SCHOOL OF OPERATIONS RESEARCH
AND INDUSTRIAL ENGINEERING
COLLEGE OF ENGINEERING
CORNELL UNIVERSITY
ITHACA, NY 14853-3801

TECHNICAL REPORT NO. 1095

May 1994
Revised June 1994

A COMPUTER PROGRAM FOR THE STATISTICAL ANALYSIS OF REPEATED EVENT DATA USING A MIXED EFFECTS REGRESSION MODEL[1]

by

R. Natarajan, B.W. Turnbull, E.H. Slate,
M.T. Wells, L.C. Clark, and H. Abu-Libdeh

[1]This research was supported principally by Grant R01 CA61120, also by grants R01 GM28364 and R01 CA 49764, all from the U.S. National Institutes of Health.

A COMPUTER PROGRAM FOR THE STATISTICAL ANALYSIS OF REPEATED EVENT DATA USING A MIXED EFFECTS REGRESSION MODEL

Ranjini NATARAJAN, Bruce W. TURNBULL, Elizabeth H. SLATE, Martin T. WELLS,

Cornell University, Ithaca NY 14853

Larry C. CLARK

University of Arizona

and

Hasan ABU-LIBDEH

Department of Mathematics, Birzeit University, West Bank, via Israel

This paper presents a computer program for fitting mixed effects regression models to repeated events data. The method has been described by Abu-Libdeh, Turnbull and Clark [1]. Such data can occur in longitudinal studies where subjects experience repeated events over time. The program allows the stepwise construction of a series of regression models which can be used to examine and test the influence of the various measured covariates upon the event rates. Two examples are provided. The first is a simple example involving the incidence of mammary tumors in rats. The second involves a very large complex data set from a clinical trial for the prevention of recurrent skin tumors.

Key Words: Multiple failure times, Covariates, Poisson regression, Random effects.

Corresponding Author: Bruce W. Turnbull,

227 ETC, School of Operations Research and Industrial Engineering,

Cornell University, Ithaca, New York 14850-3801, USA.

E-mail: turnbull@orie.cornell.edu

June 9, 1994

1 Introduction

The purpose of this paper is to present and describe a computer program which performs a stepwise (forward and/or backward) procedure for fitting regression models to repeated events data. Such data can occur in longitudinal studies, either experimental or observational, where individuals may experience repeated events (e.g. failures) over time. The events can be represented as a realization of a stochastic point process and some researchers have alternately described this situation as one with multiple failure time data. Individuals are assumed heterogeneous. Some of the heterogeneity is explained by measured covariates (e.g. treatment, age, gender); unexplained subject variability is modeled by using random effects. There are applications to many fields in biomedicine: in clinical studies, for example, of epileptic seizures, asthma, skin cancer (Abu-Libdeh, Turnbull and Clark [1]), gallstones (Thall [2]), bladder cancer (Freedman, Sylvester and Byar [3]); in animal carcinogenicity experiments to incidence of mammary tumors (Gail, Santner and Brown[4]). There are also applications in engineering to the reliability of repairable systems (Ascher and Feingold [5]) and in sociology to studies of mobility, unemployment, fertility, etc. (Allison [6]).

If at most one event (failure, death etc.) can occur to any one individual, then the survival analysis techniques such as exponential, Weibull or proportional hazards regression methods are applicable (see, e.g. Lawless 1982). For the situation where repeated events can occur over time in a single individual, as in some of the studies listed above, several statistical methods have been proposed by various researchers. This paper concerns a quite flexible parametric approach, based on a Poisson process mixture model, described by Lawless [8, Sec. 3] and later expanded by Abu-Libdeh et al. [3]. The latter applied it to data from an ongoing randomized multi-center clinical trial for the prevention of reoccurrence of skin cancer.

The methodology is described in Section 2. The computer program is described in Section 3. Some sample runs are described in Section 4. Hardware and software specifications are given in Section 5. The program is written in FORTRAN and versions are currently set up to run interactively on a SUN 4 workstation and on an IBM PC or compatible.

2 Description of the Methods

We suppose there are a total of I individuals or subjects. For individual i, $1 \leq i \leq I$, there are n_i events, possibly zero, over a period of observation T_i. The period of observation is the elapsed time from that individual's entry time on the study to the time last seen. These $\{T_i\}$ may vary from individual to individual. A set of p (≥ 1) "regressor", "explanatory" or "covariate" values, denoted by the row $p-$vector $\boldsymbol{x}_i$, is associated with each subject i, $1 \leq i \leq I$. These are the so-called "fixed effects" and might represent such variables as age, gender, treatment, smoking status etc.

The model specifies that the number of events n_i experienced by individual i, $1 \leq i \leq I$, is Poisson distributed with mean $\mu_i = \theta_i T_i \exp(\boldsymbol{x}_i\boldsymbol{\beta})$, i.e. proportional to the follow-up time T_i. Here $\boldsymbol{\beta}$ is a (column) $p-$vector of regression coefficients, representing the effect of the regressor variables, and θ_i is the baseline ($\boldsymbol{x}_i = \mathbf{0}$) value or "underlying" event rate for the ith subject. Equivalently, we may say that the events for individual i occur according as a Poisson process with event rate $\theta_i \exp(\boldsymbol{x}_i\boldsymbol{\beta})$. Another characterization is that the inter-event times are independent exponentially distributed random variables with means given by $[\theta_i \exp(\boldsymbol{x}_i\boldsymbol{\beta})]^{-1}$.

Subject heterogeneity (or "extra Poisson variation") that is not explained by the regressor variables is modeled by allowing the θ_i values to vary from subject to subject according to a gamma distribution with scale parameter γ and shape parameter ν (Johnson and Kotz, [9, p. 166, eqn (2)]). Thus unexplained patient variability is modeled as a random effect. The gamma mixing distribution is a convenient and flexible choice that has been used by Lawless [8] and others – see the remark by Abu-Libdeh et al. [3, p.1019]. The mean and variance of the mixing distribution are given by $\mu = \nu\gamma$ and $\nu\gamma^2$, respectively. Of special interest is the degenerate case when $\nu \longrightarrow \infty$ so that the θ_i are equal and constant and the baseline subject event rates can be considered homogeneous. We have constructed this model as a mixed (random and fixed) effects model; however it may be viewed in a number of ways, such as a negative binomial regression model, as a generalized linear model, or as a model with "extra-Poisson" variation (Lawless [8,10]).

Conditional on θ_i, the contribution to the likelihood for the ith subject is

$$L_i(\boldsymbol{\beta}|\theta_i) = \mu_i^{n_i} e^{-\mu_i} / n_i!$$

where $\mu_i = \theta_i T_i \exp(\boldsymbol{x}_i\boldsymbol{\beta})$,

The full likelihood is then given by

$$L(\gamma,\nu,\boldsymbol{\beta}) = \prod_{i=1}^{I} \int_0^\infty L_i(\boldsymbol{\beta}|\theta_i)\, g(\theta_i;\gamma,\nu)\, d\theta_i$$

where $g(\theta;\gamma,\nu) = \gamma^{-\nu}\theta^{\nu-1}\exp(-\theta/\gamma)/\Gamma(\nu)$ is the gamma density of the mixing distribution.

Carrying out the integration we obtain;

$$L(\gamma,\nu,\boldsymbol{\beta}) = \prod_{i=1}^{I} \frac{\Gamma(n_i+\nu)}{\Gamma(\nu)} \frac{[\gamma\exp(\boldsymbol{x}_i\boldsymbol{\beta})]^{n_i}}{[\gamma T_i \exp(\boldsymbol{x}_i\boldsymbol{\beta})+1]^{n_i+\nu}} \tag{1}$$

This likelihood, in a slightly different form, is the same as that given by Lawless (1987b, p.210). It is also a special case of Abu-Libdeh et al. (1990, Eqn (1)). For numerical reasons, it is easier to reparametrize and work with parameters $\mu = \nu\gamma$ and ν rather than γ and ν. Replacing γ by μ/ν in (1), we write the resulting likelihood as $L(\mu,\nu,\boldsymbol{\beta})$

Estimates of the mixing distribution parameters μ, ν (equivalently γ, ν) and of the regression coefficients $\boldsymbol{\beta} = (\beta_1,\ldots,\beta_p)$ can be found by maximizing the likelihood (1), or equivalently its logarithm $\log L = \mathcal{L}$, say, with respect to these variables. The maximizing values are the maximum likelihood estimates. The variance-covariance matrix of the estimates is obtained by evaluating the inverse of the information matrix at the estimated parameter values. The information matrix is given by the negative of the matrix of second derivatives (Hessian) of $\mathcal{L}(\mu,\nu,\boldsymbol{\beta})$. The approximate standard errors of the estimates are then given by the square roots of the diagonal entries of the variance-covariance matrix. This leads to construction of confidence intervals for each of the parameters estimated and to the Wald statistic for testing the significance of each of the estimates. Alternatively, a score statistic, based on the gradient of $\mathcal{L}$, or a likelihood ratio statistic can be used to test the significance of a regression coefficient – see Rao [11, Sec. 6e.3] or Cox and Hinkley [12].

3 Program Description

A flow chart for the main program is given in Figure 1. Some sample runs are described in the next section. The program consists of four parts (1) Data input; (2) Model specification; (3) Likelihood maximization; and (4) Output of parameter estimates, standard errors and test statistics. The user has the option then to return to step (2) to specify an enlarged set of regressor variables to include in the model. Thus a preferred model can be constructed in a forward stepwise manner by repeating steps (2)–(4).

3.1 Step 1: Data input

The user is first prompted for the name DICTION of the "dictionary" file. The first line of this file must contain the name of the DATA file. The second line contains the values of NVARS+1 (ID and the number of regressor variables), and NSUBS, the number of subjects (I). Lines 3 through NVARS+3 contain the descriptive names of the record identifier (ID), the number of events, the observation time, and each of the regressor variables (e.g. "age", "treatment", "smoking status" etc. in the example of Section 4.2). The dictionary file for the small example of Section 4.1 is shown in Table 1.

The DATA file named in the first line of the DICTION file is now read in. This is a rectangular file with NSUBS rows and NVARS+3 columns in free format. In the ith row, which corresponds to subject i, the first entry is its name or "ID". This item is not actually used by the program, but its presence is useful to detect errors in data entry. The second entry in the row is the number of events n_i experienced by the ith subject; the third item is the observation, followup or exposure time T_i for that subject. Entries 4 through NVARS+3 are the covariate values for subject i in the same order as that specified in the dictionary file. Missing values are designated by the code "-999". Missing values in a regressor variable cause the corresponding records to be ignored for models including this variable. Missing (-999) or negative values for the number of events and/or observation time cause the entire record to be ignored.

Finally the user is prompted for the name of the file to which output is to be written.

Optionally, the program will print out the first five records of the DATA file read, so that the user can verify that the correct values have been entered. Summary statistics (mean, min, max, number missing) are printed for each of the variables, and these can also serve as a useful check.

3.2 Step 2: Model specification

The user is prompted to specify the model to be used in the analysis. The number and names of the regressor variables to be included are entered. In addition, as long as two or more regressor variables are specified, the user is asked which 2-way interaction terms, if any, between the variables are to be included in the model. For each interaction a new covariate is formed by multiplying together the two corresponding regressor variable values for each subject in turn. Any subject who has a missing value for one or more of the included covariates will be excluded from the analysis of this model. The value of p as defined in Section 2, is then the number of variables included in the model plus the number of interactions specified (if any).

To aid in the specification of the model, at any stage all previous models fitted are listed. The user can choose any of these (including the "null" model in which no covariates are included) as a "base" model and only the additional covariates and interactions that are in the desired model but not in the base model need be specified. All covariates in the base model are automatically included.

3.3 Likelihood maximization

The NAG [13] subroutine E04UCF, which uses a sequential quadratic programming method for optimizing a nonlinear function, is employed to maximize the log-likelihood $\mathcal{L}$. The subroutine may be replaced by a Newton-Raphson or other similar algorithm; however the authors have had good experience with the one recommended. (The older NAG routine E04VDF can also be substituted.) Initially, starting values for μ and ν are obtained by the method of moments, with $\boldsymbol{\beta} = \mathbf{0}$. If, for any reason, the algorithm fails to converge to the optimum, an appropriate error message is displayed. In our experience with several data sets, large and small, we have not encountered any problems with non-convergence. (The program does however impose working lower bounds of 10^{-6} on the values of μ and ν which helps avoid possible problems.)

3.4 Step 4: Output of parameter estimates, standard errors and test statistics

After successful convergence, the program prints out the following results both to the screen and to the designated output file:

- The maximum likelihood estimates (MLEs), $\hat{\nu}$ and $\hat{\mu}$, of the shape parameter ν and the mean $\mu = \nu\gamma$ of the mixing distribution along with their standard errors.
- The deviance, i.e. $-2\mathcal{L}(\hat{\mu}, \hat{\nu}, \hat{\boldsymbol{\beta}})$. For comparison, the deviance of the null model (i.e. that without any covariates $\boldsymbol{\beta} = \mathbf{0}$) is also displayed.

- The likelihood ratio statistic which is the difference of the two deviances in the above item. The degrees of freedom are equal to p, the number of covariates fitted (including any interaction terms). By comparison with the percentage points of the chi-squared distribution with p degrees of freedom, the overall significance of the fitted covariates can be assessed. Similarly if two models have been fitted, with p_1 and p_2 degrees of freedom ($p_1 < p_2$) with one nested within the other, the significance of the improvement in fit by including the extra covariates in the second model can be assessed by comparing the difference in the respective deviances with percentage points of the chi-squared distribution with $p_2 - p_1$ degrees of freedom.

- The score test statistic. This can be used in a similar fashion as the likelihood ratio statistic to assess the overall significance of the fitted covariates.

- For each covariate (and interaction) included in the model, the program displays the estimated regression coefficient, its standard error, its standardized Z-value and a 95% confidence interval for it.

- Finally, the estimated correlation matrix for the parameter estimates $(\hat{\mu}, \hat{\nu}, \hat{\boldsymbol{\beta}})$ is written to the output file only. This is obtained from the covariance matrix which is the inverse of the information matrix as explained in Section 2.

The significance tests and confidence intervals are based on the asymptotic normal approximation. The user is then asked if he wants fit a new model with added covariates or interactions. If yes, a list of the last 10 models previously fitted with corresponding deviance values are printed to aid in the choice of variables to include. Steps 2–4 are then repeated.

4 Sample runs

Sample runs for two data sets are described. The first is a rather simple one with a small sample size, a common followup time for all subjects and a single binary covariate. The second data set is a very large one with many covariates and varying followup times. The terminal sessions for the PC version are given in Appendices A and B, respectively. Sessions for the UNIX version are almost identical.

4.1 Mammary tumor rat data

Gail, Santner and Brown [4] presented data on times to mammary tumor for 23 female rats on retinyl acetate treatment and 25 female control rats. The DICTIONary file appears in Table 1 and the DATA file in Table 2. This data set has become a standard one and has been used by several authors, including most recently Sinha [14]. Randomization occurred on day 60 of the study and the study ended on day 182. Thus the followup time T_i for each rat was 122 days. Gail et al. [4] give the actual times of each incident mammary tumor, but we only use the number of tumors occurring between days 60 and 182. There is only one covariate ($p = 1$), namely treatment ($x = 1$ for retinyl, $x = 0$ for control). The sample run is displayed in Appendix A. The null model omitting the treatment covariate is fitted first, then the full model is fitted, using the covariate. It can be seen from the output that the effect of the retinoid treatment is to significantly reduce the incidence of mammary tumors.

4.2 Skin cancer prevention clinical trial data

Clark et al. [15] describe an ongoing double blind multi-center randomized clinical trial to test the benefits of a nutritional supplement of selenium for the prevention of recurrent skin cancer. The trial has also been described by Abu-Libdeh et al. [3]. Starting in 1983, entering patients have been randomized to either selenium (Se) supplementation or to placebo. We use interim data from this trial to illustrate our program. There are 1277 patients. The response from each patient consists of a list of times (measured from date of entry) at which new squamous cell carcinomas (SCCs) were diagnosed. The length of followup varies from patient to patient – the average followup being about five years. Besides treatment assignment, a number of baseline covariates were collected from each patient at the time of entry. These included plasma selenium level in mg/dl (Se1), age, gender, weight (in kg.), sun damage, previous number of SCCs diagnosed **before** entry (#Prev_SCC), and the attending clinic. A more complete description is given in Abu-Libdeh et al. (1990, Sec.4) – see also the "Quick Data Summary" table which is part of the output from the sample run in Appendix B. The null model, without covariates, is first run. We next ran a model with covariate "baseline Se level", next a model with "age" and "gender" added; finally a model which additionally included an interaction between age and gender. We see that baseline plasma selenium level and gender are significant. Increased selenium level is protective, and males are at higher risk. (Note here that a principal use of the program would be to assess the significance of the covariate "treatment" after adjusting for the important other covariates. However, because this is only an interim analysis of an ongoing blinded trial, this cannot be done here – see the article by Green, Fleming and O'Fallon [16] on interim reporting of results from clinical trials.)

5 Hardware and Software Specifications

The program was written using FORTRAN-77 for a SUN SPARCstation [17]. CPU time for the example of Section 4.1 was 0.23 secs on a SPARC 10, 0.63 secs on a SPARC 2, and 1.47 secs on a SPARC 1. The corresponding CPU times for the example of Section 4.2, with its multiple models fitted, were 22.07, 77.37 and 190.77 seconds. A Microsoft FORTRAN PowerStation [18] program has also been compiled to run on a 386 or 486 IBM PC or compatible.

6 Mode of Availability of the Program

Copies of the program are available upon request from the corresponding author, Bruce W. Turnbull.

7 Acknowledgement

This research was supported principally by grant R01 CA61120, also by grants R01 GM28364 and R01 CA 49764, all from the U.S. National Institutes of Health.

Appendix A. Sample terminal session – Mammary tumor rat data

```
    ---------------------------------------------------------
    MIXED EFFECTS POISSON REGRESSION MODEL FOR REPEATED EVENTS
    ---------------------------------------------------------

WHAT IS THE DICTIONARY FILE?      rat.dct

SPECIFY FILE FOR PRINTED OUTPUT:  rat.out

READING INPUT DATA...PLEASE WAIT

     48 RECORDS ARE BEING READ

  0 RECORDS ARE BEING IGNORED DUE TO MISSING NUMBER OF EVENTS

  AND/OR EXPOSURE TIME

PRINT FIRST FIVE RECORDS? (Y/N):    y

RECORD  1

    1.0000    122.0000    1.0000

RECORD  2

    0.0000    122.0000    1.0000

RECORD  3

    2.0000    122.0000    1.0000

RECORD  4

    1.0000    122.0000    1.0000

RECORD  5

    4.0000    122.0000    1.0000


QUICK DATA SUMMARY:         MEAN       MIN       MAX      # MISSING

        N_Tumors            4.417       .0       13.0         0.

        Exp_Time          122.000    122.0      122.0         0.

        TREATMENT           1.521      1.0        2.0         0.


DO YOU WISH TO CONTINUE? (Y/N)    y


MAXIMIZING THE LIKELIHOOD FUNCTION FOR THE MODEL WITHOUT ANY COVARIATES

(NULL MODEL).....PLEASE WAIT


SUCCESSFUL CONVERGENCE HAS BEEN REACHED AFTER 15 ITERATIONS
```

```
                STATISTICAL OUTPUT BASED ON 48 RECORDS
                PARAMETERS OF THE MIXING DISTRIBUTION
            SHAPE  =     2.29555         MEAN    =    0.036202
            ST.DEV =  (  0.73475)        ST.DEV  = (  0.004252)
         DEVIANCE FOR NULL MODEL (-2*(LOG-LIK)) =   1794.62442

DO YOU WISH TO ADD COVARIATES TO THE MODEL? (Y/N)   y
INDEXED LIST OF COVARIATES
   1: TREATMENT
HOW MANY COVARIATES DO YOU WISH TO USE? (1 --  1)   1
ENTER INDEX OF CHOSEN COVARIATES.                   1
-----OPTIMIZING THE LIKELIHOOD FUNCTION-----
SUCCESSFUL CONVERGENCE HAS BEEN REACHED AFTER 15 ITERATIONS
                STATISTICAL OUTPUT BASED ON 48 RECORDS
                PARAMETERS OF THE MIXING DISTRIBUTION
            SHAPE =     3.75301         MEAN    =    0.009546
            ST.DEV = (  1.51572)        ST.DEV = ( 0.003431)
      DEVIANCE: CURRENT MODEL     = 1781.46912   NULL = 1794.62442
      LIK. RATIO TEST STATISTIC   =   13.15530   WITH   1  DEGREE(S) OF FREEDOM.
      SCORE TEST STATISTC         =   24.02362   WITH   1  DEGREE(S) OF FREEDOM.

      COVARIATE        M.L.E.      ST.DEV.     Z-VALUE     95% C. INTERVAL
      ---------     ---------    ---------     -------    -----------------
    TREATMENT       0.823024*    0.212742      3.8687     (  0.406 ,   1.240)

      *: SIGNIFICANT AT THE  5%  LEVEL

DO YOU WISH TO ADD COVARIATES TO THE MODEL? (Y/N)  n
```

Appendix B. Sample terminal session – Clinical trial data

```
     ----------------------------------------------------------

     MIXED EFFECTS POISSON REGRESSION MODEL FOR REPEATED EVENTS

     ----------------------------------------------------------

WHAT IS THE DICTIONARY FILE?     skin.dct

SPECIFY FILE FOR PRINTED OUTPUT: skin.out

READING INPUT DATA...PLEASE WAIT

  1277 RECORDS ARE BEING READ.

  0 RECORDS ARE BEING IGNORED DUE TO MISSING NUMBER OF EVENTS

  AND/OR EXPOSURE TIME

PRINT FIRST FIVE RECORDS? (Y/N):  y

RECORD  1

    0.0000  2166.0000    2.0000    0.0000    1.0000    0.0000   65.9000

    1.0000     4.0000    6.0000    0.0000    5.0000  122.0000    3.0000

    0.0000     0.0000    1.0000


RECORD  2

    0.0000   169.0000    2.0000    1.0000    2.0000    0.0000   62.3000

    1.0000     4.0000    5.0000    1.0000    7.5000  104.0000    2.0000

    0.0000     1.0000    1.0000


RECORD  3

    0.0000   967.0000    1.0000    1.0000    1.0000    1.0000   90.9000

    0.0000     6.0000    5.0000    0.0000    6.0000   84.0000    0.0000

    1.0000     1.0000    1.0000


RECORD  4

    0.0000  2061.0000    2.0000    1.0000    1.0000    0.0000  102.3000

    0.0000     5.0000    5.0000    0.0000    5.5000  107.0000    0.0000

    1.0000     1.0000    1.0000
```

```
RECORD   5
    0.0000  1716.0000     2.0000     0.0000     1.0000     0.0000    70.9000
    0.0000     4.0000     5.0000     0.0000     5.5000   101.0000    11.0000
    0.0000     0.0000     1.0000

QUICK DATA SUMMARY:      MEAN      MIN      MAX    # MISSING
        N_Tumors          .538       .0     13.0        0.
        Exp_Time      1593.545      1.0   3276.0        0.
        T_Group       1.501         1.0      2.0        0.
        AGE>64         .525          .0      1.0        0.
        GENDER M=1    1.255         1.0      2.0        0.
        Cur_Smoker    0.284          .0      1.0        0.
        Weight_Kg    77.338        40.5    143.2        1.
        Farm_GE_15    0.344          .0      1.0        0.
        Sunburn(a)    4.741         4.0      6.0        0.
        Suntan(b)     4.836         4.0      6.0        0.
        Sunscrn<5     0.282          .0      1.0        0.
        Sun_Damage    4.733         1.0      9.0        0.
        Se1         114.243        45.0    220.0        0.
        #Prev_BCE     2.898          .0     66.0        0.
        #Prev_SCC     0.979          .0     16.0        0.
        In_Patient    0.458          .0      1.0        0.
        Clinic_#      3.165         1.0      8.0        0.

DO YOU WISH TO CONTINUE? (Y/N)  y
MAXIMIZING THE LIKELIHOOD FUNCTION FOR THE MODEL WITHOUT ANY COVARIATES
(NULL MODEL).....PLEASE WAIT
SUCCESSFUL CONVERGENCE HAS BEEN REACHED AFTER 17 ITERATIONS
                    STATISTICAL OUTPUT BASED ON 1277 RECORDS
                    PARAMETERS OF THE MIXING DISTRIBUTION
                SHAPE  =     0.36653           MEAN    =     0.000342
                ST.DEV = (   0.03779)          ST.DEV = (   0.000021)
             DEVIANCE FOR NULL MODEL (-2*(LOG-LIK)) =  11880.57057
```

```
DO YOU WISH TO ADD COVARIATES TO THE MODEL? (Y/N) y
INDEXED LIST OF COVARIATES
   1: T_Group       2: AGE>64       3: GENDER M=1    4: Cur_Smoker    5: Weight_Kg
   6: Farm_GE_15    7: Sunburn(a)   8: Suntan(b)     9: Sunscrn<5    10: Sun_Damage
  11: Se1          12: #Prev_BCE   13: #Prev_SCC    14: In_Patient   15: Clinic_#
HOW MANY COVARIATES DO YOU WISH TO USE? (1 -- 15)  1
ENTER INDEX OF CHOSEN COVARIATES:    11
-----OPTIMIZING THE LIKELIHOOD FUNCTION-----  (PLEASE WAIT)
SUCCESSFUL CONVERGENCE HAS BEEN REACHED AFTER  30 ITERATIONS
                    STATISTICAL OUTPUT BASED ON 1277 RECORDS
                    PARAMETERS OF THE MIXING DISTRIBUTION
                SHAPE =     0.37118            MEAN    =    0.000646
                ST.DEV = (  0.03844)           ST.DEV = (   0.000209)
      DEVIANCE: CURRENT MODEL     =11876.57471   NULL =11880.57057
      LIK. RATIO TEST STATISTIC   =     3.99586   WITH    1  DEGREE(S) OF FREEDOM.
      SCORE TEST STATISTIC         =     4.03472   WITH    1  DEGREE(S) OF FREEDOM.

      COVARIATE        M.L.E.       ST.DEV.     Z-VALUE     95% C. INTERVAL
      ---------      ---------    ---------    -------    -----------------
    Se1             -0.005638*    0.002823    -1.9967    ( -0.011 ,   0.000)

      *: SIGNIFICANT AT THE  5%  LEVEL

DO YOU WISH TO ADD COVARIATES TO THE MODEL? (Y/N)    y
INDEXED LIST OF COVARIATES
   1: T_Group       2: AGE>64       3: GENDER M=1    4: Cur_Smoker    5: Weight_Kg
   6: Farm_GE_15    7: Sunburn(a)   8: Suntan(b)     9: Sunscrn<5    10: Sun_Damage
  11: Se1          12: #Prev_BCE   13: #Prev_SCC    14: In_Patient   15: Clinic_#

PREVIOUS MODEL(S) FITTED
 MDL  0: NULL MODEL
         DEVIANCE:     11880.57057
 MDL  1: Se1
         DEVIANCE:     11876.57471
```

```
WHICH MODEL DO YOU WISH TO USE AS A BASE?    1
HOW MANY NEW VARIABLES DO YOU WISH TO ADD?   2
ENTER INDEX OF CHOSEN COVARIATES:            2 3
HOW MANY NEW INTERACTIONS? (0=none)          0
-----OPTIMIZING THE LIKELIHOOD FUNCTION-----  (PLEASE WAIT)
SUCCESSFUL CONVERGENCE HAS BEEN REACHED AFTER   44 ITERATIONS
                    STATISTICAL OUTPUT  BASED ON 1277 RECORDS
                    PARAMETERS OF THE MIXING DISTRIBUTION
                SHAPE  =    0.42968             MEAN   =    0.001761
                ST.DEV = (  0.04649)            ST.DEV = (  0.000715)
      DEVIANCE: CURRENT MODEL     =11829.50071   NULL =11880.57057
      LIK. RATIO TEST STATISTIC   =   51.06986   WITH   3  DEGREE(S) OF FREEDOM.
      SCORE TEST STATISTIC         =  60.78872   WITH   3  DEGREE(S) OF FREEDOM.

      COVARIATE       M.L.E.      ST.DEV.     Z-VALUE     95% C. INTERVAL
      ---------     ---------    ---------    -------    -----------------
    Se1             -0.007171*   0.002801    -2.5602     ( -0.013 , -0.002)
    AGE>64           0.440036*   0.122981     3.5781     (  0.199 ,  0.681)
    GENDER M=1      -0.918275*   0.159514    -5.7567     ( -1.231 , -0.606)

      *: SIGNIFICANT AT THE  5%  LEVEL

DO YOU WISH TO ADD COVARIATES TO THE MODEL? (Y/N)  y
INDEXED LIST OF COVARIATES
   1: T_Group       2: AGE>64       3: GENDER M=1    4: Cur_Smoker    5: Weight_Kg
   6: Farm_GE_15    7: Sunburn(a)    8: Suntan(b)     9: Sunscrn<5    10: Sun_Damage
  11: Se1          12: #Prev_BCE    13: #Prev_SCC    14: In_Patient   15: Clinic_#

PREVIOUS MODEL(S) FITTED
 MDL  0: NULL MODEL
         DEVIANCE:     11880.57057
```

```
MDL  1: Se1
        DEVIANCE:     11876.57471

 MDL  2: Se1       AGE>64    GENDER M=1
        DEVIANCE:     11829.50071

WHICH MODEL DO YOU WISH TO USE AS A BASE?    2
HOW MANY NEW VARIABLES DO YOU WISH TO ADD?   0
HOW MANY NEW INTERACTIONS? (0=none)          1
ENTER PAIR OF INDICES FOR ONE INTERACTION    2 3
-----OPTIMIZING THE LIKELIHOOD FUNCTION-----  (PLEASE WAIT)
SUCCESSFUL CONVERGENCE HAS BEEN REACHED AFTER  45 ITERATIONS
                    STATISTICAL OUTPUT BASED ON 1277 RECORDS
                    PARAMETERS OF THE MIXING DISTRIBUTION
                SHAPE  =    0.43065            MEAN   =    0.002174
                ST.DEV = (  0.04662)           ST.DEV = (  0.000999)
      DEVIANCE: CURRENT MODEL     =11828.51503  NULL =11880.57057
      LIK. RATIO TEST STATISTIC  =   52.05554  WITH   4  DEGREE(S) OF FREEDOM.
      SCORE TEST STATISTIC        =   62.55758  WITH   4  DEGREE(S) OF FREEDOM.

      COVARIATE      M.L.E.      ST.DEV.     Z-VALUE    95% C. INTERVAL
      ---------    ---------    ---------    -------   -----------------
    Se1            -0.007165*   0.002801    -2.5582    ( -0.013 , -0.002)
    AGE>64          0.065689    0.397167     0.1654    ( -0.713 ,  0.844)
    GENDER M=1     -1.096237*   0.242267    -4.5249    ( -1.571 , -0.621)
    AGE>*GEND       0.316835    0.320077     0.9899    ( -0.311 ,  0.944)

      *: SIGNIFICANT AT THE  5% LEVEL

DO YOU WISH TO ADD COVARIATES TO THE MODEL? (Y/N)   n
```

References

[1] H. Abu-Libdeh, B.W. Turnbull and L.C. Clark, Analysis of multi-type recurrent events in longitudinal studies; application to a skin cancer prevention trial, Biometrics 46 (1990) 1017-1034.

[2] P.F. Thall, Mixed Poisson likelihood regression models for longitudinal interval count data, Biometrics 44 (1988) 197-209.

[3] L. Freedman, R. Sylvester and D.P. Byar, Using permutation tests and bootstrap confidence limits to analyze repeated events data from clinical trials, Controlled Clinical Trials 10, 129-141.

[4] M.H. Gail, T.J. Santner and C.C. Brown, An analysis of comparative carcinogenesis experiments based on multiple times to tumor, Biometrics 36 (1980) 255-266.

[5] H. Ascher and H. Feingold, Repairable Systems Reliability (Marcel Dekker, New York, NY, 1984).

[6] P.D. Allison, Event History Analysis: Regression for Longitudinal Data (Sage Publications, Beverly Hills, CA, 1984).

[7] J.F. Lawless, Statistical Models and Methods for Lifetime Data (Wiley, New York, NY, 1982).

[8] J.F. Lawless, Regression methods for Poisson process data, Journal of the American Statistical Association 82 (1987) 808-815.

[9] N.L. Johnson and S. Kotz, Distributions in Statistics: Continuous Univariate Distributions I (Houghton Mifflin, Boston, MA, 1970).

[10] J.F. Lawless, Negative binomial regression models, Canadian Journal of Statistics 15 (1987) 209-225.

[11] C.R. Rao, Linear Statistical Inference and its Applications (Wiley, New York, NY, 1973)

[12] D.R. Cox and D.V. Hinkley, Theoretical Statistics (Chapman and Hall, London, UK, 1974).

[13] The Numerical Algorithms Group, The NAG Foundation Library Handbook (NAG, Oxford UK, 1992)

[14] Sinha, D., Semiparametric Bayesian analysis of multiple event time data, Journal of the American Statistical Association 88 (1993) 979-983.

[15] L.C. Clark, B.H. Patterson, D.L. Weed, and B.W. Turnbull, Design issues in cancer chemoprevention trials using micronutrients: application to skin cancer, Cancer Bulletin 43 (1991) 519-524.

[16] S.J. Green, T.R. Fleming and J.R. O'Fallon, Policies for study monitoring and interim reporting of results, Journal of Clinical Oncology 5 (1987) 1477-1484.

[17] Sun Microsystems, Inc., Mountain View, California.

[18] Microsoft FORTRAN PowerStation, Microsoft Corp., Redmond, Washington.

CAPTIONS FOR TABLES AND FIGURE

Table 1: Dictionary File (rat.dct) for Mammary Tumor Rat Example

Table 2: Data File (rat.dat) for Mammary Tumor Rat Example

Figure 1: Flow Chart

Table 1: Dictionary File (rat.dct) for Mammary Tumor Rat Example

```
rat.dat

2 48

ID

N_Tumors

Exp_Time

TREATMENT
```

Table 2: Data File (rat.dat) for Mammary Tumor Rat Example

ID	No. of tumors	Observation time	Treatment group
1	1	122	1
2	0	122	1
3	2	122	1
4	1	122	1
5	4	122	1
6	3	122	1
7	6	122	1
8	1	122	1
9	1	122	1
10	5	122	1
11	2	122	1
12	1	122	1
13	5	122	1
14	2	122	1
15	3	122	1
16	4	122	1
17	5	122	1
18	5	122	1
19	1	122	1
20	2	122	1
21	6	122	1
22	0	122	1
23	1	122	1
24	7	122	2
25	11	122	2
26	9	122	2
27	2	122	2
28	9	122	2
29	4	122	2
30	6	122	2
31	7	122	2
32	6	122	2
33	1	122	2
34	13	122	2
35	2	122	2
36	1	122	2
37	10	122	2
38	4	122	2
39	5	122	2
40	11	122	2
41	11	122	2
42	9	122	2
43	12	122	2
44	1	122	2
45	3	122	2
46	1	122	2
47	3	122	2
48	3	122	2

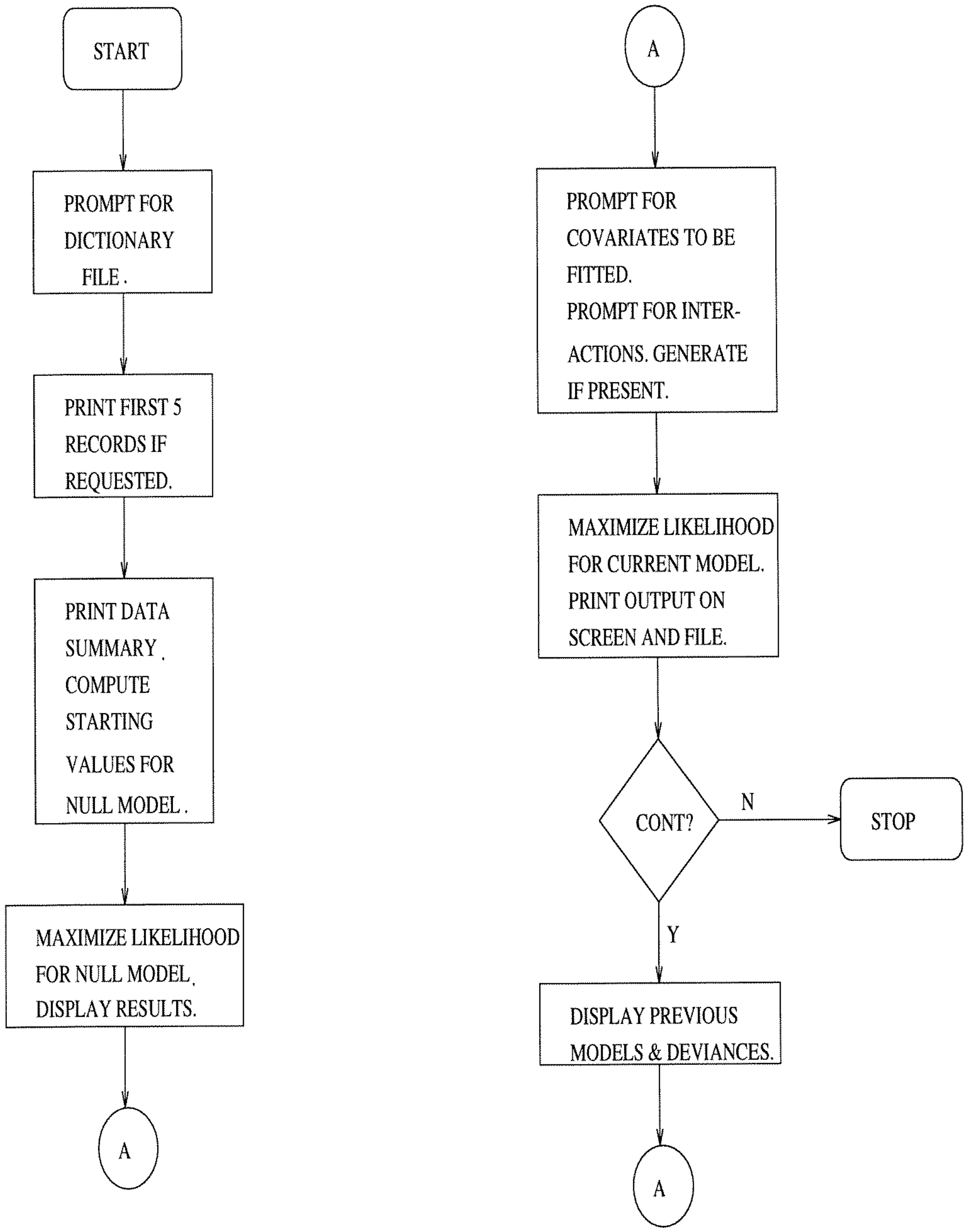

Figure 1: Flow Chart for the Program

```
C       * ****************  MIXED POISSON REGRESSION MODEL FOR ************
C       ******************     REPEATED EVENT DATA            ************


C       THE PROGRAM CALCULATES:
C 1.  MLEs AND SDs OF MIXING DISTRIBUTION PARMS
C 2.  MLEs AND SDs OF REGRESSION PARMS
C 3.  Z-VALUES FOR REGRESSION PARMS
C 4.  A NORMAL-BASED 95% CI FOR EACH REGRESSION PARM
C 5.  L.R. AND SCORE TEST STATISTICS FOR REGRESSION PARMS.
C 6.  CORRELATION MATRIX.

C     SUBROUTINES CALLED IN THE PROGRAM:

C     PARTIII:   EVALUATES SCORE VECTOR AND INFO-MATRIX
C     SCORE:     EVALUATES SCORE STATISTIC FOR H(0): REG. COEFFS = ZERO.
C     ARINV:     INVERTS HESSIAN MATRIX.
C     E04VDF:    THIS NAG ROUTINE MAXIMIZES THE LOG-LIKELIHOOD FUNCTION.
C     E04VDM:    THIS NAG SUBROUTINE IS CALLED BY E04VDF.
C     OBJFUN:    THIS SUBROUTINE COMPUTES THE OBJECTIVE FUNCTION FOR THE
C                NAG OPTIMIZATION FUNCTION FOR THE CURRENT MODEL
C     OBJNULL:   THIS SUBROUTINE COMPUTES THE OBJECTIVE FUNCTION FOR THE
C                NAG OPTIMIZATION FUNCTION FOR THE NULL MODEL


C                    DEFINITIONS OF IMPORTANT PROGRAM VARIABLES
C     A(,)         : NEGATIVE OF SAMPLE INFORMATION MATRIX.
C     ALERT        : 1 IF NUMBER OF SUBJECTS HAS BEEN REDUCED DUE TO
C                  : SOME MISSING OBSERVATION
C     ANS          : CHARACTER VARIABLE THAT INDICATES WHETHER USER
C                    USER WANTS TO CONTINUE

C     BASE         : BASE MODEL THAT USER WANTS TO BUILD ON
C     BASENULL()   : CONTAINS MLE OF BASE MODEL
C     C1,C2        : CONTAIN INFORMATION ON NUMBER OF VARIABLES IN BASE MODEL
C     CHI          : CHI SQUARE STATISTIC FOR TESTING SIGNIFICANCE OF
C                  : OF THE REGRESSION
C     COL(25)      : STORAGE FOR COVARIATE INDICES ENTERING ANALYSIS
C     COVNAME()    : NAMES OF COVARIATES IN DATA FILE
C     COVRT(  )    : NAMES OF COVARIATES AVAILABLE FOR ANALYSIS.
C     COVTEMP      : NUMBER OF COVARIATES ENTERING ANALYSIS
C     DICTION      : DICTIONARY FILE NAME
C     DLGLK        : DEVIANCE FOR NULL MODEL
C     EPS          : RADIUS OF CONVERGENCE
C     FLAG()       : 1 IF ANY MISSING ENTRIES FOR ANY COVARIATE

C     GMA(3)       : NAMES OF BASELINE PARAMETERS.
C     GMEAN        : MEAN NUMBER OF TUMORS (USED IN MOM ESTIMATION
C     ICOL         : > 0 IF ANY MODEL OTHER THAN THE NULL HAS BEEN FITTED
C     ID           : PATIENT DESCRIPTOR
C     INPUTF       : FILE CONTAINING COVARIATE DATA.
C     INTERDEX(,)  : INDICES OF VARIABLES APPEARING IN INTERACTIONS
C     LIKRATIO     : ARRAY OF DEVIANCES OF PREVIOUS MODELS
C     M            : NUMBER OF PARAMETERS TO BE ESTIMATED.
C     MAT(,)       : WORKING MATRIX FOR THE SUBROUTINE ARINV.
C     MAXX()       : ARRAY OF MAXIMUM VALUES OF EACH COVARIATE
C     MDL          : MODEL CHOICE POINTER (=2)
C     MINN()       : ARRAY OF MINIMUM OF EACH COVARIATE
C     MISSING()    : ARRAY OF NUMBER MISSING FOR EACH COVARIATE
C     MODINT(,)    : ARRAY OF PREVIOUS INTERACTIONS FITTED
C     MODVAR(,)    : ARRAY OF PREVIOUS COVARIATES FITTED
C                  : THE SIZE OF MODVAR AND MODINT LIMITS ONE TO 25 MODELS.

C     NCOV         : NUMBER OF COVARIATES ENTERING ANALYSIS.
C     NINTS        : NUMBER OF PAIRWISE-INTERACTIONS IN THE ANALYSIS
C     NRUN         : NUMBER OF MODELS ALLOWED IN ONE SESSION (25)
C     NUMMISS      : NUMBER MISSING IN ANY ANALYSIS
C     NVARS        : NUMBER OF COVARIATES IN THE DATA FILE INPUTF
C     OBJF         : VALUE OF LOG-LIKELIHOOD FUNCTION.
C     P(,)         : WORKING MATRIX.
C     PARMBASE(,)  : STORAGE OF PARM ESTIMATES OF MIXING DIST
C                  : COEFFICIENTS FOR PAST MODELS

C     PARMCOV(,)   : STORAGE OF PARM ESTIMATES OF REGRESSION

C                  : COEFFICIENTS FOR PAST MODELS
C     NSUBS        : NUMBER OF OBSERVATIONS ENTERING ANALYSIS.
C     OUTPUTF      : OUTPUT FILE NAME FOR ANALYSIS RESULTS.
C     ST()         : SQUARE ROOTS OF DIAGONAL ELEMENTS OF COV. MATRIX
C     STAT         : SCORE TEST VALUE
C     START(5)     : TEMPORARY STORAGE FOR BASELINE STARTING VALUES.
C     SUM()        : SUM OF EACH COVARIATE
C     TOTAL        : TOTAL NUMBER OF TUMORS (USED IN MOM ESTIMATION)
C     XO()         : VECTOR OF PARAMETER ESTIMATES AT ANY ITERATION.
C     XINC()       : NEGATIVE OF THE SCORE VECTOR.
C     XX(,)        : INPUT COVARIATE MATRIX OF ALL COVARIATES
C     XX1(,)       : COVARIATES ENTERING ANALYSIS.
C


      DOUBLE PRECISION  ST(25),ID,STAT,MAT(25,25),CHI
      DOUBLE PRECISION  PARMCOV(25,25),PARMBASE(25,3),BASENULL(3)
      DOUBLE PRECISION  A(25,25),XO(25),XINC(25),X(25)
      DOUBLE PRECISION  MAXX(25),MINN(25),SUM (25),MISSING(25)
      DOUBLE PRECISION  LIKRATIO(25),DLGLK
      DOUBLE PRECISION  TOTAL, GMEAN, VAR
      DOUBLE PRECISION  RESP(3000,2)
      INTEGER           M,I,J,ICOL,NVARS
      INTEGER           FLAG [ALLOCATABLE] (:)
      INTEGER           COL(25),BASE,NINTS,C1,C2,NUMMISS
      INTEGER           NRUN,MODVAR(25,25),MODINT(25,25)


      COMMON           XX(3000,30), XX1(3000,30), P(25,25),
     +                 MDL,NSUBS, START(5), COVTEMP

      INTEGER          INTERDEX (10,10)

      CHARACTER*12     OUTPUTF,DICTION,INPUTF
      CHARACTER*10     COVNAME(100)
      CHARACTER*10     N_EVENT, EXP_TIME
      CHARACTER        COVRT(25)*10,GMA(3)*5, ANS
      LOGICAL          EX


C     THE FOLLOWING DEFINITIONS ARE FOR USING THE OPTIMIZATION
C     ROUTINES IN NAG. SEE NAG MANUAL FOR A DESCRIPTION OF THESE
C     VARIABLES.


      INTEGER          N,NCLIN,NCNLN
      INTEGER          NROWA,NROWJ,NROWR
      INTEGER          LIWORK,LWORK
      DOUBLE PRECISION BIGBND,OBJF
      INTEGER          IFAIL,ITER
      DOUBLE PRECISION AA(1,25),BL(25),BU(25),C(1),CJAC(1,25),
     +                 CLAMDA(25),R(25,25),
     +                 WORK(2000)
```

```
      DOUBLE PRECISION USER(1)
      DOUBLE PRECISION OBJGRD(25)
      INTEGER          ISTATE(25),IWORK(100),IUSER(1)
      EXTERNAL         OBJFUN,E04UCF,E04UDM,E04UEF,OBJNULL

      DATA  NRUN,NINTS,M,NCOV/0,0,0,0/



C  SET ALL THE VARIABLES THAT WILL BE USED IN THE NAG SUBROUTINE

       NROWR=25
       NCLIN=0
       NCNLN=0
       NROWA=1
       NROWJ=1
       LIWORK=100
       LWORK=2000
       BIGBND=1.0E10


        DO 2000 K=1,25
           BL(K)=-BIGBND
           BU(K)=BIGBND
           CLAMDA(K)=0
           XO(K)=0
           X(K)=0

2000       CONTINUE

C       SET LOWER BOUNDARIES FOR SHAPE AND MEAN OF GAMMA DIST TO 0

        BL(1)=0.000001
        BL(2)=0.000001



C      INITIALIZE ARRAYS AND VARIABLES THAT WILL BE USED IN THE PROGRAM

      DO 1994 I = 1,3000
              RESP(I,1)=0
              RESP(I,2)=0
              DO 1994 J = 1,30
                 XX(I,J) = 0
1994             XX1(I,J) = 0

      DO 1996 I = 1,25
              DO 1996 J = 1,25
                      MODVAR(I,J) = 0
                      MODINT(I,J) = 0
                      PARMCOV(I,J) = 0
1996                  P(I,J) = 0


      NUMMISS=0
      MISS=0
      MDL=2
      BASE = 0
      BASENULL(1)=0
      BASENULL(2)=0
      BASENULL(3)=0
      GMA(1) = 'SHAPE'
      GMA(2) = ' MEAN'
      GMA(3) = ' GAP1'
      COVTEMP=0
      DO 9192 I=1,25
         DO 9192 J=1,3
9192            PARMBASE(I,J)=0


C     PRINT HEADER

      WRITE(*,1024)
      WRITE(*,72)

72    FORMAT(5X'MIXED EFFECTS POISSON REGRESSION MODEL FOR',1X,
     + 'REPEATED EVENTS')

      WRITE(*,1024)
      WRITE(*,*)
      WRITE(*,*)

C     PROMPT FOR DICTIONARY AND OUTPUT FILE

222   WRITE(*,8010)
8010  FORMAT(1X 'WHAT IS THE DICTIONARY FILE? ',$)
      READ(*,'(A)')DICTION

      INQUIRE(FILE=DICTION,EXIST=EX)
      IF (EX) THEN
         OPEN(26,FILE=DICTION,STATUS='OLD')
      ELSE
         WRITE(*,*)'NO SUCH FILE!'
         GOTO 222
      ENDIF
      WRITE(*,*)


      WRITE(*,8011)
8011  FORMAT(1X 'SPECIFY FILE FOR PRINTED OUTPUT: ',$)
      READ(*,'(A)')OUTPUTF
      OPEN(25,FILE=OUTPUTF)


      READ(26,'(A)')INPUTF

      INQUIRE(FILE=INPUTF,EXIST=EX)
      IF(EX) THEN
       OPEN(28,FILE=INPUTF,STATUS='OLD')
      ENDIF

C     READ RELEVANT INFORMATION FROM DICTIONARY FILE AND DATA FILE


      READ(26,*)NVARS,NSUBS
      READ(26,'(A)') COVNAME(1), N_EVENT, EXP_TIME,
     +     (COVNAME(J), J = 2, NVARS)
      CLOSE(26)

C     THE DATA FILE HAS ID IN THE FIRST COLUMN

      NVARS = NVARS-1
      ALLOCATE(FLAG(NSUBS))

      WRITE(*,*)
      WRITE(*,*) 'READING INPUT DATA...PLEASE WAIT'
      WRITE(*,*)
```

```
C PRINT NUMBER OF SUBJECTS FOR VERIFICATION

      WRITE(*, 6980) NSUBS
6980  FORMAT(1X, I6,1X, 'RECORDS ARE BEING READ.')

      do 999 i =1, nsubs
999         read(28,*) id, resp(i,1), resp(i,2)

      close(28)
      OPEN(28,FILE=INPUTF,STATUS='OLD')
      k=1
      DO 1 I = 1,NSUBS
         if ((resp(i,1) .ge.0) .and. (resp(i,2) .ge. 0)) then
            READ(28,*) ID,( XX(k,J), J = 1,NVARS+2)
            do 9991 j=1, nvars+2
9991           xx1(k,j) = xx(k,j)
            k=k+1
         else
            miss=miss+1
         endif
1     continue

      WRITE(*,*)
      WRITE(*,601) MISS
601   FORMAT(1X,I3,1X, 'RECORDS ARE BEING IGNORED DUE TO MISSING',
     +     1x, 'NUMBER OF EVENTS', /,4x,'AND/OR EXPOSURE TIME')
      WRITE(*,*)

      nsubs=nsubs-miss

C  PRINT FIRST FIVE RECORDS FOR DATA VERIFICATION IF REQUESTED

      WRITE(*,8012)
8012  FORMAT(1X 'PRINT FIRST FIVE RECORDS? (Y/N): ',$)
      READ(*,'(A)') ANS

      IF ((ANS .EQ. 'Y') .OR. (ANS .EQ. 'y') .OR.
     +   (ANS .EQ. 'YES') .OR. (ANS .EQ. 'yes')) THEN
          DO 1995 I = 1, 5

             WRITE(*,7093) I
7093         FORMAT(1X 'RECORD ', I2)
             WRITE(*,'(7(F10.4,1X))') (XX(I,J), J = 1,NVARS+2)
1995         WRITE(*,*)
      ENDIF

      CLOSE(28)

C CALCULATE MIN, MAX, NUMBER MISSING AND SUM FOR EACH VARIABLE
C INITIALIZE THE ARRAYS MAXX, MINN, MISSING AND SUM
C THE CODE FOR MISSING COVARIATES IN THE DATA FILE SHOULD BE -999
C IF AN ENTRY IS NEGATIVE FOR THE N_TUMORS AND EXP_TIME, IT IS
C CONSIDERED AS MISSING.


      DO 901 J = 1,NVARS+2
         MISSING(J)=0
         MAXX(J) = -100
         MINN(J) = 1000
901      SUM(J) = 0

      DO 913 I = 1,NSUBS
         DO 913 J = 1,NVARS+2
            IF ((J .EQ. 1) .OR. (J .EQ. 2)) THEN
               MAXX(J) = MAX (MAXX(J),XX(I,J))
               MINN(J) = MIN (MINN(J),XX(I,J))
               SUM(J) = SUM(J) + XX(I,J)
            ELSE
               IF (XX(I,J) .NE. -999) THEN
                  MAXX(J) = MAX (MAXX(J),XX(I,J))
                  MINN(J) = MIN (MINN(J),XX(I,J))
                  SUM(J) = SUM(J) + XX(I,J)
               ELSE
                 MISSING(J) = MISSING(J) + 1
               ENDIF
            ENDIF
913      CONTINUE

C DISPLAY BRIEF DATA SUMMARY
      WRITE(*,*)
      WRITE(*,911)
911   FORMAT(1X 'QUICK DATA SUMMARY:'5X,'MEAN',9X, 'MIN',
     +      9X, 'MAX',7X,' # MISSING')
      WRITE(*,922) N_EVENT,SUM(1)/NSUBS,MINN(1),
     +          MAXX(1), MISSING(1)
      WRITE(*,922) EXP_TIME,SUM(2)/NSUBS,MINN(2),
     +           MAXX(2),MISSING(2)

      WRITE(*,922)(COVNAME(J+1),SUM(J+2)/(NSUBS-MISSING(J+2)),
     +            MINN(J+2), MAXX(J+2), MISSING(J+2), J=1,NVARS)

922   FORMAT(1X,a18,f9.3,4X,f9.1,4X,f9.1,8X,f4.0)
      WRITE(*,*)

C CHECK TO SEE IF NO OF TUMORS OR EXPOSURE TIME IS MISSING.
C DELETE ALL RECORDS THAT HAVE EITHER OF THE ABOVE MISSING.


C ICOL = 0 IF NO MODELS HAVE BEEN FITTED YET IN THIS SESSION; 1 o/w.
      ICOL = 0

17    WRITE(*, 2003)
2003  FORMAT(1X 'DO YOU WISH TO CONTINUE? (Y/N) ',$)
      READ(*,'(A)') ANS

      IF((ANS .EQ. 'N') .OR. (ANS .EQ. 'n') .OR.
     +      (ANS .EQ. 'NO') .OR. (ANS .EQ. 'no')) THEN
          GOTO 9
      ENDIF

      IF ((ANS .EQ. 'Y') .OR. (ANS .EQ. 'y') .OR.
     +      (ANS .EQ. 'YES') .OR. (ANS .EQ. 'yes')) THEN
         GOTO 2323
      ELSE
         WRITE(*,*) 'INVALID ANSWER!'
         GOTO 17
      ENDIF

2323  WRITE(*,*)


C INPUT START VALUES FOR PARMAMETERS OF BASELINE MODEL USING
C METHOD OF MOMENTS ESTIMATION

      TOTAL=0
      GMEAN=0
      VAR=0

      DO 7070 I = 1, NSUBS
7070         TOTAL = TOTAL + XX(I,1)
```

```
      GMEAN=TOTAL/NSUBS

      DO 7071 I = 1, NSUBS
7071     VAR = VAR + (XX(I,1) - GMEAN)*(XX(I,1) - GMEAN)


      XO(1) = (VAR/TOTAL) - 1.0
      XO(2) = GMEAN

      M=2
      N=M

C  SET ERROR FLAG FOR NAG EQUAL TO -1. SEE MANUAL FOR AN EXPLANATION.


      IFAIL=-1
      COVTEMP=0


      WRITE(*,*)
      WRITE(*,7075)
      WRITE(*,*) '(NULL MODEL).....PLEASE WAIT'

7075  FORMAT(1X 'MAXIMIZING THE LIKELIHOOD FUNCTION FOR THE',1X,
     + 'MODEL WITHOUT ANY COVARIATES')

       CALL E04UEF('MAJOR PRINT LEVEL = 0')

7073   CALL E04UCF(N,NCLIN,NCNLN,NROWA,NROWJ,NROWR,AA,BL,BU,
     + E04UDM,OBJNULL,ITER,ISTATE,C,CJAC,CLAMDA,OBJF,OBJGRD,R,
     + XO,IWORK,LIWORK,WORK,LWORK,IUSER,USER,IFAIL)

C    PRINT ERROR MESSAGES DEPENDING ON THE VALUE OF IFAIL

      IF (IFAIL .EQ. 0) THEN
         WRITE(*,*)
         WRITE(*,6666) ITER
         WRITE(*,*)
      ENDIF

6666  FORMAT(1X 'SUCCESSFUL CONVERGENCE HAS BEEN REACHED AFTER',I4,
     + 1X,'ITERATIONS')


      IF (IFAIL .EQ. 1) THEN
          WRITE(*,6971)
          WRITE(*,*) 'EXITING THE PROGRAM'
          GOTO 9
      ENDIF



       IF (IFAIL .EQ. 2) THEN
           WRITE(*,6972)
           WRITE(*,*) 'EXITING THE PROGRAM'
           GOTO 9
       ENDIF



       IF (IFAIL .EQ. 3) THEN
          WRITE(*,6973)
           WRITE(*,*) 'EXITING THE PROGRAM'
           GOTO 9
       ENDIF



       IF (IFAIL .EQ. 4) THEN
           WRITE(*,*) 'ITERATION LIMIT EXCEEDED!'
           WRITE(*,*) ' EXITING THE PROGRAM'
           GOTO 9
       ENDIF


       IF (IFAIL .EQ. 6) THEN
           WRITE(*,6975)
           WRITE(*,*) ' EXITING THE PROGRAM'
           GOTO 9
       ENDIF


      IF (IFAIL .EQ. 7) THEN
          WRITE(*,6976)
          WRITE(*,*) ' EXITING THE PROGRAM'
          GOTO 9
      ENDIF

      IF (IFAIL .EQ. 9) THEN
          WRITE(*,6977)
          WRITE(*,*) ' EXITING THE PROGRAM'
          GOTO 9
      ENDIF

C     NEGATE THE OBJECTIVE FUNCTION ON SUCCESSFUL CONVERGENCE
C     SINCE THE NAG ROUTINE DOES A MINIMIZATION

      OBJF=-OBJF
      NCOV=0


C     COMPUTE STANDARD ERRORS FOR MIXING DISTRIBUTION PARAMETERS


      CALL PARTIII(XO,A,M,XINC,MAT,NCOV,OBJF)
      CALL ARINV(M,MAT)

      DO 7095  I = 1,M
         IF(MAT(I,I) .LE. 0.0) THEN
            WRITE(*,*)' INFORMATION MATRIX IS SINGULAR'
            GO TO 7078
         ENDIF
7095    ST(I)=SQRT(MAT(I,I))


C WRITE TO THE OUTPUT FILE THE MLE AND SDs OF THE PARMS OF THE MIXING DISTN


      WRITE(25,8005) NSUBS
      WRITE(25,*)
      WRITE(25,*)
      WRITE(*,8005) NSUBS
      WRITE(*,*)
8005  FORMAT(15X, 'STATISTICAL OUTPUT BASED ON', I6, ' RECORDS')
      WRITE(25,8006)
8006  FORMAT(15X, 'PARAMETERS OF THE MIXING DISTRIBUTION')
```

```
      WRITE(25,*)
      WRITE(25,*)
      WRITE(*,8006)
      WRITE(*,*)
      WRITE(*,*)


      WRITE(25,1015)XO(1),XO(2),ST(1),ST(2)
      WRITE(25,*)
      WRITE(*,1015)XO(1),XO(2),ST(1),ST(2)
      WRITE(*,*)
      DLGLK = -2*OBJF
      IF(NCOV.EQ.0) THEN
          WRITE(25,1017) DLGLK
          WRITE(*,1017) DLGLK
      ENDIF

C     STORE MIXING DIST MLE'S IN BASENULL()

      DO 7085 I = 1,M
7085     BASENULL(I)=XO(I)


      DO 7082 I = 1, NSUBS
         DO 7082 J = 1, NVARS+2
7082        XX1(I,J)=0

7078  WRITE(*,*)

C     PROMPT FOR REGRESSION

      WRITE(*,7077)
7077  FORMAT(' DO YOU WISH TO ADD COVARIATES TO THE MODEL? (Y/N) ',$)
      READ(*,'(A)') ANS

      IF((ANS.EQ.'N') .OR. (ANS .EQ. 'n') .OR.
     + (ANS .EQ. 'NO').OR. (ANS .EQ. 'no')) THEN
       GOTO 9
      ENDIF

      IF ((ANS .EQ. 'Y') .OR. (ANS .EQ. 'y')
     +  .OR. (ANS .EQ. 'yes') .OR. (ANS .EQ. 'YES')) THEN
         GOTO 7081

      ELSE
         WRITE(*,*) 'INVALID ANSWER'
         GOTO 7078
      ENDIF

C     SET UP COVARIATE SELECTION

7081  WRITE(*,*)
      WRITE(*,*) '  INDEXED LIST OF COVARIATES '
      WRITE(*,'(5(I4,A1,1X,A10))') (J,':',COVNAME(J+1),J = 1,NVARS)
      WRITE(*,*)


C     IF THIS IS THE FIRST MODEL BEING FITTED GOTO 22

      IF (ICOL .EQ. 0) GOTO 22

C     O/W DISPLAY PREVIOUS MODELS AND THEIR DEVIANCES

      NINTS=0
      NCOV=0

      WRITE(*,*) 'PREVIOUS MODEL(S) FITTED'

      WRITE(*,*)' MDL  0: NULL MODEL'
      WRITE(*,7092) DLGLK
      WRITE(*,*)
      DO 233 J = 1,NRUN
        WRITE(*,7091) J,(COVNAME(MODVAR(J,K)+1),K=2,MODVAR(J,1)+1)
7091    FORMAT(2X 'MDL',1X,I2,':',1X,10(5(A10,1X),/,10X))

        DO 2334 K = 1,MODINT(J,1)
           WRITE(*,'(10X,A5,A,A5,\)')COVNAME(MODINT(J,2*K)+1),'*',
     +        COVNAME(MODINT(J,2*K+1)+1)
2334       WRITE(*,*)
        WRITE(*,7092) LIKRATIO(J)
7092    FORMAT(10X 'DEVIANCE',':',1X,F15.5)

233     WRITE(*,'( )')

        WRITE(*,8013)
8013    FORMAT(1X 'WHICH MODEL DO YOU WISH TO USE AS A BASE?  ',$)
        READ(*,*)BASE
        NSUBS=NSUBS+NUMMISS

C RETRIEVE INFORMATION FROM BASE MODEL...E.G.: NUMBER OF COVARIATES,
C NUMBER OF INTERACTIONS ETC.

        IF(BASE .GT. 0) THEN
           C1 = MODVAR(BASE,1)
           C2 = MODINT(BASE,1)
        ELSE
          C1=0
          C2=0
        ENDIF

        WRITE(*,8014)
8014    FORMAT(1X 'HOW MANY NEW VARIABLES DO YOU WISH TO ADD? ',$)
        GOTO 2201

22      WRITE(*,7079)NVARS
2201    READ(*,*) NCOV

        IF(BASE.GT.0) THEN
         DO 2205 K = 1,C1
2205     COL(K) = MODVAR(BASE,K+1)
        ENDIF


        IF (NCOV .GT. 0) THEN
           WRITE(*,8015)
        ENDIF
8015    FORMAT(1X 'ENTER INDEX OF CHOSEN COVARIATES: ',$)

        IF(BASE .GT. 0) THEN
           NCOV = C1 + NCOV
           IF(NCOV.GT.C1) THEN
              READ(*,*) (COL(K), K = C1+1, NCOV)
           ENDIF
        ELSE
           IF(NCOV.GT.0) THEN
             READ(*,*) (COL(K), K = 1, NCOV)
           ENDIF
        ENDIF

        MODVAR(NRUN+1,1) = NCOV
        DO 2206 K = 1,NCOV
2206       MODVAR(NRUN+1,K+1) = COL(K)

        ICOL=ICOL+1
```

```
      IF ((ICOL .GT. 0) .AND. (NCOV .GT. 1)) THEN
         WRITE(*,8016)
         GOTO 2202
      ENDIF
8016  FORMAT(1X 'HOW MANY NEW INTERACTIONS? (0=none) ',$)

      IF (NCOV .GT. 1) THEN
         WRITE(*,8017)
2202  READ(*,*) NINTS
      ENDIF

8017  FORMAT(1X 'ENTER NUMBER OF PAIR-WISE INTERACTIONS (0=NONE):',$)

      IF(NINTS.GT.0) THEN
         IF(BASE .EQ. 0) THEN
            DO 20 J = 1,NINTS
             WRITE(*,8018)
8018         FORMAT(1X 'ENTER PAIR OF INDICES FOR ONE INTERACTION: ',$)
20           READ(*,*) INTERDEX(J,1),INTERDEX(J,2)

              MODINT(NRUN+1,1) = NINTS
              DO 2207 K = 1,NINTS
                 MODINT(NRUN+1,2*K) = INTERDEX(K,1)
2207             MODINT(NRUN+1,2*K+1) = INTERDEX(K,2)
         ELSE
           MODINT(NRUN+1,1) = NINTS + C2
           DO 2208 K = 1,C2
              INTERDEX(K,1) = MODINT(BASE,2*K)
2208          INTERDEX(K,2) = MODINT(BASE,2*K+1)
           DO 2209 K = 1,2*C2
2209          MODINT(NRUN+1,K+1) = MODINT(BASE,K+1)
           DO 2211 J = C2+1,C2+NINTS
           WRITE(*,8019)
8019       FORMAT(1X 'ENTER PAIR OF INDICES FOR ONE INTERACTION: ',$)
2211       READ(*,*) INTERDEX(J,1),INTERDEX(J,2)
           DO 2212 K = 1,NINTS
             MODINT(NRUN+1,2*C2+2*K) = INTERDEX(C2+K,1)
2212         MODINT(NRUN+1,2*C2+2*K+1) = INTERDEX(C2+K,2)
          NINTS = C2+NINTS
         ENDIF
      ELSE
         IF (BASE .EQ. 0) THEN
            GOTO 2214
         ENDIF

         MODINT(NRUN+1,1) = C2
         DO 2218 K = 1,C2
           INTERDEX(K,1) = MODINT(BASE,2*K)
2218       INTERDEX(K,2) = MODINT(BASE,2*K+1)
         DO 2219 K = 1,2*C2
2219     MODINT(NRUN+1,K+1) = MODINT(BASE,K+1)
         NINTS = C2
2214     ENDIF

C  CREATE DATA MATRIX IGNORING THE RECORDS WITH MISSING OBSERVATIONS.

         NUMMISS=0
         DO 32 I = 1,3000
             do 32 J=1,30
32              XX1(I,J)=0

         DO 7066 I = 1, NSUBS
7066         FLAG(I)=0

        DO 7083 I = 1, NSUBS
           DO 7083 J = 1, NCOV
               IF (XX(I,COL(J)+2).EQ.-999.00) THEN
                  FLAG(I) = 1
                  NUMMISS=NUMMISS+1
7083           ENDIF

C CREATE DATA MATRIX
          K=1
          DO 7067 I=1, NSUBS
             IF (FLAG(I) .EQ. 0) THEN
               XX1(K,1) = XX(I,1)
               XX1(K,2) = XX(I,2)
               DO 2 J = 1,NCOV
2                XX1(K,J+2) = XX(I,COL(J)+2)
               K=K+1
            ENDIF
7067      CONTINUE

C CREATE INTERACTIONS

       IF (NINTS .GT. 0) THEN
          J=1
          DO 31 K = 1, NSUBS
            IF (FLAG(K) .EQ. 0) THEN
                DO 30 L = 1,NINTS
30                 XX1(J,NCOV+L+2) = XX(K,INTERDEX(L,1)+2)
     +                          * XX(K,INTERDEX(L,2)+2)
            J=J+1
            ENDIF
31        CONTINUE
       ENDIF

C DECREMENT NUMBER OF SUBJECTS BY NUMBER WHO HAVE MISSING VALUES
C FOR THE PARTICULAR CHOICE OF COVARIATE.

       NSUBS=NSUBS-NUMMISS

C      STORE VARIABLE NAMES

       DO 106 J = 1,NCOV
106        COVRT(J) = COVNAME(COL(J)+1)
           IF(NINTS .GT. 0) THEN
           DO 107 J = 1,NINTS
107          COVRT(J+NCOV) = COVNAME(INTERDEX(J,1)+1)(:4)//'*'//
     +             COVNAME(INTERDEX(J,2)+1)(:4)
           ENDIF


       NCOV = MODVAR(NRUN+1,1) + MODINT(NRUN+1,1)
       M = NCOV + 2



C     ENTER START VALUES FOR PARMAMETERS OF BASELINE MODEL INTO X

       IF (BASE .EQ. 0) THEN
           DO 3123 I = 1, M - NCOV
               START(I) = BASENULL(I)
3123           X(I)=START(I)
       ELSE
           DO 9191 I = 1, M - NCOV
               START(I) = PARMBASE(BASE,I)
9191           X(I)=START(I)
       ENDIF
```

```
C FOR REGRESSION, ENTER START VALUES FOR THE PARMS INTO X.
C THE BASELINE PARAMETERS ARE ASSIGNED THE MLE FROM THE NULL
C MODEL AS THEIR STARTING VALUES. THE REGRESSION PARAMETERS
C ARE ASSIGNED A STARTING VALUE OF ZERO.

C      IF (NRUN .EQ. 0) THEN
            DO 6969 I = M-NCOV+1, M
6969           X(I)=0

C       ELSE
C          IF(BASE .GT. 0) THEN
C             POLD = MODVAR(BASE,1) + MODINT(BASE,1)
C             DO 3124 I = 1, POLD
C3124            X(I+2)=PARMCOV(BASE,I)
C          ELSE
C             POLD=0
C          ENDIF

C          DO 74 I = POLD+1, NCOV
C74           X(I+2)=0
C       ENDIF


C INITIALIZE THE VECTOR OF ESTIMATED S.D.s AND THE ESTIMATED COVARIANCE MATRIX

      DO  34 I = 1 , M
            ST(I)=0.0D0
            DO  33 J = 1 , M
                A(I,J)=0
33              MAT(I,J) = 0.0D0
34    CONTINUE

C     -----SOLVE PROBLEM ---------------------------

      WRITE(*,1007)
      CHI = 0.0
      ICOV=NCOV
      IFLG=0
      IF(NCOV.EQ.0) IFLG=1

C     CALCULATE THE SCORE TEST STATISTIC
      DO 8090 I=M-NCOV+1,25
          XINC(I)=0
8090      XO(I)=0
      MM=M
      CALL PARTIII(XO,A,MM,XINC,MAT,ICOV,OBJF)

      CALL SCORE(A,ICOV,XINC,STAT)


       N=M
       IFAIL=-1
       COVTEMP=NCOV
       OBJF=0
       DO 7099 I=1,25
7099     OBJGRD(I)=0

       WRITE(*,*) '-----OPTIMIZING THE LIKELIHOOD FUNCTION-----'
       WRITE(*,*) '(PLEASE WAIT)'

       CALL E04UEF('MAJOR PRINT LEVEL = 0')
       CALL E04UEF('MAJOR ITERATION LIMIT = 1000')

6970   CALL E04UCF(N,NCLIN,NCNLN,NROWA,NROWJ,NROWR,AA,BL,BU,
     + E04UDM,OBJFUN,ITER,ISTATE,C,CJAC,CLAMDA,OBJF,OBJGRD,R,
     + X,IWORK,LIWORK,WORK,LWORK,IUSER,USER,IFAIL)

C     DISPLAY ERROR MESSAGES DEPENDING ON THE VALUE OF IFAIL

       IF (IFAIL .EQ. 0) THEN
         WRITE(*,*)
         WRITE(*,6666) ITER
         WRITE(*,*)
       ENDIF


       IF (IFAIL .EQ. 1) THEN
          WRITE(*,1007)
          WRITE(*,6971)
          WRITE(1007)
          GOTO 8083
       ENDIF

6971   FORMAT(1X, 'ERROR: THE FINAL ITERATE SATISTFIES FIRST ORDER',1X,
     + 'KUHN-TUCKER CONDITIONS, BUT THE SEQUENCE OF ITERATES HAS NOT',
     + 1X,'CONVERGED! POSSIBLE REASON: REQUESTED ACCURACY IS NOT',
     + 'ATTAINABLE WITH THE GIVEN PRECISION OF THE PROBLEM.')

       IF (IFAIL .EQ. 2) THEN
          WRITE(*,1007)
          WRITE(*,6972)
          WRITE(*,1007)
       ENDIF

6972   FORMAT(1X, 'ERROR: NO FEASIBLE POINT EXISTS FOR THE LINEAR',1X,
     + 'CONSTRAINTS AND BOUNDS FOR THE GIVEN VALUE OF LINEAR',1X,
     + 'FEASIBILITY TOLERANCE')



       IF (IFAIL .EQ. 3) THEN
         WRITE(*,1007)
         WRITE(*,6973)
         WRITE(*,1007)
         GOTO 8083
       ENDIF

6973   FORMAT(1X, 'ERROR: NO FEASIBLE POINT EXISTS FOR THE NON-',1X,
     + 'LINEAR CONSTRAINTS')


       IF (IFAIL .EQ. 4) THEN
           WRITE(*,1007)
           WRITE(*,*) 'ITERATIONS EXCEEDED 1000!'
           WRITE(*,1007)
           GOTO 8083
       ENDIF


       IF (IFAIL .EQ. 6) THEN
          WRITE(*,1007)
          WRITE(*,6975)
          WRITE(*,1007)
          GOTO 8083
       ENDIF

6975   FORMAT(1X, 'FINAL ITERATE DOES NOT SATISFY FIRST',1X,
     + 'ORDER KUHN-TUCKER CONDITIONS')


       IF (IFAIL .EQ. 7) THEN
          WRITE(*,1007)
          WRITE(*,6976)
```

```
         WRITE(*,1007)
         GOTO 8083
      ENDIF

6976   FORMAT(1X, 'USER-PROVIDED DERIVITIVES OF THE',1X,
     + 'OBJECTIVE FUNCTION APPEAR TO BE INCORRECT')

      IF (IFAIL .EQ. 9) THEN
         WRITE(*,1007)
         WRITE(*,6977)
         WRITE(*,1007)
         GOTO 8083
      ENDIF

6977  FORMAT(' AN INPUT PARAMETER IS INVALID')

C     NEGATE THE OBJECTIVE FUNCTION ON SUCCESSFUL CONVERGENCE
C     SINCE THE NAG ROUTINE DOES A MINIMIZATION

      OBJF=-OBJF
      GOTO 8084


C IF CONVERGENCE HAS BEEN REACHED FOR THE REG. MODEL , EVALUATE L.R.

8083  WRITE(*,8020)
8020  FORMAT(' DO YOU WISH TO CHANGE THE MODEL? (Y/N) ',$)
      READ(*,'(A)') ANS

      IF ((ANS.EQ.'Y').OR.(ANS.EQ.'y').OR.(ANS.EQ.'YES').OR.
     +  (ANS.EQ.'yes')) THEN
          ICOL=ICOL-1

          DO 8082 I=1,25
             MODVAR(NRUN+1,I)=0
8082         MODINT(NRUN+1,I)=0
          GOTO 7081
      ENDIF

      IF ((ANS.EQ.'N').OR.(ANS.EQ.'n').OR.(ANS.EQ.'NO').OR.
     +   (ANS.EQ.'no')) THEN
          WRITE(*,*) ' EXITING THE PROGRAM'
          GOTO 9
      ELSE
         WRITE(*,*) ' INVALID ANSWER!'
         GOTO 8083
      ENDIF



8084  CHI = DLGLK - (-2*OBJF)
      LIKRATIO(NRUN+1)=-2*OBJF



C DECLARE SUCCESSFUL CONVERGENCE AND DISPLAY NUMBER OF ITERATIONS NEEDED TO
C REACH CONVERGENCE AND THE VALUE OF THE OBJECTIVE FUNCTION

        DO 3115 K = 1,2
3115       PARMBASE(NRUN+1,K) = X(K)
           DO 3116 K = 3,M
3116         PARMCOV(NRUN+1,K-2) = X(K)




        NRUN = NRUN + 1

        IF (NRUN .GT. 25) THEN
           WRITE(*,*)'MAXIMUM NUMBER OF MODEL ATTEMPTS HAS BEEN',
     +     'REACHED.'
           WRITE(*,*)'START THE PROGRAM OVER'
           GOTO 9
        ENDIF

C    COMPUTE SAMPLE INFORMATION MATRIX

        CALL PARTIII(X,A,M,XINC,MAT,NCOV,OBJF)

C INVERT THE SAMPLE INFORMATION MATRIX (-A = MAT).

        CALL ARINV ( M,MAT )

C      OUTPUT: SPECIFY WHICH MODEL HAS BEEN USED IN THE ANALYSIS.

       WRITE(25,1007)
       WRITE(25,*) 'MLE FROM MIXED POISSON REGRESSION MODEL'

C WRITE THE NAMES OF COVARIATES ENTERING ANALYSIS.

       WRITE(25,1014) (COVRT(K),K = 1 , NCOV)
       WRITE(25,*)
       WRITE(25,*)

C EVALUATE THE ESTIMATED S.D. OF THE M.L.Es OF THE PARMS.

       DO 81  I = 1,M
          IF(MAT(I,I) .LE. 0.0) THEN
             WRITE(*,*)' INFORMATION MATRIX IS SINGULAR'
             GO TO 8083
          ENDIF
81     ST(I)=SQRT(MAT(I,I))


C WRITE TO THE OUTPUT FILE THE MLE AND SDs OF THE PARMS OF THE MIXING DISTN


       WRITE(25,8005) NSUBS
       WRITE(25,*)
       WRITE(25,*)

       WRITE(*,8005) NSUBS
       WRITE(*,*)

       WRITE(25,8006)
       WRITE(25,*)
       WRITE(*,8006)
       WRITE(*,*)


       WRITE(25,1015)X(1),X(2),ST(1),ST(2)
       WRITE(25,*)
       WRITE(*,1015)X(1),X(2),ST(1),ST(2)
       WRITE(*,*)



C IF THE MODEL CONTAINS COVARIATES THEN WRITE TO FILE THE VALUES OF THE
C L.R.,SCORE STATISTIC AND D.F.

         IF(NCOV .GT. 0 ) THEN
            WRITE(25,1018) -2*OBJF,DLGLK
            WRITE(*,1018) -2*OBJF,DLGLK
            WRITE(25,1019)CHI,ICOV
            WRITE(*,1019)CHI,ICOV
```

```
            WRITE(25,1020)STAT,ICOV
            WRITE(*,1020)STAT,ICOV
            WRITE(25,1021)
            WRITE(*,1021)

C EVALUATE THE Z-VALUE FOR EACH COVARIATE (MLE/S.E).
C EVALUATE THE NORMAL-BASED 95% C.I. FOR EACH REGRESSION COEFFICIENT
C IF THE 95% C.I. EXCLUDES ZERO THEN FLAG THE COVARIATE AS SIGNIFICANT.

         IFLAG=0


         DO 90 J = 3 , M
            DZ=0.0
            CIL=0.0
            CIU=0.0
            DZ=X(J)/ST(J)
            CIL=X(J)-1.96*ST(J)
            CIU=X(J)+1.96*ST(J)
            IF(ABS(DZ).GT. 1.96) THEN
              IFLAG=1
              WRITE(25,1022)COVRT(J-2),X(J),ST(J),DZ,CIL,CIU
              WRITE(*,1022)COVRT(J-2),X(J),ST(J),DZ,CIL,CIU
            ELSE
              WRITE(25,1023)COVRT(J-2),X(J),ST(J),DZ,CIL,CIU
              WRITE(*,1023)COVRT(J-2),X(J),ST(J),DZ,CIL,CIU
            ENDIF
90        CONTINUE

10        CONTINUE


         IF(IFLAG.EQ. 1) THEN
           WRITE(25,*)
           WRITE(25,*)'       *: SIGNIFICANT AT THE  5%  LEVEL'
           WRITE(*,*)
           WRITE(*,*)'       *: SIGNIFICANT AT THE  5%  LEVEL'
         ENDIF
         WRITE(25,1024)
       ELSE
         WRITE(*,1025) -OBJF
       ENDIF
1025   FORMAT(6X,'==== -LOG-LIKLIHOOD  =',F12.5 )

C      WRITE TO OUTPUT THE CORRELATION MATRIX.

       WRITE(25,'(23X,30H ESTIMATED CORRELATION MATRIX:)')
       WRITE(25,*)
       WRITE(25,*)
       WRITE(25,'(7X,30I7)')(K, K=1,M)
       DO 110 I = 1 , M-NCOV
110      WRITE(25,'(5X,A5,30(1X,F6.3))')GMA(I),
     +     ((MAT(I,J)/(ST(I)*ST(J))),J=1,M)
       DO 12 I = M-NCOV+1 , M
12     WRITE(25,'(A10,30(1X,F6.3))') COVRT(I-M+NCOV),
     + ((MAT(I,J)/(ST(I)*ST(J))),J=1,M)
       IF(IFLG.EQ.1) THEN
          GOTO 7078
       ENDIF
99     CONTINUE
       GO TO 7078

C       ***************     FORMAT STATEMENTS   ********************

1005   FORMAT(2X,'STARTING VALUES FOR NULL MODEL., (',1X,I2,1X,'PAR.)?')
1006   FORMAT(2X,'STARTING VALUES FOR CURRENT MODEL, (',I6,1X,
     +    'PARAMETERS) ? ' )
1007   FORMAT(39H -------------------------------------,
     +    39H--------------------------------------)
1009   FORMAT (' ====',5X,'              GRADIENT VECTOR:')
1010   FORMAT(' ==== ',7H ITER =,I4,6X,'-OBJF =',F16.5,7X,8H COND = ,
     +    E16.5,/,' ==== ',
     +    32H              XO(1),.....,XO(,I2,1H),/,10(8F15.8))
1011   FORMAT(25H SUCCESSFUL CONVERGENCE: 10X,'OBJF =',F12.4)
1012   FORMAT(2X,'NEW VALUES., (',1X,I2,1X,'PAR.)?  ')
1014   FORMAT('COVARIATES ENTERING ANALYSIS: ',4A12,/,30(30X,4A12,/)
     +     ,'.')
1015   FORMAT(16X,'SHAPE =',F12.5,10X,' MEAN  =',F11.6,/,16X,'ST.DEV='
     +     , '(', F11.5,')',10X,'ST.DEV=','(', F10.6,')')
1016   FORMAT(7X,'SHAPE =',F9.4,8X,' MEAN =',F11.6,5X,'1ST-GAP='
     +    ,F9.4,/,7X,'ST.DEV=','(', F9.5,')',6X,'ST.DEV=','(', F10.6
     +    ,')',5X,'ST.DEV=','(', F9.5,')')
1017   FORMAT(6X,'DEVIANCE FOR NULL MODEL (-2*(LOG-LIK)) = ',F13.5)
1018   FORMAT(6X,'DEVIANCE: CURRENT MODEL',3X,' =',F13.5,3X,' NULL =',
     +     F13.5)
1019   FORMAT(6X,'LIK. RATIO TEST STATISTIC',2X, '=',F13.5,' WITH ',I3,
     +     '  DEGREE(S) OF FREEDOM')
1020   FORMAT(6X,'SCORE TEST STATISTIC',7X, '= ',F12.5,' WITH ',I3,
     +     '  DEGREE(S) OF FREEDOM')
1021   FORMAT(/,6X,'COVARIATE',4X,'   M.L.E.',4X,'  ST.DEV.',5X,
     + 'Z-VALUE',4X,' 95% C. INTERVAL',/6X,'---------',4X,'---------',
     +     4X,'---------',5X,'-------',4X,'-----------------')
1022   FORMAT(4X,A10,5X,F9.6,'*',3X,F9.6,4X,F7.4,4X,'(',F7.3,' ,', F7.3,
     + ')')
1023   FORMAT(4X,A10,5X,F9.6,4X,F9.6,4X,F7.4,4X,'(',F7.3,' ,',F7.3,')')
1024   FORMAT(5X,58('-'))
7079   FORMAT(' HOW MANY COVARIATES DO YOU WISH TO USE? (1 --',
     + I3, ') ',$)

9      STOP
       END


      SUBROUTINE OBJFUN(MODE,N,X,OBJF,OBJGRD,NSTATE,IUSER,USER)

C  THIS SUBROUTINE COMPUTES THE OBJECTIVE FUNCTION, AND THE FIRST
C  PARTIAL DERIVITIVE FOR THE NAG SUBROUTINE

       INTEGER          MODE, N, NSTATE,IUSER(1),USER(1)
       DOUBLE PRECISION X(25), OBJF,OBJGRD(25)
       DOUBLE PRECISION MAT(25,25),XINC(25),A(25,25)
       INTEGER          K,COV
       COMMON           XX(3000,30),XX1(3000,30),P(25,25),
     +                  MDL, NSUBS, START(5), COVTEMP



       DO 669 K=1,25

669       XINC(K)=0



       COV=COVTEMP

       CALL PARTIII(X,A,N,XINC,MAT,COV,OBJF)

       OBJF=-OBJF

       DO 671 K=1,N
671       OBJGRD(K) = XINC(K)
672    RETURN
       END
```

```
      SUBROUTINE OBJNULL(MODE,N,XO,OBJF,OBJGRD,NSTATE,IUSER,USER)

C  THIS SUBROUTINE COMPUTES THE OBJECTIVE FUNCTION, AND THE FIRST
C  PARTIAL DERIVITIVE FOR THE NAG SUBROUTINE

       INTEGER           MODE,N, NSTATE,USER(1),IUSER(1)
       DOUBLE PRECISION XO(25), OBJF,OBJGRD(25)
       DOUBLE PRECISION MAT(25,25),XINC(25),A(25,25)
       INTEGER          K,COV
       COMMON           XX(3000,30),XX1(3000,30),P(25,25),
     +                  MDL, NSUBS, START(5), COVTEMP



       DO 666 K=1,25
666       XINC(K)=0


       COV=COVTEMP

       CALL PARTIII(XO,A,N,XINC,MAT,COV,OBJF)

       OBJF=-OBJF

       DO 668 K=1,N
668        OBJGRD(K) = XINC(K)
670    RETURN
       END
```

```
      SUBROUTINE DECOMP(MDIM,M,A,COND,IPVT,WORK)
      INTEGER   MDIM,M,IPVT(MDIM)
      REAL*8    COND,WORK(MDIM),A(25,25)
      REAL*8    EK,T,ANORM,YNORM,ZNORM
      INTEGER   MM1,I,J,K,KP1,KB,KM1,M1


      IPVT(M) = 1
      IF ( M .EQ. 1 ) GO TO 80
      MM1 = M-1
      ANORM = 0.0D0
      DO 10 J = 1 , M
         T = 0.0D0
         DO  5  I  = 1 , M
             T = T +DABS( A(I,J) )
5        CONTINUE
         IF ( T .GT. ANORM ) ANORM = T
10    CONTINUE
      DO 35 K = 1 , MM1
        KP1 = K + 1
        M1 = K
        DO 15 I = KP1 , M
          IF (DABS(A(I,K)) .GT. DABS(A(M1,K)) ) M1 = I
15      CONTINUE
        IPVT(K) = M1
        IF (M1 .NE. K ) IPVT(M) = -IPVT(M)
        T = A(M1,K)
        A(M1,K) = A(K,K)
        A(K,K) = T
        IF ( T .EQ.0.0D0 ) GO TO 35
        DO 20 I = KP1 , M
          A(I,K) = -A(I,K)/T
20        CONTINUE
        DO 30 J = KP1 , M
          T = A(M1,J)
          A(M1,J) = A(K,J)
          A(K,J) = T
          IF ( T .EQ. 0.0D0 ) GO TO 30
          DO 25 I = KP1 , M
            A(I,J) = A(I,J) + A(I,K)*T
25          CONTINUE
30        CONTINUE
35    CONTINUE

      DO 50 K = 1 , M
        T = 0.0D0
        IF ( K .EQ. 1 ) GO TO 45
        KM1 = K - 1
        DO 40 I = 1 , KM1
          T = T + A(I,K)*WORK(I)
40      CONTINUE
45      EK = 1.0D0
        IF ( T .LT. 0.0D0 ) EK = -1.0D0
        IF ( A(K,K) .EQ. 0.0D0 ) GO TO 90
        WORK(K) = -(EK + T )/A(K,K)
50    CONTINUE
      DO 60 KB = 1 , MM1
        K = M - KB
        T = 0.0D0
        KP1 = K + 1
        DO 55 I = KP1 , M
          T = T + A(I,K)*WORK(K)
55      CONTINUE
        WORK(K) = T
        M1 = IPVT(K)
```

```
      IF ( M1 .EQ. K ) GO TO 60
        T = WORK(M1)
        WORK(M1) = WORK(K)
        WORK(K) = T
60    CONTINUE

      YNORM = 0.0D0
      DO 65 I = 1 , M
        YNORM = YNORM + DABS( WORK(I) )
65    CONTINUE

      CALL SOLVE ( MDIM,M,A,WORK,IPVT )

      ZNORM = 0.0D0
      DO 70 I = 1 , M
        ZNORM = ZNORM + DABS( WORK(I) )
70    CONTINUE
      COND = ANORM*ZNORM/YNORM
      IF (COND .LT. 1.0D0 ) COND = 1.0D0
      RETURN
80    COND = 1.0D0
      IF ( A(1,1) .NE. 0.0D0 ) RETURN
90    COND = 1.0E+32
      RETURN
      END

      SUBROUTINE SOLVE(MDIM,M,A,XINC,IPVT)
      INTEGER     MDIM,M,IPVT(MDIM)
      REAL*8      A(MDIM,MDIM),XINC(MDIM)
      INTEGER     KB,KM1,MM1,KP1,I,K,M1
      REAL*8      T

      IF ( M .EQ. 1 ) GO TO 50
      MM1 = M - 1
      DO 20 K = 1 , MM1
        KP1 = K + 1
        M1 = IPVT(K)
        T = XINC(M1)
        XINC(M1) = XINC(K)
        XINC(K) = T
        DO 10 I = KP1 , M
          XINC(I) = XINC(I) + A(I,K)*T
10      CONTINUE
20    CONTINUE
      DO 40 KB = 1 , MM1
        KM1 = M - KB
        K = KM1 + 1
        XINC(K) = XINC(K)/A(K,K)
        T = -XINC(K)
        DO 30 I = 1 , KM1
          XINC(I) = XINC(I) + A(I,K)*T
30      CONTINUE
40    CONTINUE
50    XINC(1) = XINC(1)/A(1,1)
      RETURN
      END

      SUBROUTINE SCORE(A,ICOV,XINC,STAT)
C     THIS SUBROUTINE EVALUATES THE SCORE TEST STATISTIC FOR TESTING
C     THE NULL HYPOTHESIS THAT THE REGRESSION COEFFICIENTS IN THE
C     REGRESSION MODEL ARE ALL ZEROS. THE FOLLOWING QUANTITIES MUST
C     BE PASSED AS INPUT:
C
C     1. SAMPLE INFORMATION MATRIX = -A(25,25).
C     2. SCORE VECTOR AT THE MLE VALUES = -XINC(25,25).
C     3. NUMBER OF COVARIATES IN THE MODEL = ICOV.
C     4. THE NUMBER OF BASELINE PARAMETERS. = MC.
C
C     ON OUTPUT, THE SUBROUTINE RETURNS THE VALUE OF THE SCORE TEST
C     STATISTIC "STAT".
C
      REAL*8   A(25,25),BTMB(25,25),MAT(25,25),BTM(25,25),GOFF(25),
     +         STAT,U(25),XINC(25)
      INTEGER  MC,ICOV
      COMMON   XX(3000,30), XX1(3000,30),P(25,25),
     +         MDL, NSUBS, START(5), COVTEMP
C
      MC = 2

C
      DO 1 I = 1 , 25
        U(I) = -XINC(I)
        DO 1 J = 1 , 25
          BTM(I,J) = 0.0D0
          BTMB(I,J) = 0.0D0
          MAT(I,J) = 0.0D0
          A(I,J) = -A(I,J)
1     CONTINUE

C
C     CALCULATION OF GOF TEST.
C
C     STEP 1: FIND THE COVARIANCE MATRIX FOR THE BASELINE MODEL
C
      DO 2 J = 1 , MC
        DO 2 I = 1 , MC
          MAT(I,J) = A(I,J)
2     CONTINUE
      CALL ARINV(MC,MAT)
C
C     STEP 2: EVALUATE TRANS(IGB)*MAT*IGB
C
      DO 3 I = 1 , ICOV
        DO 3 J = 1 , MC
          DO 3 K = 1 , MC
            BTM(I,J) = BTM(I,J) + A(I+MC,K)*MAT(K,J)
3     CONTINUE
      DO 4 I = 1 , ICOV
        DO 4 J = 1 , ICOV
          DO 4 K = 1 , MC
            BTMB(I,J) = BTMB(I,J) + BTM(I,K)*A(K,J+MC)
4     CONTINUE

C     STEP 3: EVALUATE PARTITIONED INVERSE

      DO 5 I = 1 , ICOV
        GOFF(I) = 0.0
        DO 5 J = 1 , ICOV
          MAT(I,J) = A(I+MC,J+MC) - BTMB(I,J)
5     CONTINUE
      CALL ARINV ( ICOV,MAT )

C     STEP 4: CALCULATE THE  SCORE TEST STATISTIC
      STAT = 0.0
      DO 6 J = 1 , ICOV
        DO 6 I = 1 , ICOV
          GOFF(J) = GOFF(J) + U(I+MC)*MAT(I,J)
6     CONTINUE
      DO 7 I = 1 , ICOV
        STAT = STAT + GOFF(I)*U(I+MC)
7     CONTINUE

      RETURN
      END
```

```
      SUBROUTINE USPKD  (PACKED,NCHARS,UNPAKD,NCHMTB)
C                                    SPECIFICATIONS FOR ARGUMENTS
      INTEGER            NC,NCHARS,NCHMTB
      INTEGER          UNPAKD(nchars),PACKED(nchars),LBYTE,LBLANK
      INTEGER*2          IBYTE,IBLANK
      EQUIVALENCE (LBYTE,IBYTE)
      DATA               LBLANK /1H /
      DATA               IBYTE /1H /
      DATA               IBLANK /1H /

      NCHMTB = 0
C                                    RETURN IF NCHARS IS LE ZERO
      IF(NCHARS.LE.0) RETURN
C                                    SET NC=NUMBER OF CHARS TO BE DECODED
      NC = 129
      IF (NCHARS.LT.NC) NC=NCHARS
      NWORDS = NC+NC+NC+NC
      J = 1
      DO 110 I = 1,NWORDS,4
      UNPAKD(I) = PACKED(J)
      UNPAKD(I+1) = LBLANK
      UNPAKD(I+2) = LBLANK
      UNPAKD(I+3) = LBLANK
  110 J = J+1

C  CHECK UNPAKD ARRAY AND SET NCHMTB BASED ON TRAILING BLANKS FOUND

      DO 200 N = 1,NWORDS,4
         NN = NWORDS - N - 2
         LBYTE = UNPAKD(NN)
         IF(IBYTE .NE. IBLANK) GO TO 210
  200 CONTINUE
      NN = 0
  210 NCHMTB = (NN + 3) / 4
      RETURN
      END


      SUBROUTINE PARTIII(XO,A,M,XINC,MAT,COV,OBJF)
C
C      *****************************************************************
C        SCORE VECTOR AND SAMPLE INFORMATION MATRIX CALCULATIONS FOR:
C                 PARETO MODEL, MODEL-I AND MODEL-II
C      *****************************************************************
C
C      THIS SUBROUTINE EVALUATES THE FIRST AND SECOND PARTIAL
C      DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION FOR THE PARETO
C      MODEL, MODEL-I AND MODEL-II
C
C      THE VARIABLES IN THIS SUBROUTINE ARE DEFINED AS FOLLOWS:

C       XO(25) = VECTOR OF THE UPDATED PARAMETER ESTIMATES WITH:
C
C       XO(1) = MLE OF SHAPE PAR. OF THE GAMMA MIXING DISTRIBUTION.
C       XO(2) = MLE OF MEAN OF THE GAMMA MIXING DISTRIBUTION.
C       XO(J) = MLE OF THE j\030TH\022 REGRESSION COEFFICIENT IN THE
C                REGRESSION MODEL.
C
C      XINC(25) = NEGATIVE OF THE SCORE VECTOR
C      A(25,25) = P(25,25) = - MAT(25,25) = MATRIX OF SECOND PARTIAL
C                 DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION.
C      OBJF = THE OBJECTIVE FUNCTION TO BE MAXIMIZED
C             = LOG LIKELIHOOD FUNCTION.
C      MDL = A SCALAR INDICATOR OF MODEL TO BE USED IN ANALYSIS WITH
C             VALUES:    1=PARETO MODEL    2=MODEL-I     3=MODEL-II.
C      START = VECTOR OF STARTING VALUES FOR THE BASELINE PARAMETERS.
C
C      Gi = T(i,2) + T(i,3) - 1 = NUMBER OF TUMOR VISITS FOR THE
C            iTH PATIENT.
C      XB1 = Zi*Beta
C      XB = EXP(XB1)
C      SMI = T(I,4) + T(I,5) = TOTAL FOLLOW UP TIME.
C      DEN = XO(2)*XB*SMI + XO(1)
C      GB = Gi + XO(1)
C
C
       REAL*8       SM1,SM2,SM3,DEN,GI,OBJF,XB,XB1,MAT(25,25),
     +              SMI,A(25,25),XO(25),XINC(25),GB
       INTEGER      COV,S,M
       COMMON       XX(3000,30),XX1(3000,30),P(25,25),
     +              MDL, NSUBS, START(5), COVTEMP
C
C     CHECK IF THE UPDATED ESTIMATE OF THE SHAPE OR SCALE PARAMETERS IS
C     ZERO, IF SO THEN REPLACE THE UPDATED VALUE BY THE STARTING VALUE.
C
       IF(XO(1).LE. 0.0) XO(1) = START(1)
       IF(XO(2).LE. 0.0) XO(2) = START(2)
C
C      INITIALIZE THE SCORE VECTOR AND THE MATRIX OF SECOND PARTIAL
C      DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION.
C
       OBJF = 0.0D0
       DO 4 I = 1 , 25
         XINC(I) = 0.0D0
         DO 4 J = 1 , 25
           P(I,J) = 0.0D0
4      CONTINUE
C


       DO 10 I = 1 , NSUBS



C
C      INITIALIZE THE DIFFERENT SCALARS NEEDED IN CALCULATIONS.
C
         SM1=0.0D0
         SM2=0.0D0
         SM3=0.0D0
         DEN=0.0D0
         GI  = 0.0D0
         GB  = 0.0D0
         SMI= 0.0D0
         XB  =1.0D0
         XB1=0.0D0
C
C      SET UP THE VALUES OF Gi AND SMI. THE VALUES OF THESE
C      QUANTITIES DEPEND ON THE MODEL CHOSEN.
C

           GI = XX1(I,1)
           SMI = XX1(I,2)
C
C      IF THE NUMBER OF FAILURES FOR THE ith PATIENT IS > 0 THEN
C      COMPUTE SM1, SM2 AND SM3.
C

         IF ( GI.GT. 0 ) THEN
           DO 2 J = 1 , GI
              SM1 = SM1 + DLOG( XO(1)+J-1 )
              SM2 = SM2 + 1/(XO(1)+J-1)
              SM3 = SM3 + 1/((XO(1)+J-1)**2)
```

```
2         CONTINUE
        ENDIF
C
C     COMPUTE THE SCALAR XB1, XB, DEN, GB, SCORE VECTOR AND THE
C     NEGATIVE OF THE SAMPLE INFORMATION MATRIX.
C
        DO 3 J = 1 , COV
          XB1 = XB1 + XX1(I,J+2)*XO(J+2)
3       CONTINUE
        XB=DEXP(XB1)
C
        DEN = XB*SMI*XO(2) + XO(1)
        GB  = XO(1) + GI
C


        XINC(1)=XINC(1) - ( SM2 - DLOG(DEN) +
     +                   (1+DLOG(XO(1)) - GB/DEN)    )
C
        XINC(2)=XINC(2) - ( GI/XO(2) - GB*XB*SMI/DEN )
C
        DO 5 J = 1 , COV
          XINC(J+2)=XINC(J+2) - (  GI*XX1(I,J+2) -
     +                        GB*XO(2)*XX1(I,J+2)*XB*SMI/DEN )
5       CONTINUE
C
C
        P(1,1)=P(1,1) - SM3 + ( 1/XO(1) - (2*DEN - GB)/(DEN**2) )
        P(1,2)=P(1,2) - XB*SMI/DEN + GB*SMI*XB/(DEN**2)
        DO 7 J = 1 , COV
          P(1,J+2)=P(1,J+2) - XO(2)*XB*SMI*XX1(I,J+2)/DEN +
     +                 GB*XO(2)*SMI*XX1(I,J+2)*XB/(DEN**2)
7       CONTINUE
C
        P(2,2) = P(2,2) - GI/(XO(2)**2) + GB*( (XB*SMI/DEN)**2 )
        DO 8 J = 1 , COV
          P(2,J+2)=P(2,J+2) - XO(1)*GB*XX1(I,J+2)*XB*SMI/(DEN**2)
8       CONTINUE
C
        DO 9 S = 1 , COV
          DO 9 K = S , COV
         P(S+2,K+2)=P(S+2,K+2) - XO(1)*XO(2)*GB*XX1(I,S+2)
     +                 *XX1(I,K+2)*XB*SMI/(DEN**2)
9         CONTINUE
C


        OBJF = OBJF + GI*( DLOG(XO(2)) + XB1 ) + SM1 -
     +                      GB*DLOG( DEN ) + XO(1)*DLOG(XO(1))
C
10     CONTINUE
C
       DO 11 J = 1 , M
         DO 11 I = J , M
           P(I,J) = P(J,I)
11     CONTINUE
C
       DO 12 I = 1 , M
         DO 12 J = 1 , M
           A(I,J) = P(I,J)
           MAT(I,J) = -P(I,J)
12     CONTINUE
C

       RETURN
       END
```

```
      SUBROUTINE ARINV ( M,MAT )
C
      INTEGER          IJOB,N,IA,IER
      REAL*8           MAT(M,M),D1,D2,WKAREA(30),B(1)

      N     = M
      IA    = 25
      IJOB  =  1
      D1    = -1.0
C
      CALL LINV3F( MAT,B,IJOB,N,IA,D1,D2,WKAREA,IER )
C
      RETURN
      END

      SUBROUTINE LINV3F (A,B,IJOB,N,IA,D1,D2,WKAREA,IER)
C
      DOUBLE PRECISION   A(IA,1),B(1),WKAREA(1),C1,C2,D1,D2,WA,ZERO,
     *                   ONE,SUM,C
      DATA               ZERO/0.0D0/,ONE/1.0D0/
C                                 FIRST EXECUTABLE STATEMENT
C                                 LU DECOMPOSITION OF A
      CALL LUDATN (A,IA,N,A,IA,0,C1,C2,WKAREA,WKAREA,WA,IER)
      IF (D1 .LT. ZERO .AND. IJOB .GE. 1 .AND. IJOB .LT. 4) GO TO 5
      D1 = C1
      D2 = C2
    5 IF (IER .GE. 128) GO TO 60
      IF (IJOB .LE. 0 .OR. IJOB .GT. 4) GO TO 55
C                                 SOLVE AX = B
      IF (IJOB .EQ. 2 .OR . IJOB .EQ. 3) CALL LUELMN (A,IA,N,B,WKAREA,B)
      IF (IJOB .NE. 1 .AND. IJOB .NE. 3) GO TO 9005
C                                 MATRIX INVERSION
      A(N,N) = ONE/A(N,N)
      NM1 = N-1
      IF (NM1 .LT. 1) GO TO 9005
      DO 40 II=1,NM1
         L = N-II
         M = L+1
         DO 15 I=M,N
            SUM = ZERO
            DO 10 K=M,N
               SUM = SUM-A(I,K)*A(K,L)
   10       CONTINUE
            WKAREA(N+I) = SUM
   15    CONTINUE
         DO 20 I=M,N
            A(I,L) = WKAREA(N+I)
   20    CONTINUE
         DO 30 J=L,N
            SUM = ZERO
            IF (J .EQ. L) SUM = ONE
            DO 25 K=M,N
               SUM = SUM-A(L,K)*A(K,J)
   25       CONTINUE
            WKAREA(N+J) = SUM/A(L,L)
   30    CONTINUE
         DO 35 J=L,N
            A(L,J) = WKAREA(N+J)
   35    CONTINUE
   40 CONTINUE
C                                 PERMUTE COLUMNS OF A INVERSE
      DO 50 I=1,N
         J = N-I+1
         JP = WKAREA(J)
         IF (J .EQ. JP) GO TO 50
         DO 45 K=1,N
            C = A(K,JP)
```

```
            A(K,JP) = A(K,J)
            A(K,J) = C
   45    CONTINUE
   50 CONTINUE
      GO TO 9005
   55 CONTINUE
C     WARNING WITH FIX - IJOB WAS SET INCORRECTLY
      IER = 65
      GO TO 9000
C     TERMINAL ERROR - MATRIX A IS  ALGORITHMICALLY SINGULAR
   60 IER = 130
 9000 CONTINUE
      CALL UERTST(IER,6HLINV3F)
 9005 RETURN
      END

      SUBROUTINE LUDATN (A,IA,N,LU,ILU,IDGT,D1,D2,APVT,EQUIL,WA,IER)

      DIMENSION          A(IA,1),LU(ILU,1),APVT(1),EQUIL(1)
      DOUBLE PRECISION   A,LU,D1,D2,EQUIL,WA,ZERO,ONE,FOUR,SIXTN,SIXTH,
     *                   RN,WREL,BIGA,BIG,P,SUM,AI,WI,T,TEST,Q,APVT
      DATA               ZERO,ONE,FOUR,SIXTN,SIXTH/0.D0,1.D0,4.D0,
     *                   16.D0,.0625D0/
C                                  FIRST EXECUTABLE STATEMENT
C                                  INITIALIZATION
      IER = 0
      RN = N
      WREL = ZERO
      D1 = ONE
      D2 = ZERO
      BIGA = ZERO
      DO 10 I=1,N
         BIG = ZERO
         DO 5 J=1,N
            P = A(I,J)
            LU(I,J) = P
            P = DABS(P)
            IF (P .GT. BIG) BIG = P
    5    CONTINUE
         IF (BIG .GT. BIGA) BIGA = BIG
         IF (BIG .EQ. ZERO) GO TO 110
         EQUIL(I) = ONE/BIG
   10 CONTINUE
      DO 105 J=1,N
         JM1 = J-1
         IF (JM1 .LT. 1) GO TO 40
C                                  COMPUTE U(I,J), I=1,...,J-1
         DO 35 I=1,JM1
            SUM = LU(I,J)
            IM1 = I-1
            IF (IDGT .EQ. 0) GO TO 25
C                                  WITH ACCURACY TEST
            AI = DABS(SUM)
            WI = ZERO
            IF (IM1 .LT. 1) GO TO 20
            DO 15 K=1,IM1
               T = LU(I,K)*LU(K,J)
               SUM = SUM-T
               WI = WI+DABS(T)
   15       CONTINUE
            LU(I,J) = SUM
   20       WI = WI+DABS(SUM)
            IF (AI .EQ. ZERO) AI = BIGA
            TEST = WI/AI
            IF (TEST .GT. WREL) WREL = TEST
            GO TO 35
C                                  WITHOUT ACCURACY
   25       IF (IM1 .LT. 1) GO TO 35
            DO 30 K=1,IM1
               SUM = SUM-LU(I,K)*LU(K,J)
   30       CONTINUE
            LU(I,J) = SUM
   35    CONTINUE
   40    P = ZERO
C        COMPUTE U(J,J) AND L(I,J), I=J+1,...,
         DO 70 I=J,N
            SUM = LU(I,J)
            IF (IDGT .EQ. 0) GO TO 55
C                                  WITH ACCURACY TEST
            AI = DABS(SUM)
            WI = ZERO
            IF (JM1 .LT. 1) GO TO 50
            DO 45 K=1,JM1
               T = LU(I,K)*LU(K,J)
               SUM = SUM-T
               WI = WI+DABS(T)
   45       CONTINUE
            LU(I,J) = SUM
   50       WI = WI+DABS(SUM)
            IF (AI .EQ. ZERO) AI = BIGA
            TEST = WI/AI
            IF (TEST .GT. WREL) WREL = TEST
            GO TO 65
C                                  WITHOUT ACCURACY TEST
   55       IF (JM1 .LT. 1) GO TO 65
            DO 60 K=1,JM1
               SUM = SUM-LU(I,K)*LU(K,J)
   60       CONTINUE
            LU(I,J) = SUM
   65       Q = EQUIL(I)*DABS(SUM)
            IF (P .GE. Q) GO TO 70
            P = Q
            IMAX = I
   70    CONTINUE
C        TEST FOR ALGORITHMIC SINGULARITY
         IF (RN+P .EQ. RN) GO TO 110
         IF (J .EQ. IMAX) GO TO 80
C        INTERCHANGE ROWS J AND IMAX
         D1 = -D1
         DO 75 K=1,N
            P = LU(IMAX,K)
            LU(IMAX,K) = LU(J,K)
            LU(J,K) = P
   75    CONTINUE
         EQUIL(IMAX) = EQUIL(J)
   80    APVT(J) = IMAX
         D1 = D1*LU(J,J)
   85    IF (DABS(D1) .LE. ONE) GO TO 90
         D1 = D1*SIXTH
         D2 = D2+FOUR
         GO TO 85
   90    IF (DABS(D1) .GE. SIXTH) GO TO 95
         D1 = D1*SIXTN
         D2 = D2-FOUR
         GO TO 90
   95    CONTINUE
         JP1 = J+1
         IF (JP1 .GT. N) GO TO 105
C        DIVIDE BY PIVOT ELEMENT U(J,J)
         P = LU(J,J)
         DO 100 I=JP1,N
            LU(I,J) = LU(I,J)/P
  100    CONTINUE
  105 CONTINUE
```

```
C     PERFORM ACCURACY TEST
      IF (IDGT .EQ. 0) GO TO 9005
      P = 3*N+3
      WA = P*WREL
      IF (WA+10.D0**(-IDGT) .NE. WA) GO TO 9005
      IER = 34
      GO TO 9000
C     ALGORITHMIC SINGULARITY
  110 IER = 129
      D1 = ZERO
      D2 = ZERO
 9000 CONTINUE
C                                 PRINT ERROR
      CALL UERTST(IER,6HLUDATN)
 9005 RETURN
      END

      SUBROUTINE LUELMN (A,IA,N,B,APVT,X)
C
      DIMENSION          A(IA,1),B(1),APVT(1),X(1)
      DOUBLE PRECISION   A,B,X,SUM,APVT
C                  FIRST EXECUTABLE STATEMENT
C                  SOLVE LY = B FOR Y
      DO 5 I=1,N
    5 X(I) = B(I)
      IW = 0
      DO 20 I=1,N
         IP = APVT(I)
         SUM = X(IP)
         X(IP) = X(I)
         IF (IW .EQ. 0) GO TO 15
         IM1 = I-1
         DO 10 J=IW,IM1
            SUM = SUM-A(I,J)*X(J)
   10    CONTINUE
         GO TO 20
   15    IF (SUM .NE. 0.D0) IW = I
   20 X(I) = SUM
C     SOLVE UX = Y FOR X
      DO 30 IB=1,N
         I = N+1-IB
         IP1 = I+1
         SUM = X(I)
         IF (IP1 .GT. N) GO TO 30
         DO 25 J=IP1,N
            SUM = SUM-A(I,J)*X(J)
   25   CONTINUE
   30 X(I) = SUM/A(I,I)
      RETURN
      END

      SUBROUTINE UERTST (IER,NAME)
C       SPECIFICATIONS FOR ARGUMENTS
      INTEGER            IER
      INTEGER            NAME(1)
C      SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            I,IEQ,IEQDF,IOUNIT,LEVEL,LEVOLD,NAMEQ(6),
     *                   NAMSET(6),NAMUPK(6),NIN,NMTB
      DATA               NAMSET/1HU,1HE,1HR,1HS,1HE,1HT/
      DATA               NAMEQ/6*1H /
      DATA               LEVEL/4/,IEQDF/0/,IEQ/1H=/
C                                  UNPACK NAME INTO NAMUPK
C                                  FIRST EXECUTABLE STATEMENT
      CALL USPKD (NAME,6,NAMUPK,NMTB)
C                                  GET OUTPUT UNIT NUMBER
      CALL UGETIO(1,NIN,IOUNIT)
C                                  CHECK IER
      IF (IER.GT.999) GO TO 25
      IF (IER.LT.-32) GO TO 55
      IF (IER.LE.128) GO TO 5
      IF (LEVEL.LT.1) GO TO 30
C                                  PRINT TERMINAL MESSAGE
      IF (IEQDF.EQ.1) WRITE(IOUNIT,35) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,35) IER,NAMUPK
      GO TO 30
    5 IF (IER.LE.64) GO TO 10
      IF (LEVEL.LT.2) GO TO 30
C                                  PRINT WARNING WITH FIX MESSAGE
      IF (IEQDF.EQ.1) WRITE(IOUNIT,40) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,40) IER,NAMUPK
      GO TO 30
   10 IF (IER.LE.32) GO TO 15
C                                  PRINT WARNING MESSAGE
      IF (LEVEL.LT.3) GO TO 30
      IF (IEQDF.EQ.1) WRITE(IOUNIT,45) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,45) IER,NAMUPK
      GO TO 30
   15 CONTINUE
C                                  CHECK FOR UERSET CALL
      DO 20 I=1,6
         IF (NAMUPK(I).NE.NAMSET(I)) GO TO 25
   20 CONTINUE
      LEVOLD = LEVEL
      LEVEL = IER
      IER = LEVOLD
      IF (LEVEL.LT.0) LEVEL = 4
      IF (LEVEL.GT.4) LEVEL = 4
      GO TO 30
   25 CONTINUE
      IF (LEVEL.LT.4) GO TO 30
C                PRINT NON-DEFINED MESSAGE
      IF (IEQDF.EQ.1) WRITE(IOUNIT,50) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,50) IER,NAMUPK
   30 IEQDF = 0
      RETURN
   35 FORMAT(19H *** TERMINAL ERROR,10X,7H(IER = ,I3,
     1       20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
   40 FORMAT(27H *** WARNING WITH FIX ERROR,2X,7H(IER = ,I3,
     1       20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
   45 FORMAT(18H *** WARNING ERROR,11X,7H(IER = ,I3,
     1       20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
   50 FORMAT(20H *** UNDEFINED ERROR,9X,7H(IER = ,I5,
     1       20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
C
C                  SAVE P FOR P = R CASE
C                  P IS THE PAGE NAMUPK
C                  R IS THE ROUTINE NAMUPK
   55 IEQDF = 1
      DO 60 I=1,6
   60 NAMEQ(I) = NAMUPK(I)
   65 RETURN
      END

      SUBROUTINE UGETIO(IOPT,NIN,NOUT)
C              SPECIFICATIONS FOR ARGUMENTS
      INTEGER            IOPT,NIN,NOUT
C              SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            NIND,NOUTD
      DATA               NIND/5/,NOUTD/6/
C              FIRST EXECUTABLE STATEMENT
      IF (IOPT.EQ.3) GO TO 10
      IF (IOPT.EQ.2) GO TO 5
      IF (IOPT.NE.1) GO TO 9005
      NIN = NIND
```

```
      NOUT = NOUTD
      GO TO 9005
    5 NIND = NIN
      GO TO 9005
   10 NOUTD = NOUT
 9005 RETURN
      END
```