)
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*PPM01,2=2F7.4* DEP=3(2F7.4, 3X1, X,Z=2F4.0, 2X, 6F4.0)

FOR QL COMPARISONS, ROUTINE IS CALLED FOR EACH OF THREE
ROBUSTNESS SETS

ROBUSTNESS SET 0. ORIGINAL PRIOR.

56 II = 1
L = 2
ICK = 0

DETERMINE WHETHER ANY P COMPONENT IS ZERO

IF (P1 - E-10) 59, 57, 57
57 IF (P2 - E-10) 59, 58, 58
58 IF (P3 - E-10) 59, 59, 61, 61
59 ICK = 1
61 DO 75 I = II, 4
SEQL(I) = 0.
DO 610 J = 1, 3
V(I) = SEQL(I) + DOL(I, J) * DQL(I, J)
610 CONTINUE
IF (IR - 2) 62, 615, 64
615 IF (I - 3) 75, 75, 63
62 K = 1
GO TO 65
K = 5
GO TO 65
K = K + 3
IF (ABS(DQL(I, 1) - 1.0 E-13) > 69) GO TO 67
IF (DQL(I, 1) < 0.) GO TO 68
C NEGATIVE BIAS
SBIASQL(1/K) = SBIASQL(1/K) + 1.
GO TO 75
C ZERO BIAS (CONSIDER ONLY 13 DECIMAL PLACES)
SBIASQL(2/K) = SBIASQL(2/K) + 1.
GO TO 75
C POSITIVE BIAS
SBIASQL(3/K) = SBIASQL(3/K) + 1.
CONTINUE
CALL ASORT(V, L, 4)
DO 76 I = L, 4
IF (QL(I) .EQ. V(L)) BESTOL(I) = BESTOL(I) + 1.
76 CONTINUE

C UNSORT V FOR ROBUSTNESS SETS
C DO 770 I = L, 4
V(I) = QL(I)
770 CONTINUE
C IF ANY ESTIMATOR IS ZERO, SKIP DIV FOR RELATIVE DIFF FOR ALL EST.
C IF (ICK .GT. 0) RETURN
CTRDQL(1) = CTRDQL(1) + 1.
DO 78 I = 1, 4
RELQ(1) = ABS(DQL(I)) / P1
RELQ(2) = ABS(DQL(1)) / P2
RELQ(3) = ABS(DQL(3)) / P3
RDQ(I) = 0.
DO 77 J = 1, 3
W(J) = RDQ(I) = RDQ(I) + RELQ(J)
77 CONTINUE
78 CONTINUE
CALL ASORT(W, L, 4)
DO 79 J = L, 4
IF (RDQ(J) .EQ. W(L)) BESTOL(J) = BESTOL(J) + 1.
79 CONTINUE
C UNSORT W FOR ROBUSTNESS SETS
C
ROBUSTNESS SET 1. UNIFORM PRIOR.

100 II=3
L=2

SET POSTERIOR MODE EQUAL TO M.L.E.

DO 102 I=1,3
DQL(3,I)=DQL(2,I)
102 CONTINUE
GO TO 61

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

\[ X_{i+1} = 43490275647445 \times X_i \mod(2^{48}). \]

Each random number is generated from the previous one by taking the lower order 48 bits of the 96 bit product produced by \( X_{i+1} = 43490275647445 \times X_i \). 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:

<table>
<thead>
<tr>
<th>C_2</th>
<th>C_3</th>
<th>C_4</th>
<th>C_5</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.839</td>
<td>2.095</td>
<td>1.819</td>
<td>0.978</td>
</tr>
</tbody>
</table>


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 < 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

\[ X_{i+1} = 43490275647445 \cdot X_i \mod(2^{48}). \]

Each random number is generated from the previous one by taking the lower order 48 bits of the 96 bit product produced by \( X_{i+1} = 43490275647445 \cdot X_i \). 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:

<table>
<thead>
<tr>
<th></th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
<th>$C_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.839</td>
<td>2.095</td>
<td>1.819</td>
<td>0.978</td>
</tr>
</tbody>
</table>


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
$\text{Det}(a) = 10^{(100 \times ISCALE)} \times \text{DETERM}$ 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 $A(\text{MAX,MAX})$, $B(\text{MAX,M})$, IPIVOT(\text{MAX}), and IWK(\text{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 $l_n, l_{n-1}, \ldots, l_1$. 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 $\text{DETERM} \neq 0$ does not guarantee accuracy in the solutions of inverse.


Storage: 516 octal locations

Subroutine date: January 1, 1975
REFERENCES


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 $p$ 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.