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 limita-
tion that it not be published.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 informa-
tion 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 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>i</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 \( \mathbf{p} \) 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 \( L(\mathbf{p} - \hat{\mathbf{p}}) = (\mathbf{p} - \hat{\mathbf{p}})'(\mathbf{p} - \hat{\mathbf{p}}) \) for \( \hat{\mathbf{p}} \) an estimator of \( \mathbf{p} \).

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 \( \mathbf{p} \);
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 \( \mathbf{P}_2 \) 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 (4×2²×3) having four levels* and two replications per cell. DESIGN 2 has a mixed-effects model constituting nested factorials (4×10×2²×3) having four levels and two replications per cell within each of four variations of the prior parameter ψ. 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

\[
\text{DATA ALABEL/10HA. NU} = (,10H0.1,0.1,9.,10H8). \quad \text{//}
\]
\[
\text{DATA TLABEL/10HTABLE 7.1} / 
\]

are changed as needed. For values of \(y\) that are less than 10, \(\text{GAM}(y+1)\) is calculated from the relationship \(\text{GAM}(y+1) = y \cdot \text{GAM}(y)\) and a starting value. For \(y\) an integer, the starting value is \(\text{GAM}(2) = 1\) and for \(y = x \cdot 1/3\), for \(x\) an integer, the starting value is \(\text{GAM}(3 \cdot 1/3) = 2.7781584804296\). 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.
<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>
LISTING OF COMPUTER PROGRAM

PROGRAM MAIN (INPUT, OUTPUT, TAPES, TAPE12)

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

INTEGER COVSKIP
REAL MSE(6,7)
DIMENSION BESTQL(4,2,3), CTSEOL(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 QLMS51(10,2,2), QLMS52(10,2,2), QLMS53(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), DOL(4,3), E(4,3), IROBUST, NTS, PI, P2, P3, TPID
COMMON/BEST/BESTEP(3,2), CTRDEP, CTRQL(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, CONVCR, 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), PEMP1), (E(1,2), PEMP2), (E(1,3), PEMP3)
EQUIVALENCE (E(2,1), PMLE1), (E(2,2), PMLE2), (E(2,3), PMLE3)
EQUIVALENCE (E(3,1), PPM1), (E(3,2), PPM2), (E(3,3), PPM3)
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), DEPM1), (DOL(1,2), DEPM2), (DOL(1,3), DEPM3)
EQUIVALENCE (DOL(2,1), DML1), (DOL(2,2), DML2), (DOL(2,3), DML3)
EQUIVALENCE (DOL(3,1), DPM1), (DOL(3,2), DPM2), (DOL(3,3), DPM3)
EQUIVALENCE (DOL(4,1), DAPM1), (DOL(4,2), DAPM2), (DOL(4,3), DAPM3)
DATA ALABEL/10HA./, NU = (1.10HO., 1.0.1, 9.10H8) /
DATA TLABEL/10HTABLE 7.1 /
CONVCR1 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

PMLCI I-TH COMPLETE M.L.E.

PMLEI I-TH INCOMPLETE-DATA M.L.E.

PPMDI I-TH POSTERIOR MODE

SS SAMPLE SIZE

SSN SS * SUM OF PRIOR PARAMETERS XNU1

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
   2 GL=GAMMA(XNU1)
   2 G2=GAMMA(XNU2)
   2 G3=GAMMA(XNU3)
   2 G=GL+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
ISS=1
IPRINT=0
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 *, 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=NXR1=NXR2=NCOV=NXZ
ISTOP=0
COVS=0
AVTPID=AVDPI=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
      SBIAS(I)=0.
29 CONTINUE
   DO 30 I=1,27
      PROEP(I)=0.
30 CONTINUE
   CTRDEP=CTRDL1=CTRDL2=CTRDL3=0.

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

   DO 42 J=1,2
      BESTEP(1,J)=BESTOL(1,J,1)=BESTOL(1,J,2)=BESTOL(1,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

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

   INITIALIZE AVERAGES AND ERROR SUMMARIES TO ZERO
   AVERAGE ESTIMATORS AND THEIR STANDARD ERRORS (S.E.)
   APML1=APEP1=APAP1=APPM1=APMLC1=AP1=0.
   XAPML1=XAPEP1=XAPAP1=XAPP1=XPMLC1=XP1=0.
   APML1=APEPM2=APAM2=APPM2=APMLC2=AP2=0.
   XAPML2=XAPEPM2=XAAPP2=XAPPMD2=XPMLC2=XP2=0.
AVPMN1=AVPMN2=APPMN1=APPMN2=APMD1=APMD2=0.
APFMN1=APFMN2=APPMN1=APPMN2=PGMD1=PGMD2=0.

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

XAPMN1=XAPMN2=XAPMN4=XAPMNSQ=0.
XPM01 =XPM02 =XPM04 =XPM0SQ =0.
XML1 =XML2 =XML4 =XMLSQ =0.

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

YPEMN1 =YPEMN2 =YPEMN4 =YPEMSQ =0.
YPEMN =YPEMN2 =YPEMN4 =YPEMSQ =0.
YPEMSQ =YPEMSQ =YPEMSQ =YPEMSQ =0.
REGAPMN=REGPMN=REGMNC=REGMLC=YPEMN2=YPEMN4=YPEMSQ=YPMD0=YPMD2=YPMSQ=0.
REGAPMN=REGPMN=REGMNC=REGMLC=YPEMN2=YPEMN4=YPEMSQ=YPMD0=YPMD2=YPMSQ=0.
REGAPMN=REGPMN=REGMNC=REGMLC=YPEMN2=YPEMN4=YPEMSQ=YPMD0=YPMD2=YPMSQ=0.

INITIALIZE FOR S.E. OF AVERAGE BIAS

WAPMN1=WPM01=WML1= WAPMN2=WPM02=WML2=0.
UMLC1=UEPN1=UAPMN1=UPM01=UML1=FAPMN1=GAPMN1=GPMD0=0.
UMLC2=UEPMN2=UAPMN2=UPM02=UML2=FAPMN2=GAPMN2=GPMD2=0.

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

AEPMC11=AEPMC11=APMC11=API1MSE=VA1MSE=APROC11=0.
AEPMC12=APM0C12=APMC12=AP1MSE=APROC12=0.
AEPMC22=APM0C22=APMC22=AP2MSE=APROC22=0.
A2EPC11=A2EPC12=A2EPC22=VARMSQ=0.
A2APC11=A2APC12=A2APC22=VAR4=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(UU+NSS)
IF (ISTOP-1) 46, 58, 58

CALCULATE EXACT POSTERIOR MEAN

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 + PPM01
APPMD2 = APPMD2 + PPM02
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 + PPM01

ORIGINAL PAGE IS
OF POOR QUALITY!
XAPPMD2 = XAPPMD2 + PPM2
XAPMLC1 = XAPMLC1 + PMLC1
XAPMLC2 = XAPMLC2 + PMLC2
XAPI = XAPI + P1
XAP2 = XAP2 + P2

Differences of estimators from exact posterior mean
Bias of T.S. approx posterior mean from exact posterior mean
XAPMN1 = XAPMN1 + EAPMN1
XAPMN2 = XAPMN2 + EAPMN2
XXAPMN = EAPMN1 + EAPMN1 + EAPMN2 + EAPMN2 + EAPMN3 + EAPMN3
Mean square error of T.S. APM from exact posterior mean
XAPMNSQ = XAPMNSQ + XXAPMN
For variance of mean square error
XAPMN4 = XAPMN4 + XXAPMN

Bias of posterior mode from exact posterior mean
XPMDO = XPMDO + EPMD0
XPMDSQ = XPMDSQ + XXPMDO
Mean square error of posterior mode from exact posterior mean
For variance of mean square error
XPMDO = XPMDO + XXPMDO

Bias of MLE from exact posterior mean (incomplete data)
XML1 = XML1 + EML1
XML2 = XML2 + EML2
XXML = EML1 + EML1 + EML2 + EML2 + EML3 + EML3
Mean square error of IC-D MLE from exact posterior mean
XMLSQ = XMLSQ + XXML
FOR VARIANCE OF MSE

\[ \text{DIFFERENCE OF APPROXIMATED POSTERIOR COVARIANCES FROM EXACT POSTERIOR COVARIANCES} \]

\[
\begin{align*}
\text{IF (COV_SKIP = 0) } & \text{GO TO 51, 51*50} \\
\text{NCOV} & = \text{NCOV} - 1 \\
\text{GO TO 60} \\
\end{align*}
\]

CUMULATE VALUES TO AVERAGE EXACT POSTERIOR VARIANCES

\[
\begin{align*}
\text{AEPMC11} & = \text{AEPMC11} + \text{EPMC11} \\
\text{AEPMC12} & = \text{AEPMC12} + \text{EPMC12} \\
\text{AEPMC22} & = \text{AEPMC22} + \text{EPMC22} \\
\end{align*}
\]

CUMULATE VALUES TO AVERAGE APPROXIMATE POSTERIOR VARIANCES

\[
\begin{align*}
\text{AAPMC11} & = \text{AAPMC11} + \text{APMC11} \\
\text{AAPMC12} & = \text{AAPMC12} + \text{APMC12} \\
\text{AAPMC22} & = \text{AAPMC22} + \text{APMC22} \\
\end{align*}
\]

FOR S.E. OF COVARIANCE AVERAGES

\[
\begin{align*}
\text{A2EPC11} & = \text{A2EPC11} + \text{EPMC11} - \text{EPMC11} \\
\text{A2EPC12} & = \text{A2EPC12} + \text{EPMC12} - \text{EPMC12} \\
\text{A2EPC22} & = \text{A2EPC22} + \text{EPMC22} - \text{EPMC22} \\
\text{A2APC11} & = \text{A2APC11} + \text{APMC11} - \text{APMC11} \\
\text{A2APC12} & = \text{A2APC12} + \text{APMC12} - \text{APMC12} \\
\text{A2APC22} & = \text{A2APC22} + \text{APMC22} - \text{APMC22} \\
\end{align*}
\]

CUMULATE DIFFERENCES BETWEEN EXACT AND APPROXIMATE POSTERIOR VARIANCES

\[
\begin{align*}
\text{BAPMC11} & = \text{BAPMC11} + \text{EAPMC11} \\
\text{BAPMC12} & = \text{BAPMC12} + \text{EAPMC12} \\
\text{BAPMC22} & = \text{BAPMC22} + \text{EAPMC22} \\
\end{align*}
\]

FOR AVERAGE PERCENT RELATIVE DIFFERENCE

\[
\begin{align*}
\text{PRDC11} & = \text{ABS}(\text{EAPMC11}/\text{EPMC11}) \\
\text{PRDC12} & = \text{ABS}(\text{EAPMC12}/\text{EPMC12}) \\
\end{align*}
\]
PRDC22 = abs(EAPMC22)/EPMC22
APRC11 = APROC11 + PROC11
APRC12 = APROC12 + PROC12
APRC22 = APROC22 + 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
VAR4 = VAR4 + VVV + 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, B)

DIFFERENCE OF ESTIMATORS FROM GENERATED P

YMLC1 = YMLC1 + DMLC1
YMLC2 = YMLC2 + DMLC2
YYMLCD = DMLC1 * DMLC1 + DMLC2 * DMLC2 + DMLC3 * DMLC3
YMLC20 = YMLC20 + YYMLCD
SQMLCD = YYMLCD + 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 * DEPMN1 + DEPMN2 * DEPMN2 + DEPMN3 * DEPMN3

FOR USUAL MEAN SQUARE ERROR OF EPM FROM GENERATED P

YEPMSQ = YEPMSQ + YYEPMN

FOR REGRESSION ESTIMATION AND CONTROL VARIATE MSE EPM

REGEPMO = REGEPMO + YYEPMN + YYMLCD

FOR VARIANCE OF ALL 3 MEAN SQUARE ERRORS

YEPMN4 = YEPMN4 + YYEPMN + YYEPMN

BIAS OF APPROX POSTERIOR MEAN FROM GENERATED P

YAPMN1 = YAPMN1 + DAPMN1
YAPMN2 = YAPMN2 + DAPMN2
YYAPMN = DAPMN1 * DAPMN1 + DAPMN2 * DAPMN2 + DAPMN3 * DAPMN3

USUAL MEAN SQUARE ERROR FOR APMN FROM GENERATED P

YAPMSQ = YAPMSQ + YYAPMN
FOR REGRESSION ESTIMATION AND CONTROL VARIATE MSE

REGAPMO = REGAPMO + YYAPMN * YYMLCD

FOR VARIANCE OF MEAN SQUARE ERRORS

YAPMN4 = YAPMN4 + YYAPMN * YYAPMN

FOR BIAS OF POSTERIOR MODE FROM GENERATED P

YPMD1 = YPMD1 + DPM1
YPMD2 = YPMD2 + DPM2
YPMD = DPM1 * DPM1 + DPM2 * DPM2 + DPM3 * DPM3
YPMDSQ = YPMDSQ + YYPM
REGPMDO = REGPMDO + YYPM * YYMLCD
YPMD4 = YPMD4 + YPMD * YYPM

BIAS OF MLE FROM GENERATED P

YML1 = YML1 + DML1
YML2 = YML2 + DML2
YYML = DML1 * DML1 + DML2 * DML2 + DML3 * DML3
YMLSQ = YMLSQ + YYML
REGMLO = REGMLO + YYML * YYMLCD
YML4 = YML4 + YYML * YYML

BY TWO DIFF CRITERION CALC BEST EST FOR THIS TRINOMIAL TRIAL

CALL BESTEST(BESTOL11)

FOR S.E. OF AVERAGE BIAS

WAPMN1 = WAPMN1 + EAPMN1 * EAPMN1
WAPMN2 = WAPMN2 + EAPMN2 * EAPMN2
WPM1 = WPM1 + EPM1 * EPM1
WPM2 = WPM2 + EPM2 * EPM2
WML1 = WML1 + EML1 * EML1
WML2 = WML2 + EML2 * EML2
UMLC1 = UMLC1 + DMLC1 * DMLC1
UMLC2 = UMLC2 + DMLC2 * DMLC2
UEPMN1 = UEPMN1 + DEPM1 * DEPM1
UEPMN2 = UEPMN2 + DEPM2 * DEPM2
UAPMN1 = UAPMN1 + DAPMN1 * DAPMN1
UAPMN2 = UAPMN2 + DAPMN2 * DAPMN2
C CALCULATE ROBUSTNESS ESTIMATORS (FOR QUADRATIC-LOSS COMPARISON ONLY). USE WRONG PRIOR IN ESTIMATES.
C EXCLUDE EPM BECAUSE OF EXPENSE.

C ROBUSTNESS SET 1. UNIFORM PRIOR. RECALCULATE 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 + 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 + SMLCD

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

C 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
SSM=SS*XNU
IROBUST=2
CALL METHODS
IF (ISTOP-1) 57,58
57 GAPMN1=GAPMN1+DAPMN1
   GAPMN2=GAPMN2+DAPMN2
   AQPMN1=AQPMN1+PAPM1
   AQPMN2=AQPMN2+PAPM2
   QAPMN=DAPMN1+DAPMN1+DAPMN2+DAPMN2+DAPMN3+DAPMN3
   QAPMSQ=QAPMSQ+QQAPHN
   REGAPM2=REGAPM2+QQAPMN+YYMLCD
   QAPMN4=QQAPMN+QQAPMN+QQAPMN
   YMMLCD=YMLC22+YYMLCD
   REGMLC2=REGMLC2+SQMLCD
C
   QPMD1=QPMD1+DPMD1
   QPMD2=QPMD2+DPMD2
   AQPM1=AQPM1+PPMD1
   AQPM2=AQPM2+PPMD2
   QPMD=DPMD1+DPMD1+DPMD2+DPMD2+DPMD3+DPMD3
   QPMDSQ=QPMDSQ+QQPMDO
   REGPMDO=REGPMDO+QQPMDO+YYMLCD
   QPMD4=QQPMDO+QQPMDO
C
   GAPMN1=GAPMN1+DAPMN1+DAPMN1
   GAPMN2=GAPMN2+DAPMN2+DAPMN2
   GPMD1=GPMD1+DPMD1+DPMD1
   GPMD2=GPMD2+DPMD2+DPMD2
   AQPM1=AQPM1+PAPM1+PAPM1
   AQPM2=AQPM2+PAPM2+PAPM2
   GPMD1=GPMD1+PAPM2+PAPM2
   GPMD2=GPMD2+PAPM2+PAPM2
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
AVERAGE OVER NUMXZ TRINOMIAL RESULTS FOR FIXED XNU={XNU1,XNU2,XNU3} 
GENERATED P={P1,P2,P3}, PID, AND SS

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
AVPMM1=AVPMM1/NXZR1
AVPMM2=AVPMM2/NXZR1
AQPMM1=AQPMM1/NXZR2
AQPMM2=AQPMM2/NXZR2

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

STANDARD ERRORS OF ESTIMATOR MEANS

SE1=SQR((XAPMLE1-APMLE1)*APMLE1/ND)
SE2=SQR((XAPMLE2-APMLE2)*APMLE2/ND)
SE3=SQR((XAPEPM1-APEPM1)*APEPM1/ND)
SE4=SQR((XAPEPM2-APEPM2)*APEPM2/ND)
SE5=SQR((XAPAPM1-APAPM1)*APAPM1/ND)
SE6=SQR((XAPAPM2-APAPM2)*APAPM2/ND)
SE7=SQR((XAPPMD1-APPMD1)*APPMD1/ND)
SE8=SQR((XAPPMD2-APPMD2)*APPMD2/ND)
| SE9  | SORT(U(XAPMLC1-NUMXZ*APMLC1*APMLC1)/ND) |
| SE10 | SORT(U(XAPMLC2-NUMXZ*APMLC2*APMLC2)/ND) |
| SE11 | SORT(U(XAP1-NUMXZ*AP1*AP1)/ND) |
| SE12 | SORT(U(XAP2-NUMXZ*AP2*AP2)/ND) |
| SE13 | SORT(U(AFPMN1-NXZR1*AVPMN1*AVPMN1)/NDR1) |
| SE14 | SORT(U(AFPMN2-NXZR1*AVPMN2*AVPMN2)/NDR1) |
| SE15 | SORT(U(AGPMN1-NXZR2*AQPMN1*AQPMN1)/NDR2) |
| SE16 | SORT(U(AGPMN2-NXZR2*AQPMN2*AQPMN2)/NDR2) |
| SE17 | SORT(U(PGMD1-NXZR2*AOPMD1*AOPMD1)/NDR2) |
| SE18 | SORT(U(PGMD2-NXZR2*AOPMD2*AOPMD2)/NDR2) |
| PRINT 1079, NUMXZ |
| PRINT 1080, APMLE1, SE1, APEPM1, SE3, APAPM1, SE5, APPMD1, SE7, APMLE2, SE2, APEPM2, SE4, APAPM2, SE6, APPMD2, SE8, APMLE2, SE9 |

1079 FORMAT(12F11.7) 1080 FORMAT(12F11.7) 12F11.7

C

AVERAGE BIASES AND COVARIANCE ESTIMATORS

| XAPMN1 = XAPMN1 / NUMXZ |
| XAPMN2 = XAPMN2 / NUMXZ |
| XPMD1 = XPMD1 / NUMXZ |
| XPMD2 = XPMD2 / NUMXZ |
| XXLM1 = XXLM1 / NUMXZ |
| XXLM2 = XXLM2 / 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/NXZR1
QAPMN2-QAPMN2/NXZR2
QPM1=QPM1/NXZR2
QPM2=QPM2/NXZR2
EBIAS1(NREPLIC,ISS,IPID)=XAPMN1
EBIAS2(NREPLIC,ISS,IPID)=XPM1
EBIAS3(NREPLIC,ISS,IPID)=XML1

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

SE21=SQR{(WAPMN1-NUMXZ*XAPMN1*XAPMN1)/ND)
SE22=SQR{(WAPMN2-NUMXZ*XAPMN2*XPM12)/ND)
SE23=SQR{(WPM1-NUMXZ*XPM1*XPM1)/ND)
SE24=SQR{(WPM2-NUMXZ*XPM2*XPM2)/ND)
SE25=SQR{(WML1-_NUMXZ*XML1*XML1)/ND)
SE26=SQR{(WML2-NUMXZ*XML2*XML2)/ND)
SE27=SQR{(A2EPC11-NCOV*AEMC11*AEMC11)/NC)
SE28=SQR{(A2EPC12-NCOV*AEMC12*AEMC12)/NC)
SE29=SQR{(A2ECC22-NCOV*AEMC22*AEMC22)/NC)
SE30=SQR{(A2APC11-NCOV*AAPMC11*AAPMC11)/NC)
SE31=SQR{(A2APC12-NCOV*AAPMC12*AAPMC12)/NC)
SE32=SQR{(A2APC22-NCOV*AAPMC22*AAPMC22)/NC)
SE33=SQR{(C11MSE-NCOV*BAPMC11*BAPMC11)/NC)
SE34=SQR{(C12MSE-NCOV*BAPMC12*BAPMC12)/NC)
SE35=SQR{(C22MSE-NCOV*BAPMC22*BAPMC22)/NC)
SE36=SQR{(UMLC1-NUMXZ*YMLC1*YMLC1)/NC)
SE37=SQR{(UMLC2-NUMXZ*YMLC2*YMLC2)/NC)
SE38=SQR{(UEPMN1-NUMXZ*YEPMN1*YEPMN1)/ND)
SE39=SQR{(UEPMN2-NUMXZ*YEPMN2*YPM2)/ND)
SE40=SQR{(UAPMN1-NUMXZ*YAPMN1*YAPMN1)/ND)
SE41=SQR{(UAPMN2-NUMXZ*YAPMN2*YAPMN2)/ND)
SE42=SQR{(UPM1-NUMXZ*YPM1*YPM1)/ND)
SE43=SQR{(UPM2-NUMXZ*YPM2*YPM2)/ND)
SE44=SQR{(UML1-NUMXZ*YML1*YML1)/ND)
SE45=SQR{(UML2-NUMXZ*YML2*YML2)/ND)
SE46=SQR{(FAPMN1-NXZR1*VAPMN1*VAPMN1)/NDR1)
SE47=SQR{(FAPMN2-NXZR1*VAPMN2*VAPMN2)/NDR1)
SE48=SQR{(GAPMN1-NXZR2*QAPMN1*QAPMN1)/NDR2)
SE49=SQR{(GAPMN2-NXZR2*QAPMN2*QAPMN2)/NDR2)
SE50=SQR{(QPM1=QPM1/NXZR2+QPMD1*QPM1)/NDR2)
$SE51=SQR((GPM2-NXZ2\times QPM2\times QPM2)/\text{ND2})$

PRINT 1150, NUMXZ, XAPMN1, SE21, XPM12, SE24, XML2, SE26, YEMN1, SE38, YAPMN1, SE40, YPM1, SE42, YML1, SE44, YMLC2, SE37
1150 FORMAT(// * AVERAGE BIASES AND THEIR STANDARD ERRORS OVER * 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*YEPH*8X*S.*
3E.*,8X*YPMD*8X*S.E.*,9X*YML*8X*S.E.*,8X*YMLC*8X*S.E.*,4
4/10X*P1,10F12.10/10X*P2,10F12.10)
PRINT 1154, AVPMN1, SE13, APMLE1, SE1, AQPMN1, SE15, AQPM1, SE17, AVPMN2,
1SE14, APML2, SE2, AQPM2, SE16, AQPM2, SE18
1154 FORMAT(// * ROBUST-ESTIMATOR AVERAGES AND S.E.* W
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.*4
3P1,4*(F1
32.8,F18.14)// P2,4*(F12.8, F18.14)//)
PRINT 1155, VAPMN1, SE46, YML1, SE44, QAPM1, SE48, QPM1, SE50, VPMN2, SE
147, YML2, SE45, QAPMN2, SE49, QPM2, SE51
1155 FORMAT(// * ROBUST-ESTIMATORS BIASES AND STANDARD ERRORS.*//10X, *
1VAPM*11X*S.E.*,9X*YPMD*YML*9X*S.E.*,11X*QAPM*11X*S.E.*,11X*QPM*1
21X*S.E.*,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(NREP,JSS,IPID)=XAPMNSQ
EMS2(NREP,JSS,IPID)=XPMDSQ
EMS3(NREP, JSS, IPID)=XMLSQ
YMLC20=YMLC20/NUMXZ
YEPMSQ=YEPMSQ/NUMXZ
YAPMSQ=YAPMSQ/NUMXZ
YPMDSQ=YPMDSQ/NUMXZ
YMLSQ=YMLSQ/NUMXZ
SD1=SQR((XAPMN4-NUMXZ\times XAPMNSQ\times XAPMNSQ)/\text{ND})
SD2=SQR((XPM4-NUMXZ\times XPMDSQ\times XPMDSQ)/\text{ND})
SD3=SQR((XHL4-NUMXZ\times XMLSQ\times XMLSQ)/\text{ND})
SD4=SQR((REGMLC0-NUMXZ\times YMLC20\times YMLC20)/\text{ND})
SD5=SQR((YEPMN4-NUMXZ\times YEMPSQ\times YEMPSQ)/\text{ND})
SD6=SQR((YAPMN4-NUMXZ\times YAPMSQ\times YAPMSQ)/\text{ND})
SD7=SQR((YPM4-NUMXZ\times YPMDSQ\times YPMDSQ)/\text{ND})
SD8=SQR((YML4-NUMXZ\times YMLSQ\times YMLSQ)/\text{ND})

C
C DIFFERENCES BETWEEN MEAN SQUARE ERRORS
DM1-XAPMNS-O-XPMDSO
DM2-XAPMNS-O-XPMDSO
DM3-XPMDSO-O-XMLSQ
DM4-YMLC20-O-YEPMSO
DM5-YMLC20-O-YAPMSO
DM6-YMLC20-O-YPMDSQ
DM7-YMLC20-O-YMLSQ
DM8-YEPMSO-O-YAPMSO
DM9-YEPMSO-O-YPMDSQ
DM10-YEPMSO-O-YMLSQ
DM11-YAPMSQ-O-YPMDSQ
DM12-YAPMSQ-O-YMLSQ
DM13-YPMDSQ-O-YMLSQ

PRINT 1200, NUMXZ, XAPMNSQ, SD1, YEPMSO, SD5, YMLC20, SD4, XPMDSQ, SD2, YAP
1MSQ, SD6, YMLSQ, SD8, XMLSQ, SD3, YPMDSQ, SD7

1200 FORMAT(//, 'AVERAGE MEAN SQUARE ERRORS AND THEIR STANDARD ERRORS OVER
1ER+I4+ TRIALS=//17X, 'MSE=12X*S.E.*25X='MSE=12X*S.E.*25X='MSE=12X*S.E
PRINT 1300, DM1, DM9, 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-DEPM=14.9,12X*DEPM-DAPM=14.9,12X
2*DEPM-DAPM=14.9,12X*DEPM-DAPM=14.9,12X*DEPM-DAPM=14.9,12X*DEPM-DAPM=14.9,12X
3*DEPM-DAPM=14.9,12X*DEPM-DAPM=14.9,12X*DEPM-DAPM=14.9,12X*DEPM-DAPM=14.9,12X
4.9,12X*DEPM-DAPM=14.9,12X*DEPM-DAPM=14.9,12X*DEPM-DAPM=14.9,12X*DEPM-DAPM=14.9,12X

XN=NUMXZ
DO 71 I=1,6
IF (I-I4) 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)=AVNUMIT(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=F
2127, NUMXZ=I3, NXZR1=I3, NXZR2=I3, AV NUM ITER FOR MLE=F7
3,3, PHDRO=F7.3, PAPRO=F7.3, APHR1=F7.3, PHDR2=F7.3, APHR2=F7.3
43//)

PRINT 2100, ((CTNUMIT(I,J), J=1,10), I=1,4,3,((CTNUMIT(I,J), J=1,10), I=2,5,3)
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 WAS
IS SPECIFIED AMOUNTS*/10X*1 2 3 4 5 6 7 8
2 8-10 11-15 GT 15*11X*1 2 3 4 5 6 7 8
3-10 11-15 GT 15*11X*1 2 3 4 5 6 7 8
4 PMD2 *10F6.3* APMR2 *10F6.3* PMDR2 *10F6.3*/
APMPSQ YEPMSO*NUMIX*YEPMQ
YAPPSQ*NUMIX*YAPPSQ
YPDNSO*NUMIX*YPMDSQ
YMLPSQ*NUMIX*YMLPS
YMLC20*NUMIX*YMLC20
T=(1-1+P1*P1+P2*P2+P3*P3))/SS
CALL ESTMSE(YEPMSQ»YEPMN4,REGEPMO,YMDS20»YMGREO7,NUMIX,MSE(11))
CALL ESTMSE(YAPMSQ»YAPMN4,REGAPMO,YMC20»YMGAP10,NUMIX,MSE(13))
CALL ESTMSE(YPMDSQ»YPM4,REGPMDO,YMLC20»YMGLO7,NUMIX,MSE(19))
CALL ESTMSE(YMDSQ»YMGLO7,REGMLO1,YMLC21,NUMIX,MSE(25))
CALL ESTMSE(YPMDSQ»YPM4,REGPMO,YMLC22,NUMIX,MSE(31))
CALL ESTMSE(YMDSQ»YMGLO7,REGMLO2,YMLC22,NUMIX,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-L0
1SS COMPARISONS. IP=I2, PID=F4.2, SS=F3.0, NREPLIC=12/18X*
2EPN15X*APM15X*PMO15X*MLE14X*APMR13X*PMDR13X*PMDR2// REG M
3SE *7E18.77* VAR(MSE)*7E18.77* CV MSE *7E18.77* VAR(MSE)*7E18.77*
4 RE MSE *7E18.77* VAR(MSE)*7E18.77/)
C
IF (ISS-1) 2035, 2035, 2040
2035 QLMS1I(IGEN,IPID,NREPLIC)*MSE(5,2)
QLMS2I(IGEN,IPID,NREPLIC)*MSE(5,3)
QLMS3I(IGEN,IPID,NREPLIC)*MSE(5,4)
QLMS4I(IGEN,IPID,NREPLIC)*MSE(5,5)
QLMS5I(IGEN,IPID,NREPLIC)*MSE(5,6)
QLMS6I(IGEN,IPID,NREPLIC)*MSE(5,7)
GO TO 2045
2040 QLMS1I(IGEN,IPID,NREPLIC)*MSE(5,2)
QLMS2I(IGEN,IPID,NREPLIC)*MSE(5,3)
QLMS3I(IGEN,IPID,NREPLIC)*MSE(5,4)
QLMS4I(IGEN,IPID,NREPLIC)*MSE(5,5)
QLMS5I(IGEN,IPID,NREPLIC)*MSE(5,6)
QLMS6I(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 CTEQL(1)=NUMXZ
CTEQ(2)=NXZR1
CTEQ(3)=NXZR2
DO 75 I=1,4
DO 72 IR=1,3
BESTQ(I,1,IR)=BESTQ(I,1,IR)/CTEQ(IR)
BESTQ(I,2,IR)=BESTQ(I,2,IR)/CTEQDQ(IR)
SBIASQ(I,IR)=SBIASQ(I,IR)/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
SBIASQ(K,5)=SBIASQ(K,5)/NXZR1
SBIASQ(K,6)=SBIASQ(K,6)/NXZR2
SBIASQ(K,7)=SBIASQ(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 (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
83 DO 84 II=1,8
PRDQL(II,I)=PRDQL(II,I)/CTRDEP(IR)
84 CONTINUE
85 CONTINUE
DO 88 I=1,3
DO 88 II=1,8
PRDEP(I,J) = PRDEP(I,J)/CTRDEP

PRINT 2050, (BESTEP(I,J), I=1,3), (BESTOL(I,J,K), I=2,4, K=1,3), J=1,2)

PRDEP(I,J) = PRDEP(I,J)/CTRDEP

PRINT 2050, (BESTEP(I,J), I=1,3), (BESTOL(I,J,K), I=2,4, K=1,3), J=1,2)

PRINT 2070, (SBIASEP(I,J), J=1,3), (SBIASQL(I,K), K=1,7), I=1,3

PRINT 2080, (PRDEP(I,J), I=1,3), (PROOL(I,J), J=1,8)

PAROC 11-100. *(AEPMC11-AAPMC11)/AEPMC11
PAROC 12-100. *(AEPMC12-AAPMC12)/AEPMC12
PAROC 22-100. *(AEPMC22-AAPMC22)/AEPMC22

PAROC 11-100. *(AEPMC11-AAPMC11)/AEPMC11
PAROC 12-100. *(AEPMC12-AAPMC12)/AEPMC12
PAROC 22-100. *(AEPMC22-AAPMC22)/AEPMC22

AVERAGE PERCENT RELATIVE DIFFERENCE

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

SQUARE ROOT MSE DIVIDED BY AVERAGEEPV

C11MSE=C11MSE/NCOV
C12MSE = C12MSE / NCOV
C22MSE = C22MSE / NCOV
C11RT1 = SORT(C11MSE) / AEPMC11
C12RT1 = SORT(C12MSE) / AEPMC12
C22RT1 = SORT(C22MSE) / AEPMC22
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 / C11MSE
C12RT2 = SE12MSE / C12MSE
C22RT2 = SE22MSE / C22MSE
DO 91 J = 1, 8
DO 90 J = 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, AEPMC12, SE30, PARDC11, APRDC11, C11RT1, C11RT2,
12, AEPMC12, SE28, AEPMC12, SE31, PARDC12, APRDC12, C12RT1, C12RT2, AEPMC22,
25E29, AEPMC22, 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, S, SS, NREP, (COUNTRD(I, 1), I = 1, 6), (COUNTRD(I,
13), I = 1, 6), (COUNTRD(I, 8), I = 1, 6), (COUNTRD(I, 1), I = 1, 6), (COUNTRD(I,
13), I = 1, 6), (COUNTRD(I, 8), I = 1, 6)
3005 FORMAT(///* PROPORTION DF*I4* CASES IN WHICH PERCENT REL DIFF WAS
1 LESS THAN VARYING AMOUNTS. IP = 12, PID = F4.2, SS = F4.0, NREP =
2C = *12//8X*01 01 1.0 5.0 10, 15,.12X*01 01 1.0 5.0 10, 15,.12X*01 01 1.0
35.0 10, 15.12X*01 01 1.0 5.0 10, 15,.12X*01 01 1.0 5.0 10, 15,.12X*01 01 1.0
43, 6X*C22*F6.3, 6X*C33*F6.3, 6X*C23*F6.3
5//)
PRINT 3010, NCOV, (COUNTB(I, 1), I = 1, 3), (COUNTB(I, 3), I = 1, 3), (COUNTB(I,
18), I = 1, 3), (COUNTB(I, 2), I = 1, 3), (COUNTB(I, 6), I = 1, 3), (COUNTB(I,
7), I = 21, 3)
3010 FORMAT(* PROPORTN DF*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 ZERO POS*/*
2 ZERO POS*7X*NEG ZERO POS*7X*NEG ZERO POS*/*
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+PID
AVDPID=AVDPID+DPID
PRINT 3070, PID, NUMXZ, AVTPID, AVDPID

3070 FORMAT(* GIVEN PID IS F4.2* AVERAGE OVER NUMXZ=13 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
ISS=2
GO TO 1

9904 PID=0.50
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 TRINOMIAL
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************,X,28H ****************************, *SS=50,27H ****
2************************** ESTI-DIR. *,9H*********, * PID=15 *,10H**
3**************,5X,9H*********, * PID=40 *,10H**********, 5X,9H*********, *
4PID=15 *F10H**********,5X,9H*********, * PID=40 *,10H**********/* M
5ATOR GEN. REPLICATION1 REPLICATION2 REPLICATION1 REPLICATION2
6ON2 REPLICATION1 REPLICATION2 REPLICATION1 REPLICATION2
72*/
WRITE(12,8011) (I, (QLMS11(I,J,K), K=1,2), J=1,2), (QLMS12(I,J,K), K
I=1,2), J=1,2), I=1,10)

8011 FORMAT(* APMRO *I2,4(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=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) TABLE(1),(ALABEL(I),I=1,3)
WRITE(12,8020)
8020 FORMAT(* 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(* APHR1 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
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(I
PIPID,NREPLIC,3),T11(IPID,NREPLIC,4),T11(IPID,NREPLIC,5))
T11(IPID,NREPLIC,6)=SUM/10.
T11(IPID,NREPLIC,7)=SQR(SOSM-10.*T11(IPID,NREPLIC,6))
T11(IPID,NREPLIC,8)=(T11(IPID,NREPLIC,7)/T11(IPID,NREPLIC,6))**2
T11(IPID,NREPLIC,9)=(T11(IPID,NREPLIC,3)+T11(IPID,NREPLIC,2)+T11(I
PIPID,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.
SUBROUTINE SUM

DO 9912 I = 1, NREPLIC
  SUM = SUM + OLMS12(I, IPID, NREPLIC)
  SOSM = SOSM + OLMS12(I, IPID, NREPLIC) * OLMS12(I, IPID, NREPLIC)
  CALL SUMMARY(TUKEY(I), OLMS12(I, IPID, NREPLIC), 1, T12(I, IPID, NREPLIC, 1), T12(I, IPID, NREPLIC, 2), T12(I, IPID, NREPLIC, 3), T12(I, IPID, NREPLIC, 4), T12(I, IPID, NREPLIC, 5))
  T12(I, IPID, NREPLIC, 6) = SUM / 10.
  T12(I, IPID, NREPLIC, 7) = SQRT((SOSM - 10. * T12(I, IPID, NREPLIC, 6) * T12(I, IPID, NREPLIC, 7)) / 90.)
  T12(I, IPID, NREPLIC, 8) = T12(I, IPID, NREPLIC, 7) + T12(I, IPID, NREPLIC, 3)
  T12(I, IPID, NREPLIC, 10) = T12(I, IPID, NREPLIC, 4) - T12(I, IPID, NREPLIC, 2)
  T12(I, IPID, NREPLIC, 11) = T12(I, IPID, NREPLIC, 5) - T12(I, IPID, NREPLIC, 1)
  SUM = 0.
  SOSM = 0.
  DO 9913 I = 1, NREPLIC
    SUM = SUM + OLMS21(I, IPID, NREPLIC)
    SOSM = SOSM + OLMS21(I, IPID, NREPLIC) * OLMS21(I, IPID, NREPLIC)
    CALL SUMMARY(TUKEY(I), OLMS21(I, IPID, NREPLIC), 1, T21(I, IPID, NREPLIC, 1), T21(I, IPID, NREPLIC, 2), T21(I, IPID, NREPLIC, 3), T21(I, IPID, NREPLIC, 4), T21(I, IPID, NREPLIC, 5))
    T21(I, IPID, NREPLIC, 6) = SUM / 10.
    T21(I, IPID, NREPLIC, 7) = SQRT((SOSM - 10. * T21(I, IPID, NREPLIC, 6) * T21(I, IPID, NREPLIC, 7)) / 90.)
    T21(I, IPID, NREPLIC, 8) = T21(I, IPID, NREPLIC, 3)
    T21(I, IPID, NREPLIC, 9) = (T21(I, IPID, NREPLIC, 2) + 2. * T21(I, IPID, NREPLIC, 3) + T21(I, IPID, NREPLIC, 4) / 4.
    T21(I, IPID, NREPLIC, 10) = T21(I, IPID, NREPLIC, 4) - T21(I, IPID, NREPLIC, 2)
    T21(I, IPID, NREPLIC, 11) = T21(I, IPID, NREPLIC, 5) - T21(I, IPID, NREPLIC, 1)
    SUM = 0.
    SOSM = 0.
    DO 9914 I = 1, NREPLIC
      SUM = SUM + OLMS22(I, IPID, NREPLIC)
      SOSM = SOSM + OLMS22(I, IPID, NREPLIC) * OLMS22(I, IPID, NREPLIC)
      CALL SUMMARY(TUKEY(I), OLMS22(I, IPID, NREPLIC), 1, T22(I, IPID, NREPLIC, 1), T22(I, IPID, NREPLIC, 2), T22(I, IPID, NREPLIC, 3), T22(I, IPID, NREPLIC, 4), T22(I, IPID, NREPLIC, 5))
      T22(I, IPID, NREPLIC, 6) = SUM / 10.
      T22(I, IPID, NREPLIC, 7) = SQRT((SOSM - 10. * T22(I, IPID, NREPLIC, 6) * T22(I, IPID, NREPLIC, 7)) / 90.)
      T22(I, IPID, NREPLIC, 8) = T22(I, IPID, NREPLIC, 3)
      T22(I, IPID, NREPLIC, 9) = (T22(I, IPID, NREPLIC, 2) + 2. * T22(I, IPID, NREPLIC, 3) + T22(I, IPID, NREPLIC, 4) / 4.
      T22(I, IPID, NREPLIC, 10) = T22(I, IPID, NREPLIC, 4) - T22(I, IPID, NREPLIC, 2)
      T22(I, IPID, NREPLIC, 11) = T22(I, IPID, NREPLIC, 5) - T22(I, IPID, NREPLIC, 1)
31

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) + 2. * T31(IPID, NREPLIC, 5)
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) + 2. * T32(IPID, NREPLIC, 5)
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.)
**Program Listing**

```plaintext
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 TUK1Y(I) = QLMS42(I, IPID, NREPLIC)
CALL SUMMARY(TUKEY, I0, T42(IPID,NREPLIC), 1), T42(IPID,NREPLIC), 2), T42(I
PID,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
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
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 TUK1Y(I) = QLMS51(I, IPID, NREPLIC)
CALL SUMMARY(TUKEY, I0, T51(IPID,NREPLIC), 1), T51(IPID,NREPLIC), 2), T51(I
PID,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
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
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 TUK1Y(I) = QLMS52(I, IPID, NREPLIC)
CALL SUMMARY(TUKEY, I0, T52(IPID,NREPLIC), 1), T52(IPID,NREPLIC), 2), T52(I
PID,NREPLIC), 3), T52(IPID,NREPLIC), 4), T52(IPID,NREPLIC,5))
```
T52(IPID, NREPLIC, 6) = SUM/10.
T52(IPID, NREPLIC, 7) = SQRT((SOSM - 10. * T52(IPID, NREPLIC, 6) * T52(IPID, NREPLIC, 6)) / 90.)
T52(IPID, NREPLIC, 8) = T52(IPID, NREPLIC, 3)
T52(IPID, NREPLIC, 9) = T52(IPID, NREPLIC, 2) + 2. * T52(IPID, NREPLIC, 3) + T52(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.
SOSM = 0.
DO 9921 I = 1, 10
SUM = SUM + QLMS61(I, IPID, NREPLIC)
SOSM = SOSM + 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(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 + QLMS62(I, IPID, NREPLIC)
SOSM = SOSM + 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(IPID, NREPLIC, 3), T62(IPID, NREPLIC, 4), T62(IPID, NREPLIC, 5))
T62(IPID, NREPLIC, 6) = SUM/10.
T62(IPID, NREPLIC, 7) = SQRT((SOSM - 10. * T62(IPID, NREPLIC, 6) * T62(IPID, NREPLIC, 6)) / 90.)
T62(IPID, NREPLIC, 8) = T62(IPID, NREPLIC, 3)
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(9, 5100) (ALABEL(I), I = 1, 3)
5100 FORMAT(63X*TABLE 7.5*//* DATA SUMMARIES, CENTRAL VALUES, AND SPRE IADS, MULT. BY 10, OVER 10 DIRICHLET SIMULATIONS FOR Q.L. REG-EST M
5105 FORMAT(19X, 53H**************************DATA SUMMARY**************************
1*5X, 36H********CENTRAL VALUES**********X17H******SPREADS******
2*ESTIMATOR SS PID L EXTREME L HINGE MEDIAN U HINGE U
3EXTREME*7X*MEAN (S.E.) MEDIAN TRIMEAN MIDSREAD RANGE*
4/)
4991 WRITE(5, 5105) 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*APMR0*3X, I2, F4.2, 5F11.7, 3XF9.6(*F9.7*)*2F9.6, 3X*2F9.6/3
1X*PMRO*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*5F11.7, 3X*F9.6(*F9
3.7*)*2F9.6, 3X*2F9.6/3X*APMR2*5F11.7, 3X*F9.6(*F9.7*)*2F9.6, 3X
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
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 IF (I=4) 14,14,8
  16 IF (F-0.) 40,40,15
  18 B=(2.7182818284590+F)/2.7182818284590
  19 OF=1./F
  20 GF=IF-1.
  21 IF (GF=0.) 10,12,12
  22 GI=-ALOG(GP)

OBTAIN FRACTIONAL PART OF GAMMA

GF=0.
IF (F-0.) 40,40,15

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,DMC1,DMC2,DML
IC3,DPI0,EAPMC11,EAPMC12,EAPMC22,EAPMC11,EAPMC12,EAPMC22,ISTOP,N12,N13
2,N23,P1,PMLC1,PMLC2,PMLC3,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,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)

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
C
INITIALIZE Z AND DUMMY VARIABLES Y1, W1, AND V1

Z1=0,
Z2=0,
Z3=0,
Y1=0
Y2=0
W1=0
W3=0
V2=0
V3=0

GENERATE X, Z DATA

CALL UNIFORM PSEUDO RANDOM-NUMBER GENERATOR
X(N+1) = (43490275647445*H(N)/MOD(2EXP(48))
SPECTRAL NUMBERS C(2)=2.839, C(3)=2.659, 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
Z12=N12
Z13=N13
Z23=N23
C OBTAIN COMPLETE DATA X
X1=Z1+Y1+W1
X2=Z2+Y2+V2
X3=Z3+W3+V3
C CALCULATE COMPLETE-DATA MAXIMUM-LIKELIHOOD ESTIMATES
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, PI0, SS, X1, X2, X3, Z1, Z2, Z3, Z12, Z13
1, N23, PMLC1, PMLC2, PMLC3, NSS, NTS
92 FORMAT(6*GENXZ, COMPLETE-DATA MLE NEGATIVE P. XNU=*3F10.4, P ID=*6.2, SS=*F, X=*3F6.0, Z=*6F6.0, P 2ML=*3F10.6, NSS=*I3, XNU=*6F6.0, TH TRINOMIAL SIMUL=1)
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 CORRECTLY RUN IF BLANK COMMON X,Z DATA IS PUT INTO IKITER, COUNTS,
AND BESTEST (PERHAPS PROBLEMS WITH E EQUIVALENCE STATEMENTS)

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
DPID=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,TPID
COMMON/CALEST/APMC11,APMC12,APMC22,CONVCR1,COVSKIP,OMLC1,OMLC2,OML
1C3,OPID,EAPMC11,EAPMC12,EAPMC22,EPMC11,EPMC12,EPMC22,ISTOP,N12,N13
2,N23,OPID,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),DPMN1),(DQL(3,2),DPMN2),(DQL(3,3),DPMN3)
EQUIVALENCE (DQL(4,1),DAPMN1),(DQL(4,2),DAPMN2),(DQL(4,3),DAPMN3)

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 COMPLETE-DATA CASE, ALL ZIJ=0.

6 SUM12=YY1A*GAM(Z2N)*GAM(Z3N)
S12P1=ZIN*SUM12
S12P1SQ=(ZIN+1.)*S12P1
S12P2=Z2N*SUM12
S12P2SQ=(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=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.EQ.0) 30,30,31
30 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
   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
   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
$S12P2 = S12P2 + T12P2$
$S12P1S = S12P1S + T12P1S$
$S12P2S = S12P2S + T12P2S$
$S12P1P = S12P1P + T12P1P$

30 CONTINUE

**ELEMENTS OF POSTERIOR MEAN OF P GIVEN Z**

59 $D1 = \sum 12 \dagger SSN$
$D2 = D1 * (SSN + 1.)$
$PEPM1 = S12P1 / D1$
$PEPM2 = S12P2 / D1$
$IF (PEPM3 < 0.) 230, 201, 201$

201 CALL SECOND(TIM2)

**ELEMENTS OF POSTERIOR COVARIANCE MATRIX OF P GIVEN Z**

X1X2 = S12P1P / D2
EPMC12 = X1X2 - PEPM1 * PEPM2
X1S = S12P1S / D2
EPMC11 = X1S - PEPM1 * PEPM1
X2S = S12P2S / D2
EPMC22 = X2S - 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, X1S, S12P1, S12P1S, EPMC22, X2S, S12P2
1, S12P2S, PEPM3, EPMC12, X1X2, SUM12, S12P1P, NTS

240 FORMAT(///, X1X2, S12P1P, EPMC12, X1X2, S12P1P, X1S, S12P1S, EPMC22, X2S, S12P2S, PEPM3, SUM12, S12P1P, NTS, I3//)
1STOP = 1
RETURN
END
FUNCTION GAM(X)

EVALUATE GAM(X) FOR CASES FROM PRIOR (0.1, 1.9, 9.8)

IF (X<9.2) 1, 35, 35

ARGUMENT IS LESS THAN OR EQUAL TO 9.1. TO INCREASE 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
2 GAM=49973.708949629
RETURN
4 IF (ABS(X-8.1)-1.E-13) 6, 6, 8
6 GAM=6169.5936974851
RETURN
8 IF (ABS(X-7.1)-1.E-13) 10, 10, 12
10 GAM=868.9568580072
RETURN
12 IF (ABS(X-6.1)-1.E-13) 14, 14, 16
14 GAM=142.45194406569
RETURN
16 IF (ABS(X-5.1)-1.E-13) 18, 18, 20
18 GAM=27.931753783371
RETURN
20 IF (ABS(X-4.1)-1.E-13) 22, 22, 24
22 GAM=6.8126228630175
RETURN
24 IF (ABS(X-3.1)-1.E-13) 26, 26, 28
26 GAM=2.1976202783927
RETURN
28 IF (ABS(X-2.1)-1.E-13) 30, 30, 32
30 GAM=1.046465846854
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

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

INTEGER COVSKIP

DIMENSION A(3,3),B(3,1),PIVOT(3),IWK(6)

COMMON DEP(3,3),DOL(3,1),E(4,3),IROBUST,NTS,P1,P2,P3,P3ID

COMMON/CALES/DEPMC11,DEPMC12,DEPMC22,DEPMC2,DEPMC3,DEPMC4,DEPMC12,DEPMC22,DEPMC2,DEPMC3,DEPMC4,DEPMC12,DEPMC22,DEPMC2,DEPMC3,DEPMC4,DEPMC12,DEPMC22,DEPMC2,DEPMC3,DEPMC4,DEPMC12,DEPMC22,DEPMC2,DEPMC3,DEPMC4

COMMON/TIM/TIMAP,TIMET,TIMET,TIMET,TIMET,TIMET,TIMET,TIMET,TIMET

COMMON/XMU,XMU,XMU,XMU,XMU,XMU,XMU,XMU,XMU

COMMON/ITKT/ITK,ITK,ITK,ITK,ITK,ITK,ITK,ITK,ITK

EQUIVALENCE (E(1,1),PEPM1), (E(1,2),PEPM2), (E(1,3),PEPM3)

EQUIVALENCE (E(2,2),PMLE1), (E(2,3),PMLE2), (E(2,4),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(2,2),EPM1), (DEP(2,3),EPM2), (DEP(2,4),EPM3)

EQUIVALENCE (DEP(3,2),EPMN1), (DEP(3,3),EPMN2), (DEP(3,4),EPMN3)

EQUIVALENCE (DOL(1,1),DEPMN1), (DOL(1,2),DEPMN2), (DOL(1,3),DEPMN3)

EQUIVALENCE (DOL(2,2),DML1), (DOL(2,3),DML2), (DOL(2,4),DML3)

EQUIVALENCE (DOL(3,2),DPM1), (DOL(3,3),DPM2), (DOL(3,4),DPM3)

EQUIVALENCE (DOL(4,1),DAPM1), (DOL(4,2),DAPM2), (DOL(4,3),DAPM3)

C CHECK TO INSURE THAT CONVERGENCE CRITERION IS NOT TOO STRICT

IF (IROBUST-1) 1,100,50

B INCOMPLETE-DATA MAXIMUM-LIKELIHOOD ESTIMATE

PMLE1=PEPM1
PMLE2=PEPM2
PMLE3=PEPM3
K=1

PMLE1=PMLE1
PMLE2=PMLE2
PMLE3=PMLE1+PMLE2
PMLE3=PMLE3
PMLE3=PMLE2
PMLE3=PMLE3
IF (PMLE1-1.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 = 1.0 - PMLE1 - PMLE2
IF (PMLE3 = 0.0) 21, 14, 14

14 IF (PMLE1 = 0.1) 15, 15, 16
15 IF (ABS(PMLE1 - PL1) = 0.00001) 17, 20, 20
16 IF (ABS(PMLE1 - PL1) / PMLE1 - CONVCR) 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 - CONVCR) 25, 20, 20
20 K = K + 1
21 IF (K = 1000) 5, 5, 21
22 PRINT 22, K, XNU1, P1, PL1, PL2, CONVCR, IROBUST, NTS, TPID, PMLE3, XNU2, P2
1, PMLE1, PMLE2
22 FORMAT(// * EXCESS NUMBER OF ITERATIONS FOR INCOMPLETE-DATA M.L.E
IF (K = 1030) 25, 25, 250

CONVERGENCE FOR MAXIMUM-LIKELIHOOD ESTIMATE INCOMPLETE DATA:

C C 25 EML1 = PMLE1 - PEPM1
EML2 = PMLE2 - PEPM2
EML3 = PMLE3 - PEPM3
DML1 = PMLE1 - P1
DML2 = PMLE2 - P2
DML3 = PMLE3 - P3

INCREMENT COUNTER FOR NUMBER OF ITERATIONS FOR CONVERGENCE

CALL KTITER(K, 1)

POSTERIOR MODE

50 T1 = Z1N - 1.
T2 = Z2N - 1.
K = 1
D = SSH - 3.
PPMD1 = PEPM1
PPMD2 = PEPM2
PPMD3 = PEPM3
PL1 = PPM1
PL2 = PPM2
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))/0  
IF (PPMD1.LT.0.) PPMD1=0.  
PPMD2=(T2*PPMD2*(TEMP+Z23/PPMD23))/0  
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(PPM01-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-CONVCR1) 75,70,70  
70 K=K+1  
IF (K-10001) 55,55,71  
71 PRINT 72, K,XNU1,PL1,PL1,PL2,CONVCR1,IROBUST,NTS,TPIO,PPM03,XNU2,P2  
1,PPMD1,PPMD2  
72 FORMAT(* EXCESS NUMBER OF ITERATIONS FOR INCOMPLETE-DATA PPMD  
1. IS*I3,** XNU1**F9.4,** PL1**F9.6,3X*PL1**F11.8,3X*PL2**F11.8/* CONV  
2(RD)**F7.5* IROBUST**I2* NTS**I3* TPIO**F6.2* PPM03**F6.4* XNU2**F  
39.4* P2**F9.6* PPMD1**F11.8* PPMD2**F11.8)  
IF (K-10301) 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=(Z11+PAPM1*(TEMP+Z13/PAPM13))/SSN
     PAPM2=(Z22+PAPM2*(TEMP+Z23/PAPM23))/SSN
     PAPM3=-PAPM1-PAPM2
     IF (PAPM3<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(PAPM1-PL1)/PAPM1<CONVCR) 117 120 120
117 IF (PAPM2<0.1) 118 118 119
118 IF (ABS(PAPM2-PL2)<0.00001) 125 125 125
119 IF (ABS(PAPM2-PL2)/PAPM2<CONVCR) 125 125 125
120 K=K+1
     IF (K<1000) 105 105 121
121 PRINT 122; K,XNU1,P1,PL1,PL2,CONVCR,IBROBUST,NTS,TPIO,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, CONVCR*F7.5
     2, ROI*F7.5, IBROBUST*I2, NTS*I3, TPIO*F6.2, PAPM3*F6.4, XNU2*F
     3, 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 EAPM1=PAPM1-PEPM1
     EAPM2=PAPM2-PEPM2
     EAPM3=PAPM3-PEPM3
     DAPM1=PAPM1-P1
     DAPM2=PAPM2-P2
DAPM3 = PAPM3 - P3

INCREMENT COUNTER FOR NUMBER OF ITERATIONS FOR CONVERGENCE

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

APPROXIMATED POSTERIOR VAR/COV MATRIX. NONITERATIVE METHOD.

150 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 XNUI

T = SSN * (SSN + 1.)
P12SQ = P12 * P12
P13SQ = P13 * P13
P23SQ = P23 * P23
ZR121 = Z12 * R12 / P12
ZR131 = Z13 * R13 / P13
ZR122 = Z12 * R12 / P12
ZR123 = Z23 * R23 / P23
C CALCULATE A(1, 1)
A112 = (ZR121 + Z13 / P13) * Z12 / T
A113 = (ZR121 * R11) / (P12 * T)
A114 = Z13 / (P13 * T)
A(1, 1) = -1. + A112 - A113 - A114
C CALCULATE A(1, 2)
A121 = (ZR13 - ZRP12) * (ZR121 + Z13 / P13)
A122 = ZRP12 * R11 / P12
A_{12} = ZRP_{13}/P_{13}
A(1,2) = 2 \times (A_{12} + A_{13} - A_{123})/T
\text{TEMP} = Z_{12}/P_{123}
\begin{align*}
\text{C} \ & \text{CALCULATE A\{1,3\}} \\
A_{131} &= (ZRP_{12} - ZRP_{13})^2 \\
A_{132} &= ZRP_{12} \times R_{12}/P_{12} + ZRP_{13} \times R_{13}/P_{13} \\
A(1,3) &= (A_{131} - A_{132})/T
\end{align*}
\begin{align*}
\text{C} \ & \text{CALCULATE B\{1,1\}} \\
B\{1,1\} &= -(SSN \times PAPM_1 \times P_{23} \times ZRP_{12} \times PAPM_2 \times ZRP_{13} \times PAPM_3)/T
\end{align*}
\begin{align*}
\text{C} \ & \text{CALCULATE A\{2,1\}} \\
A_{211} &= (ZRP_{21} \times Z_{13}/P_{13}) \times (ZRP_{23} - ZRP_{21}) \\
A_{212} &= \text{TEMP} \times R_{21}^2 \\
A(2,1) &= (A_{211} + A_{212})/T
\end{align*}
\begin{align*}
\text{C} \ & \text{CALCULATE A\{2,2\}} \\
A_{221} &= \text{TEMP} \times PAPM_1 \times Z_{23} \times (1 - 2 \times \text{PAPM}_1)/P_{23} \\
A_{222} &= \text{TEMP} \times PAPM_2 \times Z_{13} \times (1 - 2 \times \text{PAPM}_2)/P_{13} \\
A_{223} &= \text{TEMP} \times (Z_{12} - 2.) \times \text{PAPM}_1 \times \text{PAPM}_2 \times P_{12} \\
A_{224} &= Z_{13} \times Z_{23} \times (P_{12} - 2. \times \text{PAPM}_1 \times \text{PAPM}_2)/(P_{13} \times P_{23}) \\
A(2,2) &= (\text{TEMP} \times A_{221} + A_{222} + A_{223})/T
\end{align*}
\begin{align*}
\text{C} \ & \text{CALCULATE A\{2,3\}} \\
A_{231} &= (ZRP_{12} \times Z_{23}/P_{23}) \times (ZRP_{13} - ZRP_{12}) \\
A_{232} &= \text{TEMP} \times R_{12}^2 \\
A(2,3) &= (A_{231} + A_{232})/T
\end{align*}
\begin{align*}
\text{C} \ & \text{CALCULATE B\{2,1\}} \\
B\{2,1\} &= \text{PAPM}_1 \times \text{PAPM}_2 \times (SSN \times \text{TEMP})/T
\end{align*}
\begin{align*}
\text{C} \ & \text{CALCULATE A\{3,1\}} \\
A_{311} &= (ZRP_{21} - ZRP_{23}) \times (1 - 2 \times \text{PAPM}_1)/P_{23} \\
A_{312} &= ZRP_{21} \times R_{21}/P_{12} \\
A_{313} &= ZRP_{23} \times R_{23}/P_{23} \\
A(3,1) &= (A_{311} - A_{312} - A_{313})/T
\end{align*}
\begin{align*}
\text{C} \ & \text{CALCULATE A\{3,2\}} \\
A_{321} &= (-ZRP_{12} - Z_{23}/P_{23}) \times (ZRP_{21} - ZRP_{23}) \\
A_{322} &= ZRP_{12} \times R_{21}/P_{12} \\
A_{323} &= ZRP_{23} \times P_{23} \\
A(3,2) &= 2 \times (A_{321} + A_{322} - A_{323})/T
\end{align*}
\begin{align*}
\text{C} \ & \text{CALCULATE A\{3,3\}} \\
A_{332} &= (ZRP_{12} \times Z_{23}/P_{23}) \times (1 - 2 \times \text{PAPM}_2)/P_{13} \\
A_{333} &= ZRP_{12} \times R_{12}/P_{12} \\
A_{334} &= Z_{23}/P_{23} \\
A(3,3) &= -(T + A_{332} - A_{333} - A_{334})/T
\end{align*}
\begin{align*}
\text{C} \ & \text{CALCULATE B\{3,1\}} \\
B\{3,1\} &= -(SSN \times \text{PAPM}_2 \times P_{13} \times ZRP_{21} \times \text{PAPM}_1 \times ZRP_{23} \times \text{PAPM}_3)/T
\end{align*}
SOLVE SYSTEM A*X=B FOR X. X IS VECTOR OF COVARIANCES C11, C12, C22

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(I)


334X, 3E20.8, 5X*, X2*, 3X*, **, E23.8/34X, 3E20.8, 5X*, X3*, 3X*, **, E23.8/)

GO TO 230

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

220 APMC11=B(1, 1)
221 APMC12=B(2, 1)
222 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, PAPM1, PAPM2, NTS


230 COVSKIP=1

RETURN

250 ISTOP=1

RETURN

END
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/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 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
5 25 PERCENT RELATIVE DIFFERENCE
9 IF (RELDIFF-0.25) 8, 8, 30
8 COUNTRD(8,J)=COUNTRD(8,J)+1.
C 20 PERCENT RELATIVE DIFFERENCE
10 IF (RELDIFF-0.20) 10,10,30
9 COUNTRD(9,J)=COUNTRD(9,J)+1.
C 15 PERCENT RELATIVE DIFFERENCE
12 IF (RELDIFF-0.15) 12,12,30
11 COUNTRD(11,J)=COUNTRD(11,J)+1.
C 10 PERCENT RELATIVE DIFFERENCE
14 IF (RELDIFF-0.10) 14,14,30
13 COUNTRD(13,J)=COUNTRD(13,J)+1.
C 5 PERCENT RELATIVE DIFFERENCE
16 IF (RELDIFF-0.05) 16,16,30
15 COUNTRD(15,J)=COUNTRD(15,J)+1.
C 1 PERCENT RELATIVE DIFFERENCE
   IF (RELDIFF-0.01) 10,18,30
10 COUNTRD(3,J)=COUNTRD(3,J)+1.
C 0.1 PERCENT RELATIVE DIFFERENCE
   IF (RELDIFF-0.001) 20,20,30
20 COUNTRD(2,J)=COUNTRD(2,J)+1.
C 0.01 PERCENT RELATIVE DIFFERENCE
   IF (RELDIFF-0.0001) 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),E(1),E(1),E(1)
   TPID=*F6.2*, IROBUST=*I2*, PPM1,2=*2F7.4*, PAPM1,2=*2F7.4*)
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
TERMCV=(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*TERM
XY = SUM OF N TERM*TERMCV
X = SUM OF N TERMCV
X2 = SUM OF N TERMCV*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)
XN=N*(N-2)
TO=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-TO+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*TO+B2*X2+B*T2*(MSE(1)-B*MSECV)+B2*T3-N*MSE(5)*MSE(5))/XN
C
RETURN
END
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 DEP(3,3), DQLU,3), E(4,3), IROBUST, NTS, P1, P2, P3, TPID
COMMON/DATA/XDATA(2), ZDATA(6)
COMMON/ITK/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, 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=II+ (MLE, PMDRO, APMRO, APMR1, PMDR2, APMR2), TPID=F4.2*, X(C, D)*)
59

Z=2*F7.2/5x* IROBUST=*I1* PEPMI2=2*F6.4* PMLE1=2=2*F6.4* PPM1,
32=2*F6.4* PAPM1=2=2*F6.4* Z=6*F4.01
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
C DIMENSION X(N)
C REAL LE,LH,M
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
SUBROUTINE BESTEST(BESTQL)

BY TWO DIFFERENT CRITERION (SUMMED ABSOLUTE RELATIVE DIFFERENCE RELD-- 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),RELQDL(4),RELDEP(3,3),RELQDL(4,3)
DIMENSION SEEP(3),SEQLU(4),W(4),X(3),Y(3)
COMMON DEP(3,3),QDL(4,3),E(4,3),IROBUST,NTS,P1,P2,P3,TPID
COMMON/BEST/BESTEP(3,2),CTRDIP,CTRDQDL(3),PREDIP(9,3),PRQDL(9,7),SBIASEP(3,3),SBIASEQL(3,7)
COMMON/DATA/XDATA(2),ZDATA(6)

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

FOR EPM 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

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 RESETTERICS.

DETERMINE SIGN OF FIRST COMPONENT OF ESTIMATOR

1 IF (ABS(DEP(I,1))<1.E-13) 3,5,3
  IF (DEP(I,1)<0.) 4,4,6
  NEGATIVE BIAS
  4 SBIASEP(I,1)=SBIASEP(I,1)+1.
  GO TO 10
  ZERO BIAS (CONSIDER ONLY 13 DECIMAL PLACES)
  5 SBIASEP(2,1)=SBIASEP(2,1)+1.
  GO TO 10
C POSITIVE BIAS
6 SBIASEP(3,1)=SBIASEP(3,1)*1.
10 CONTINUE
   CALL ASORT(X,1,3)
   DO 20 I=1,3
      IF (SEEPI.EQ.X(I)) BESTEP(I,1)=BESTEP(I,1)+1.
   20 CONTINUE
C DETERMINE WHETHER ANY PEPM COMPONENT IS ZERO
C
ICK=0
   IF (E(I,1)-1.E-10) 30,25,25
   25 IF (E(I,2)-1.E-10) 30,26,26
   26 IF (E(I,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)+RELDEP(I,J)
   34 CONTINUE
   35 CONTINUE
   CALL ASORT(Y,1,3)
   DO 40 I=1,3
      IF (Y(I).EQ.Y(I)) BESTEP(I,2)=BESTEP(I,2)+1.
   40 CONTINUE
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
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(I,I)=PRDEP(I,I)+1.
55 CONTINUE
IF (RELDEP(3,1)-0.15) 541,543,543
541 IF (RELDEP(3,2)-0.15) 546,543,543
543 PRINT 544, NTS, IROBUST, TPID, RELDEP(3,1), RELDEP(3,2), (E(I,J), J=1,2 I=1,4,3), (E(I,J), J=1,2, I=2,3), (DEP(I,J), J=1,2, I=1,3), XDATA(I I), XDATA(I, I=1,6)
544 FORMAT(* SUBR BESTEST, NTS=*I3, IROBUST=*I1, TPID=*F4.2, RELDEP(*1F7.4, *1F7.4, *1F7.4)
FOR QL COMPARISONS, ROUTINE IS CALLED FOR EACH OF THREE
ROBUSTNESS SETS

ROBUSTNESS SET 0. ORIGINAL PRIOR.

56 II=1
57 L=2
58 ICK=0

DETERMINE WHETHER ANY P COMPONENT IS ZERO

59 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=II,4
62 SEG(I)=0.
63 DO 610 J=1,3
64 V(I)*SEQL(I)=SEQL(I)*DQL(I,J)*DQL(I,J)
610 CONTINUE
615 IF (IR=2) 62,615,64
62 K=1
GO TO 65
K = 5
GO TO 65
K = I + 3
IF (ABS(DQL(I, 1)) - 1.E-13) 69, 69, 67
IF (DQL(I, 1) = 0.) 68, 68, 70
NEGATIVE BIAS
SBIASQL(1, K) = SBIASQL(1, K) + 1.
GO TO 75
ZERO BIAS (CONSIDER ONLY 13 DECIMAL PLACES)
SBIASQL(2, K) = SBIASQL(2, K) + 1.
GO TO 75
POSITIVE BIAS
SBIASQL(3, K) = SBIASQL(3, K) + 1.
CONTINUE
CALL ASORT(V, L)
DO 76 I = L, 4
IF (SEQL(I) = V(L)) BESTOL(I, 1) = BESTOL(I, 1) + 1.
CONTINUE
C UNSORT V FOR ROBUSTNESS SETS
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
CTRDLQL(IR) = CTRDLQL(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)
DO 79 I = L, 4
IF (RDQL(I) = W(L)) BESTOL(I, 2) = BESTOL(I, 2) + 1.
79 CONTINUE
C
C UNSORT W FOR ROBUSTNESS SETS
C
ROBUSTNESS SET 1. UNIFORM PRIOR.

C C SET POSTERIOR MODE EQUAL TO M.L.E.

DO 102 I=1,3
   DQL(3,I)=DQL(2,I)
102 CONTINUE
C 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: \( Y = \text{URAN}(X) \)

\( X \) 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 $_\leq$ 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

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

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></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
$\text{Det}(a) = 10^{-100 \times \text{ISCALE}} \times \text{DETERM}$ 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 $A(MAX,MAX)$, $B(MAX,M)$, $IPIVOT(MAX)$, 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 $L_n$, $L_{n-1}, \ldots, L_1$. 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 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.