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

by

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

1This 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 $p \ (\geq 1)$ “regressor”, “explanatory” or “covariate” values, denoted by the row $p$-vector $x_i$, 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 $\mu_i = \theta_i T_i \exp(x_i \beta)$, i.e. proportional to the follow-up time $T_i$. Here $\beta$ is a (column) $p$-vector of regression coefficients, representing the effect of the regressor variables, and $\theta_i$ is the baseline ($x_i = 0$) value or “underlying” event rate for the $i$th subject. Equivalently, we may say that the events for individual $i$ occur according as a Poisson process with event rate $\theta_i \exp(x_i \beta)$. Another characterization is that the inter-event times are independent exponentially distributed random variables with means given by $[\theta_i \exp(x_i \beta)]^{-1}$.

Subject heterogeneity (or “extra Poisson variation”) that is not explained by the regressor variables is modeled by allowing the $\theta_i$ values to vary from subject to subject according to a gamma distribution with scale parameter $\gamma$ and shape parameter $\nu$ (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 $\nu \to \infty$ so that the $\theta_i$ 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 $\theta_i$, the contribution to the likelihood for the $i$th subject is

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

where $\mu_i = \theta_i T_i \exp(x_i \beta)$,

The full likelihood is then given by
\[ L(\gamma, \nu, \beta) = \prod_{i=1}^{I} \int_{0}^{\infty} L_i(\beta_i;\theta_i) g(\theta_i; \gamma, \nu) \, d\theta_i \]

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

Carrying out the integration we obtain:

\[
L(\gamma, \nu, \beta) = \prod_{i=1}^{I} \frac{\Gamma(n_i + \nu)}{\Gamma(\nu)} \left[ \frac{\gamma \exp(x_i; \beta)}{\gamma T_i \exp(x_i; \beta) + 1} \right]^{n_i} \]

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 \( \nu \) rather than \( \gamma \) and \( \nu \). Replacing \( \gamma \) by \( \mu/\nu \) in (1), we write the resulting likelihood as \( L(\mu, \nu, \beta) \).

Estimates of the mixing distribution parameters \( \mu, \nu \) (equivalently \( \gamma, \nu \)) and of the regression coefficients \( \beta = (\beta_1, \ldots, \beta_p) \) can be found by maximizing the likelihood (1), or equivalently its logarithm \( L = \mathcal{L} \), 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 \( \mathcal{L}(\mu, \nu, \beta) \). 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 i-th 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 ni experienced by the i-th subject; the third item is the observation, followup or exposure time Ti 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 \( \mu \) and \( \nu \) are obtained by the method of moments, with \( \beta = 0 \). 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 \( \mu \) and \( \nu \) 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 \( \nu \) and the mean \( \mu = \nu \gamma \) of the mixing distribution along with their standard errors.

- The deviance, i.e. \(-2\mathcal{L}(\hat{\mu}, \hat{\nu}, \hat{\beta})\). For comparison, the deviance of the null model (i.e. that without any covariates \( \beta = 0 \)) 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 $(\hat{\mu}, \hat{\nu}, \hat{\beta})$ 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 = ( 1.61572)  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 STATISTIC = 24.02362  WITH 1 DEGREE(S) OF FREEDOM.

<table>
<thead>
<tr>
<th>COVARIATE</th>
<th>M.L.E.</th>
<th>ST.DEV.</th>
<th>Z-VALUE</th>
<th>95% C. INTERVAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>TREATMENT</td>
<td>0.823024*</td>
<td>0.212742</td>
<td>3.8687</td>
<td>( 0.406 , 1.240)</td>
</tr>
</tbody>
</table>

*: 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

<table>
<thead>
<tr>
<th>RECORD 1</th>
<th>0.0000</th>
<th>2186.0000</th>
<th>2.0000</th>
<th>0.0000</th>
<th>1.0000</th>
<th>0.0000</th>
<th>65.0000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.0000</td>
<td>4.0000</td>
<td>6.0000</td>
<td>0.0000</td>
<td>5.0000</td>
<td>122.0000</td>
<td>3.0000</td>
</tr>
<tr>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RECORD 2</th>
<th>0.0000</th>
<th>169.0000</th>
<th>2.0000</th>
<th>1.0000</th>
<th>2.0000</th>
<th>0.0000</th>
<th>62.3000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.0000</td>
<td>4.0000</td>
<td>5.0000</td>
<td>1.0000</td>
<td>7.5000</td>
<td>104.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td></td>
<td>0.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RECORD 3</th>
<th>0.0000</th>
<th>967.0000</th>
<th>1.0000</th>
<th>1.0000</th>
<th>1.0000</th>
<th>1.0000</th>
<th>90.9000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0000</td>
<td>6.0000</td>
<td>5.0000</td>
<td>0.0000</td>
<td>6.0000</td>
<td>84.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RECORD 4</th>
<th>0.0000</th>
<th>2061.0000</th>
<th>2.0000</th>
<th>1.0000</th>
<th>1.0000</th>
<th>0.0000</th>
<th>102.3000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0000</td>
<td>5.0000</td>
<td>5.0000</td>
<td>0.0000</td>
<td>5.5000</td>
<td>107.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
RECORD 5

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>1716.0000</td>
<td>2.0000</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>70.9000</td>
</tr>
<tr>
<td>0.0000</td>
<td>4.0000</td>
<td>5.0000</td>
<td>0.0000</td>
<td>5.5000</td>
<td>101.0000</td>
<td>11.0000</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

QUICK DATA SUMMARY: MEAN MIN MAX # MISSING

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>Min</th>
<th>Max</th>
<th># Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>N_Tumors</td>
<td>.538</td>
<td>.0</td>
<td>13.0</td>
<td>0</td>
</tr>
<tr>
<td>Exp_Time</td>
<td>1583.545</td>
<td>1.0</td>
<td>3276.0</td>
<td>0</td>
</tr>
<tr>
<td>T_Group</td>
<td>1.501</td>
<td>1.0</td>
<td>2.0</td>
<td>0</td>
</tr>
<tr>
<td>AGE&gt;64</td>
<td>.525</td>
<td>.0</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>GENDER M=1</td>
<td>1.255</td>
<td>1.0</td>
<td>2.0</td>
<td>0</td>
</tr>
<tr>
<td>Cur_Smoker</td>
<td>0.284</td>
<td>.0</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>Weight_Kg</td>
<td>77.338</td>
<td>40.5</td>
<td>143.2</td>
<td>1</td>
</tr>
<tr>
<td>Farm_GE_15</td>
<td>0.344</td>
<td>.0</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>Sunburn(a)</td>
<td>4.741</td>
<td>4.0</td>
<td>6.0</td>
<td>0</td>
</tr>
<tr>
<td>Suntan(b)</td>
<td>4.836</td>
<td>4.0</td>
<td>6.0</td>
<td>0</td>
</tr>
<tr>
<td>Sunscrn&lt;5</td>
<td>0.282</td>
<td>.0</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>Sun_Damage</td>
<td>4.733</td>
<td>1.0</td>
<td>9.0</td>
<td>0</td>
</tr>
<tr>
<td>Sel</td>
<td>114.243</td>
<td>45.0</td>
<td>220.0</td>
<td>0</td>
</tr>
<tr>
<td>#Prev_BCE</td>
<td>2.898</td>
<td>.0</td>
<td>66.0</td>
<td>0</td>
</tr>
<tr>
<td>#Prev_SCC</td>
<td>0.979</td>
<td>.0</td>
<td>16.0</td>
<td>0</td>
</tr>
<tr>
<td>In_Patient</td>
<td>0.458</td>
<td>.0</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>Clinic_#</td>
<td>3.165</td>
<td>1.0</td>
<td>8.0</td>
<td>0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Shape</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHAPE = 0.36653</td>
<td>MEAN = 0.000342</td>
</tr>
<tr>
<td>ST.DEV = ( 0.03779)</td>
<td>ST.DEV = ( 0.000021)</td>
</tr>
</tbody>
</table>

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

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

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)

DEVIANE: 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.

<table>
<thead>
<tr>
<th>COVARIATE</th>
<th>M.L.E.</th>
<th>ST.DEV.</th>
<th>Z-VALUE</th>
<th>95% C. INTERVAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Se1</td>
<td>-0.007171*</td>
<td>0.002801</td>
<td>-2.5602</td>
<td>(-0.013 , -0.002)</td>
</tr>
<tr>
<td>AGE&gt;64</td>
<td>0.440036*</td>
<td>0.122981</td>
<td>3.5781</td>
<td>( 0.199 , 0.681)</td>
</tr>
<tr>
<td>GENDER M=1</td>
<td>-0.918275*</td>
<td>0.159514</td>
<td>-5.7567</td>
<td>(-1.231 , -0.606)</td>
</tr>
</tbody>
</table>

*: 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

PREVIOUS MODEL(S) FITTED

MDL  0: NULL MODEL

DEVIANE:  11880.57057

15
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.

<table>
<thead>
<tr>
<th>COVARIATE</th>
<th>M.L.E.</th>
<th>ST.DEV.</th>
<th>Z-VALUE</th>
<th>95% C. INTERVAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Se1</td>
<td>-0.007165*</td>
<td>0.002801</td>
<td>-2.5562</td>
<td>(-0.013, 0.002)</td>
</tr>
<tr>
<td>AGE&gt;64</td>
<td>0.065689</td>
<td>0.397167</td>
<td>0.1654</td>
<td>(-0.713, 0.844)</td>
</tr>
<tr>
<td>GENDER M=1</td>
<td>-1.096237*</td>
<td>0.242267</td>
<td>-4.5249</td>
<td>(-1.571, -0.621)</td>
</tr>
<tr>
<td>AGE&gt;*GEND</td>
<td>0.316835</td>
<td>0.320077</td>
<td>0.9899</td>
<td>(-0.311, 0.944)</td>
</tr>
</tbody>
</table>

*: SIGNIFICANT AT THE 5% LEVEL

DO YOU WISH TO ADD COVARIATES TO THE MODEL? (Y/W) n
References


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

<table>
<thead>
<tr>
<th>ID</th>
<th>No. of tumors</th>
<th>Observation time</th>
<th>Treatment group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>5</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>5</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>21</td>
<td>6</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>7</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>25</td>
<td>11</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>26</td>
<td>9</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>27</td>
<td>2</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>28</td>
<td>9</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>29</td>
<td>4</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>30</td>
<td>6</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>31</td>
<td>7</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>32</td>
<td>6</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>33</td>
<td>1</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>34</td>
<td>13</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>35</td>
<td>2</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>36</td>
<td>1</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>37</td>
<td>10</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>38</td>
<td>4</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>39</td>
<td>5</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>40</td>
<td>11</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>41</td>
<td>11</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>42</td>
<td>9</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>43</td>
<td>12</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>44</td>
<td>1</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>45</td>
<td>3</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>46</td>
<td>1</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>47</td>
<td>3</td>
<td>122</td>
<td>2</td>
</tr>
<tr>
<td>48</td>
<td>3</td>
<td>122</td>
<td>2</td>
</tr>
</tbody>
</table>
Figure 1: Flow Chart for the Program
**Program Run**


1

---

**Definitions of Important Program Variables**

- **A(1)**: Negative of Sample Information Matrix
- **BASE**: Base model that user wants to build on
- **BASENULL()**: Contains MLE of base model
- **C1,C2**: Contains information on number of variables in base model
- **CHI**: Chi square statistic for testing significance of
- **COVNAME()**: Names of covariates in data file
- **COVTEMP**: Number of covariates entering analysis
- **COL(25)**: Storage for covariate indices entering analysis
- **C
- **cular**: Names of covariates available for analysis.
- **DICTON**: Dictionary file name
- **DICTION**: DICTIONARY FILE NAME
- **DLGJK**: Deviance for null model
- **EPS**: Radius of convergence
- **FILE**: IF ANY MISSING ENTRIES FOR ANY COVARIATE
- **GMA(1)**: Names of baseline parameters.
- **GMEAN**: Mean number of tumors (used in mom estimation)
- **ICOL**: > 0 IF ANY MODEL OTHER THAN THE NULL HAS BEEN FITTED
- **ID**: Patient descriptor
- **INPUT**: File containing covariate data.
- **INTERDE()**: Indices of variables appearing in interactions
- **LKRATIO**: Array of deviance of previous models
- **M**: Number of parameters to be estimated.
- **MAT(1)**: Working matrix for the subroutine ARIN.
- **MAXX**: Array of maximum values of each covariate
- **MDL**: Model choice pointer (=2)
- **MINN**(): Array of minimum of each covariate
- **MISSING()**: Array of number missing for each covariate
- **MODINT()**: Array of previous interactions fitted
- **MODVAR()**: Array of previous covariates fitted
- **MODVAR**: The size of MODVAR and MODINT limits one to 25 models.
- **NCOV**: Number of covariates entering analysis.
- **NINTS**: Number of pairwise-interactions in the analysis
- **NNRUN**: Number of models allowed in one session (25)

---

**The Following Definitions Are for Using the Optimization Routines in NAG. See NAG Manual for a Description of These Variables.**

- **N**: NCLIN, NCON
- **IC**: NCOA, NRCAP, NROW
- **INTEGER**: LWORK, LWORK
- **DOUBLE PRECISION**: BIGN, OBIF
- **INTEGER**: IFAIL, ITER
- **DOUBLE PRECISION**: AA(1,25), BL(25), BU(25), C(1), CJAC(1,25), CLAMDA(25), R(2,25), WORK(2000)
C SET ALL THE VARIABLES THAT WILL BE USED IN THE NAG SUBROUTINE

NRNMR=25
NCLIN=0
NCNLR=0
NRNA=1
NRCR=1
LIMLRK=100
LIW=2000
BIGBND=1.0E10

2000 CONTINUE

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

BL(1)=0.0000001
BL(2)=0.0000001

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

DO 1994 I = 1, 13000
   RESP(I,1)=0
   RESP(I,2)=0
   DO 1994 J = 1, 30
      XX(I,J)=0
   1994 CONTINUE

DO 1996 I = 1, 125
   XXI(I,J)=0
   MODVAR(I,J)=0
   P(I,J)=0
   MODINT(I,J)=0
   PARSER(I,J)=0
   MODVAR(I,J)=0
   P(I,J)=0

1996 CONTINUE

2000 C

NUMISS=0
MISS=0
MDL=3
BASE = 0
BASLULL(1)=0
BASLULL(2)=0
BASLULL(3)=0
GMA(1) = 'SHAPE'

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

WRITE(*,8010)

8010 FORMAT(1X,'WHAT IS THE DICTIONARY FILE? ',A)
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(*,*)

8011 FORMAT(1X,'SPECIFY FILE FOR PRINTED OUTPUT: ',A)
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,A) 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
   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 (.not. (resp(i,1) .ge. 0) .and. .not. (resp(i,2) .ge. 0)) then
   READ(28,*) ID, (XX(k,j), J = 1,NVARS+2)
do 9991 j=1, nvars+2
   x(k,j) = xx(k,j)
k=k+1
else
   miss=miss+1
endif
1 continue
write(*,8012)
write(*,601) miss
601 FORMAT(1X,13,1X, 'RECORDS ARE BEING IGNORED DUE TO MISSING', + 1X, 'NUMBER OF EVENTS', /,4X,'AND/OR EXPOSURE TIME')
write(*,*)
naube=naube-miss
C PRINT FIRST FIVE RECORDS FOR DATA VERIFICATION IF REQUESTED
WRITE(*,8012)
8012 FORMAT(1X, 'PRINT FIRST FIVE RECORDS? (Y/N): ',$,)
read(*, '(A1)') ans
   IF (ANS .EQ. 'Y') .OR. (ANS .EQ. 'Y') .OR.
       (ANS .EQ. 'YES') .OR. (ANS .EQ. 'yes') THEN
      DO 1995 I = 1, 5
         WRITE(*,7093) I
            7093 FORMAT(1X,'RECORD ', I2)
               WRITE(*,7(F10.4,1X)) (XX(I,J), J = 1,NVARS+2)
               WRITE(*,*)
      1995 ENDIF
   C CLOSE (28)
   C CALCULATE MIN, MAX, NUMBERS MISSING AND SUM FOR EACH VARIABLE
   C INITIALIZE THE ARRAYS MAXX, MINN, MISSING AND SUM
   C THE CODE FOR MISSING COVARIATES IN THE DATA FILE SHOULD BE -999
   C IF AN ENTRY IS NEGATIVE FOR THE N_TUMORS AND EXP_TIME, IT IS
   C CONSIDERED AS MISSING.
   DO 901 J = 1,NVARS+2
      MISSING(J)=0
      MAXX(J) = -100
      MINN(J) = 1000
      SUM(J) = 0
   901 DO 913 I = 1, NSUBS
      IF (J .EQ. 1) .OR. (J .EQ. 2)) THEN
         DO 901 J = 1,NVARS+2
            MAXX(J) = MAXX(J), XX(I,J)
            MINN(J) = MINN(J), XX(I,J)
            SUM(J) = SUM(J) + XX(I,J)
         ELSE
            IF (XX(I,J) .LT. -999) THEN
               MAXX(J) = MAXX(J), XX(I,J)
               MINN(J) = MINN(J), XX(I,J)
               SUM(J) = SUM(J) + XX(I,J)
            ELSE
               MISSING(J) = MISSING(J) + 1
         ENDIF
C DISPLAY BRIEF DATA SUMMARY
WRITE(*,*)
911 FORMAT(1X 'QUICK DATA SUMMARY: ',5X,'MEAN',9X, 'MIN', + 9X, 'MAX',9X, 'MISSING')
WRITE(*,922) NVARS, SUM(J)/NSUBS, MINN(J), + MAXX(J), MISSING(J)
WRITE(*,922) EXP_TIME, SUM(J)/NSUBS, MINN(J), + MAXX(J), MISSING(J)
WRITE(*,922) (COWNAME(J), SUM(J))/NSUBS-MISSING(J), + MINN(J), MAXX(J), MISSING(J), J=1,NVARS
922 FORMAT(1X,A8,F9.3,4X,F9.1,4X,F9.1,8X,F4.0)
WRITE(*,*)
C CHECK TO SEE IF NO OF TUMORS OR EXPOSURE TIME IS MISSING.
C DELETE ALL RECORDS THAT HAVE EITHER OF THE ABOVE MISSING.
C ICOL = 0 IF NO MODELS HAVE BEEN FITTED YET IN THIS SESSION; I O/W.
   ICOL = 0
   17 WRITE(*,2003)
2003 FORMAT(1X 'DO YOU WISH TO CONTINUE? (Y/N) ',$,)
read(*, '(A1)') 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 PARAMETERS OF BASELINE MODEL USING
   C METHOD OF MOMENTS ESTIMATION
   C TOTAL = TOTAL + XX(I,1)
GMEAN=TOTAL/NSUBS
DO 7071 I = 1, NSUBS
VAR = VAR + (XX(I,1) - GMEAN)**2/(XX(I,1) - GMEAN)
X0(1) = (VAR/TOTAL) - 1.0
X0(2) = GMEAN
N=2
N=M
C SET ERROR FLAG FOR NAG EQUAL TO -1. SEE MANUAL FOR AN EXPLANATION.
IPAIL=-1
COVTEMP=0
WRITE(*,*)
WRITE(*,7075)
WRITE(*,*) 'NULL MODEL....PLEASE WAIT'
7075 FORMAT(1X 'MAXIMIZING THE LIKELIHOOD FUNCTION FOR THE',IX, + 'MODEL WITHOUT ANY COVARIATES')
CALL E04UF('MAJOR PRINT LEVEL = 0')
7073 CALL E04UCF(N,NCLIN,NCNLN,NROWA,NROWJ,AA,BL,BU, + E04UM,OBJFUN,ITER,ISTATE,C,CLAMDA,OBJF,OBJGRD,IN, + X0,INWORK,INWORK,WORK,INWORK,INUSER,USER,IPAIL)
C PRINT ERROR MESSAGES DEPENDING ON THE VALUE OF IPAIL
IF (IPAIL .EQ. 0) THEN
WRITE(*,*)
WRITE(*,6666) ITER
WRITE(*,*)
ENDIF
6666 FORMAT(1X 'SUCCESSFUL CONVERGENCE HAS BEEN REACHED AFTER',I4, + 1X,'ITERATIONS')
IF (IPAIL .EQ. 1) THEN
WRITE(*,6971)
WRITE(*,*) 'EXITING THE PROGRAM'
GOTO 9
ENDIF
IF (IPAIL .EQ. 2) THEN
WRITE(*,6972)
WRITE(*,*) 'EXITING THE PROGRAM'
GOTO 9
ENDIF
IF (IPAIL .EQ. 3) THEN
WRITE(*,6973)
ENDIF
WRITE(*,*) 'EXITING THE PROGRAM'
GOTO 9
ENDIF
IF (IPAIL .EQ. 4) THEN
WRITE(*,*) 'ITERATION LIMIT EXCEEDED'
WRITE(*,*) 'EXITING THE PROGRAM'
GOTO 9
ENDIF
IF (IPAIL .EQ. 6) THEN
WRITE(*,6975)
WRITE(*,*) 'EXITING THE PROGRAM'
GOTO 9
ENDIF
IF (IPAIL .EQ. 7) THEN
WRITE(*,6976)
WRITE(*,*) 'EXITING THE PROGRAM'
GOTO 9
ENDIF
IF (IPAIL .EQ. 9) THEN
WRITE(*,6977)
WRITE(*,*) 'EXITING THE PROGRAM'
GOTO 9
ENDIF
C NEGATIVE 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 PARTII(XO,A,M,XINC,MAJ,OBJF)
CALL ARINV(M,MAT)
DO 7095 I = 1,M
IF (MAT(I,I) .LE. 0.0) THEN
WRITE(*,*) 'INFORMATION MATRIX IS SINGULAR'
GOTO 7078
ENDIF
7095 ST(I)=SQRT(MAT(I,I))
C WRITE TO THE OUTPUT FILE THE MLE AND SDES 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')
program

WRITE(25,*)
WRITE(25,*)
WRITE(*,8006)
WRITE(*,*)
WRITE(25,1015) X0(I),X0(J),ST(I),ST(J)
WRITE(25,*)
WRITE(*,1015) X0(I),X0(J),ST(I),ST(J)
WRITE(*,*)
DGLKL = -2*OBJF
IF(NCOV.EQ.0) THEN
  WRITE(25,1017) DGLKL
  WRITE(*,1017) DGLKL
ENDIF

C
STORE MIXING DIST MLE'S IN BASENULL()

DO 7095 I = 1,M
  BASENULL(I)=XO(I)
  DO 7099 I = 1,NSUBS
    DO 7097 J = 1,NVARS+2
      XX(I,J)=0
 7097 WRITE(*,*)
C
PROMPT FOR REGRESSION

WRITE(*,7077)
FORMAT('DO YOU WISH TO ADD COVARIATES TO THE MODEL? (Y/N) ',F,9)
READ(*,*) ANS
IF((ANS.EQ.‘Y’).OR.(ANS.EQ.‘y’)) THEN
  GOTO 7081
ENDIF
IF((ANS.EQ.‘N’).OR.(ANS.EQ.‘n’)) THEN
  GOTO 7091
ELSE
  WRITE(*,*) 'INVALID ANSWER'
ENDIF

C
SET UP COVARIATE SELECTION

7081 WRITE(*,*)
WRITE(*,*) 'INDEXED LIST OF COVARIATES'
WRITE(*,*) (5A(I4,A1,1X,A10)): (J,‘Y’,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'

DO 233 J = 1,NRUN
  WRITE(*,7091) J,(COVNAME(MCOVAR(J,J)+1),K+2,MCOVAR(J+1)+1)
  DO 234 K = 1,MODINT(J,1)
    WRITE(*,‘(1X,A5,A5,\”)’):COVNAME(MODINT(J,2*K+1),’\”)’),
    MODINT(MODINT(J,2*K)+1)
  2334 WRITE(*,*)
  WRITE(‘,’(,))
  WRITE(*,8013)
  FORMAT(‘ WHICH MODEL DO YOU WISH TO USE AS A BASE? ‘,F,9)
  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 = MODINT(BASE,1)
  C2 = MODINT(BASE,1)
ELSE
  C1=0
  C2=0
ENDIF

WRITE(*,8014)
FORMAT(‘ HOW MANY NEW VARIABLES DO YOU WISH TO ADD? ‘,F,9)
GOTO 2201

22 WRITE(*,7079) NVARS
2201 READ(*,*) NCOV
IF(BASE.GT.0) THEN
  DO 2205 K = 1,C1
    COL(K) = MODVAR(BASE,K+1)
 2205 IF(NCOV.GT.0) THEN
    WRITE(*,8015)
  ENDIF
  IF (NCOV.GT.0) THEN
    WRITE(*,8015)
ENDIF
8015 FORMAT(‘ ENTER INDEX OF CHOSEN COVARIATES ‘,F,9)
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
NCOV=HRUN+1,1 = NCOV
DO 2206 K = 1,NCOV
  MODVAR(HRUN+1,K+1) = COL(K)
2206 ICOL=ICOL+1
IF (ICOL .GT. 0) AND. (NCOV .GT. 1)) THEN
  WRITE(*,8016)
  GOTO 2202
ENDIF

8016 FORMAT('HOW MANY NEW INTERACTIONS? (0=none) ',$)

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

2202 FORMAT('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)
  ENDIF

  8018 FORMAT('Enter pair of indices for one interaction: ',$)

  20 READ(*,8018) INTERDEX(J,1),INTERDEX(J,2)

  MODINT(NRUN+1,1) = NINTS
  DO 2207 K = 1,NINTS
    MODINT(NRUN+1,2*K) = INTERDEX(K,1)
    MODINT(NRUN+1,2*K+1) = INTERDEX(K,2)
  ENDIF

  ELSE
    MODINT(NRUN+1,1) = NINTS + C2
    DO 2208 K = 1,C2
      INTERDEX(K,1) = MODINT(BASE,2*K)
      INTERDEX(K,2) = MODINT(BASE,2*K+1)
    ENDIF

    MODINT(NRUN+1,K+1) = INTERDEX(K,1)
    DO 2211 J = 1,C2+NINTS
      WRITE(*,8019)
    ENDIF

  8019 FORMAT('Enter pair of indices for one interaction: ',$)

  2111 READ(*,8019) INTERDEX(J,1),INTERDEX(J,2)

  MODINT(NRUN+1,2*C2+2*K) = INTERDEX(C2+K,1)
  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)
    INTERDEX(K,2) = MODINT(BASE,2*K+1)
  ENDIF

  MODINT(NRUN+1,K+1) = INTERDEX(K,1)
  DO 2219 J = 1,2*C2
    WRITE(*,8019)
  ENDIF

  ELSE
    IF (BASE EQ 0) THEN
      GOTO 2214
    ENDIF

    MODINT(NRUN+1,1) = C2
    DO 2220 K = 1,C2
      INTERDEX(K,1) = MODINT(BASE,2*K)
      INTERDEX(K,2) = MODINT(BASE,2*K+1)
    ENDIF

    MODINT(NRUN+1,K+1) = INTERDEX(K,1)
    NINTS = C2
    2214 ENDIF

C CREATE DATA MATRIX

IF (XX1(I,COL(J)+2),EQ,999.00) THEN
  FLAG(I) = 1
  NUMMISS = NUMMISS + 1
ENDIF

7083 CONTINUE

C CREATE INTERACTIONS

IF (X(I,COL(J)+2),EQ,0) THEN
  J=1
  DO 31 K = 1,NSUBS
    IF (FLAG(K),EQ,0) THEN
      DO 30 L = 1,NINTS
        XX1(K,NCOV+L,2) = XX(K,INTERDEX(L,1)+2)
        XX(K,INTERDEX(L,2)+2) = XX1(J,NCOV+L,2)
      ENDIF
    30 CONTINUE
  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,NSUBS
      107 COVRT(J,NCOV) = COVNAME(INTERDEX(J,1)+1:4)///**/**
      COVRT(J,NCOV) = COVRT(J,NCOV)///**/**
    ENDIF
  ENDIF

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

C ENTER START VALUES FOR PARAMETERS OF BASELINE MODEL INTO X

IF (BASE EQ 0) THEN
  DO 3123 I = 1, M - NCOV
    3123 X(I) = START(I)
ELSE
  DO 9191 I = 1, M - NCOV
    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
C IF (NRUN .EQ. 0) THEN
   DO 6969 I = M-NCOV+1, M
      X(I) = 0
   6969
C ELSE
C IF (BASE .GT. 0) THEN
C   POLD = MOVAR(BASE,1) + MOINT(BASE,1)
C   DO 3124 I = 1, POLD
C      X(I+2) = FARMCOV(BASE,1)
C   3124
C ELSE
C   POLD = 0
C ENDIF
C DO 74 I = POLD+1, NCOV
C    X(I+2) = 0
C 74
C ENDIF
C
C INITIALIZE THE VECTOR OF ESTIMATED S.D.'S AND THE ESTIMATED COVARIANCE MATRIX
C
C DO 34 I = 1, M
ST(I) = 0.000
DO 33 J = 1, M
   A(I,J) = 0
33 MAT(I,J) = 0.000
34 CONTINUE
C
C ------SOLVE PROBLEM---------------------
WRITE(*,1007)
CHI = 0.0
ICOV=NCOV
IFLG=0
IF(NCOV.EQ.0) IFLG=1
C
C CALCULATE THE SCORE TEST STATISTIC
DO 8090 I=M-NCOV+1,1,25
   XINC(I) = 0
8090
MM = M
CALL PARTIII(XO,A,MM,XINC,MAT,ICOV,OBJF)
CALL SCORE(A,ICOV,XINC,STAT)

M=M
IFAIL=-1
COVTEMP=NCOV
OBJF=0
DO 7099 I=1,25
   OBJGRD(I) = 0
7099
WRITE(*,*) '-----OPTIMIZING THE LIKELIHOOD FUNCTION-----'
WRITE(*,*) ' (PLEASE WAIT)'
CALL EDFU('MAJOR PRINT LEVEL = 0')
CALL EDFU('MAJOR ITERATION LIMIT = 1000')

6970 CALL EDFU(N,NCLIN,NCNLN,NROWA,NROWB,NROWS,AA,BL,BU,
   + EPSUM,OBJFUN,ITER,ISTATE,C,CJAC,CLAMS,A,OBJF,OBJGRD,R,
   + X,IWORK,LWORK,WORK,LWORK,USER,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 SATISFIES FIRST ORDER',1X,
   + 'KUHN-TUCKER CONDITIONS, BUT THE SEQUENCE OF ITERATES HAS NOT',
   + 'CONVERGED! POSSIBLE REASON: REQUIRED 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
6974 FORMAT(1X,'FINAL ITERATE DOES NOT SATISFY FIRST',1X,
   + 'ORDER KUHN-TUCKER CONDITIONS')
IF (IFAIL .EQ. 7) THEN
   WRITE(*,1007)
   WRITE(*,6976)
IF (NUN .GT. 25) THEN
WRITE(*,1007)
WRITE(*,1007)
GOTO 8083
ENDIF

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

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

C COMPUTE SAMPLE INFORMATION MATRIX
CALL PARTII(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,*)
C EVALUATE THE ESTIMATED S.D. OF THE M.L.Es OF THE PARAMS.
DO 81 I = 1, M
IF(MAT(I,1) .LE. 0.0) THEN
WRITE(*,*) 'INFORMATION MATRIX IS SINGULAR'
GO TO 8083
ENDIF
81 ST(I) = SQRT(MAT(I,1))
C WRITE TO THE OUTPUT FILE THE MLE AND S.D.s OF THE PARAMS OF THE MIXING DISTR
WRITE(25,8005) NSUBS
WRITE(25,*)
WRITE(25,8006)
WRITE(25,*)
WRITE(25,8007)
WRITE(25,*)
WRITE(25,1015) X(1), X(2), ST(1), ST(2)
WRITE(25,*)
WRITE(25,1015) X(1), X(2), ST(1), ST(2)
WRITE(*,*)

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, 3
PARMBASE(NRUN+1,K) = X(K)
DO 3116 K = 3, N
PARMCOV(NRUN+1,K-2) = X(K)
3115
3116
NRUN = NRUN + 1
**C** EVALUATE THE Z-VALUE FOR EACH COVARIATE (MLE/ESE).
**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**

**END IF**

**90 CONTINUE**

**10 IF(IFLAG.EQ.1) THEN**

**WRITE(25,*)**

**': SIGNIFICANT AT THE 5% LEVEL'**

**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,'(*)')**

**WRITE(25,'(7X,30I7)') (K, K=1,M)**

**DO 110 I = 1, M-NCOV**

**WRITE(25,'(5X,A5,30(I1,F6.3))') GMAT(I), + (MAT(I,J)/ST(I)*ST(J)),J=1,M**

**DO 12 I = M-NCOV+1, M**

**WRITE(25,'(A10,30(I1,F6.3))') COVRT(I-M+NCOV), + (MAT(I,J)/ST(I)*ST(J)),J=1,M**

**IF(IFLAG.EQ.1) THEN**

**GOTO 7078**

**END IF**

**110 CONTINUE**

**99 CONTINUE**

**GOTO 7078**

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

**1005 FORMAT(2X,'STARTING VALUES FOR NULL MODEL..(',I12,I12,'.,PARAMETERS?)')

**1006 FORMAT(2X,'STARTING VALUES FOR CURRENT MODEL..(',I16,I12,

+ 'PARAMETERS) ?')

**107 FORMAT(39H------------------------------------------------------)

**109 FORMAT('===>,5X,'GRADIENT VECTOR:')

**110 FORMAT('===>,7H ITER =',I4,6X,'-OBJF = ',F16.5,7X,'SH COND =

+ E25.5,7X,'===>','

+ 32H

+ X0(1),....,X0(I2),1H,10(8F15.8))

**111 FORMAT(25H SUCCESSFUL CONVERGENCE: 10X,-OBJF =',F12.4)

**112 FORMAT(2X,'NEW VALUES:(',I12,I12,'.,PAR.)?')

**114 FORMAT('COVARIATES ENTERING ANALYSIS: ',A412,30(30X,A412,7X)

+ '+

+ ')

**115 FORMAT(16X,'SHAPE =',F12.4,10X,'MEAN = ',F11.5,6X,'ST.DEV =

+ ',F11.5,7X,'10X,'ST.DEV = ',F11.5,6X,'1ST-GAP =

+ ',F9.4,5X,'2ST-DEV = ',F9.5,6X,'ST.DEV = ',F10.6

+ ',F9.5,7X,'3ST-DEV = ',F10.6,6X,'ST.DEV = ',F13.5)

**116 FORMAT(6X,'DEVIANCE FOR NULL MODEL (-2*LOG-LIK) = ',F13.5)

**117 FORMAT(6X,'DEVIANCE: CURRENT MODEL',3X,'-OBJF = ',F13.5,'NULL =

+ ',F13.5)

**118 FORMAT(6X,'L.R. RATIO TEST STATISTIC',2X,'=',F13.5,' WITH ',I3,

+ ' DEGREE(S) OF FREEDOM')

**119 FORMAT(6X,'SCORE TEST STATISTIC',7X,='F13.5,' WITH ',I3,

+ ' DEGREE(S) OF FREEDOM')

**120 FORMAT(6X,'-LOG-LIKLIHOOD =','F12.5')

**121 FORMAT(6X,'=',I12,7X,'WITH ',I3,' DEGREE(S) OF FREEDOM')

**122 FORMAT(6X,'= ',F12.5)

**123 FORMAT(6X,'= ',F12.5)

**124 FORMAT(5X,5B7(''))

**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 DERIVATIVE FOR THE NAG SUBROUTINE

**INTEGER**

**MODE, N, NSTATE, IUSER(1), USER(1)**

**DOUBLE PRECISION**

**X(N), OBF, OBGRD(N);**

**DOUBLE PRECISION**

**MAT(M,N),XINC(N),A(25,25)**

**INTEGER**

**K, COV**

**COMMON**

**XX(3000,30), XX1(3000,30), P(25,25),

+ MEL, NSUBG, START(5), COVTEMP**

**DO 669 K=1,25**

**669 XINC(K)=0**

**COV=COTEMP**

**CALL PARTIII(X,A,N,XINC,MAT,COV,OBFJ)**

**OBFJ=OBFJ**

**DO 671 K=1,N**

**671 OBFGRD(K) = XINC(K)**

**672 RETURN END**
SUBROUTINE OBJNULL(MODE,N,XO,OBJF,OBGRD,NSSTATE,IUSER,USER)

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

INTEGER MODE, N, NSSTATE, USER(1), IUSER(1)
DOUBLE PRECISION XO(25), OBJF, OBGRD(25)
INTEGER K, COV

COMMON XX(3000,10), X1(3000,30), P(25,25),
+ MNL, NGUSS, START(5), COVTEMP

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

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

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

OBJF=OBJF

DO 668 K=1,N
   OBGRD(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, XNORM
INTEGER MML, I, J, K, KP1, KB, KML, MI

IFVT(M) = 1
IF ( M .EQ. 1 ) GO TO 80
MML = M-1
A NORM < 0.00D
DO 10 J = 1 , M
   T = 0.00D
   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 , MML
   KP1 = K + 1
   M1 = K
   DO 25 I = KP1 , M
      IF ( DABS(A(I,K)) .GT. DABS(A(I,M1)) ) M1 = I
   25 CONTINUE
IFVT(K) = M1
IF ( M1 .NE. K ) IPVT(M) = -IPVT(M)
T = A(M1,K)
A(M1,K) = A(K,K)
A(K,K) = T
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. 0.00D ) 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 = 0.00D
   IF ( K .EQ. 1 ) GO TO 45
   KML = K - 1
   DO 40 I = 1 , KML
      T = T + A(I,K)*WORK(I)
   40 CONTINUE
   EK = 1.00D
   IF ( T .LT. 0.00D ) EK = -1.00D
   IF ( A(K,K) .EQ. 0.00D ) GO TO 90
   WORK(K) = -EK + T)/A(K,K)

50 CONTINUE
DO 60 KB = 1 , MML
   K = MB
   T = 0.00D
   KP1 = K + 1
   DO 55 I = KP1 , M
      T = T + A(I,K)*WORK(K)

55 CONTINUE
WORK(K) = T
M1 = IPVT(K)
IF ( M .EQ. K ) GO TO 60
T = WORK(M)
WORK(M) = 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
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)
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
XINC(1) = XINC(1)/A(1,1)
RETURN
END

SUBROUTINE SCORE(A,ICOV,XINC,STAT)
THIS SUBROUTINE EVALUATES THE SCORE TEST STATISTIC FOR TESTING
THE NULL HYPOTHESIS THAT THE REGRESSION COEFFICIENTS IN THE
REGRESSION MODEL ARE ALL ZEROS. THE FOLLOWING QUANTITIES MUST
BE PASSED AS INPUT:
1. SAMPLE INFORMATION MATRIX = A(25,25).
2. SCORE VECTOR AT THE MLE VALUES = -XINC(25,25).
3. NUMBER OF COVARIATES IN THE MODEL = ICOV.
4. THE NUMBER OF BASELINE PARAMETERS = MC.
ON OUTPUT, THE SUBROUTINE RETURNS THE VALUE OF THE SCORE TEST
STATISTIC *STAT*.

REAL*8 A(25,25),BTMB(25,25),MAT(25,25),BTM(25,25),GOFFP(25),
STAT,U(25),XINC(25)
INTEGER MC,ICOV
COMMON XX(3000,30),XX1(3000,30),P(25,25),
MDL,HSUBS,START($),COVTEMP

MC = 2

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

CALCULATION OF GOF TEST.

STEP 1: FIND THE COVARIANCE MATRIX FOR THE BASELINE MODEL

DO 2 J = 1, MC
DO 2 I = 1, MC
MAT(I,J) = A(I,J)
2 CONTINUE

CALL ARINV(MC,MC)

STEP 2: EVALUATE TRANS(IGB)*MAT*IGB

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

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,MC )

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, UNPAKED, NCHMBT)

INTEGER NC, NCHARS, NCHMBT
INTEGER UNPAKED(nchars), PACKED(nchars), NBYTE, NBLANK

EQUIVALENCE (NBYTE, NBLANK)
DATA NBLANK /1H/ 
DATA NBYTE /1H/ 

NCHMBT = 0
RETURN IF NCHARS IS LE ZERO

IF(NCHARS.LE.0) RETURN
SET NC=NUMBER OF CHARS TO BE DECODED
NC = 129
IF (NCHARS.LT.NC) NC=NCHARS
NWORDS = NC-NC+1+NC
J = 1
DO 110 I = 1, NWORDS, 4
UNPAKED(I) = PACKED(J)
UNPAKED(I+1) = NBYTE
UNPAKED(I+2) = NBLANK
UNPAKED(I+3) = NBLANK
110 J = J+1

CHECK UNPAKED ARRAY AND SET NCHMBT BASED ON TRAILING BLANKS FOUND
DO 200 N = 1, NWORDS, 4
NN = NWORDS - N - 2
NBYTE = UNPAKED(NN)
IF(NBYTE .NE. NBLANK) GO TO 210
200 CONTINUE
NN = 0
210 NCHMBT = (NN + 3) / 4
RETURN
END

SUBROUTINE PARTII(XO,A,M,XINC,MAT,COV,OBJF)

******************************************************************************
SCORE VECTOR AND SAMPLE INFORMATION MATRIX CALCULATIONS FOR:
PARETO MODEL. MODEL-I AND MODEL-II
******************************************************************************

THIS SUBROUTINE EVALUATES THE FIRST AND SECOND PARTIAL
DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION FOR THE PARETO
MODEL. MODEL-I AND MODEL-II

THE VARIABLES IN THIS SUBROUTINE ARE DEFINED AS FOLLOWS:
XO(25) = VECTOR OF THE UPDATED PARAMETER ESTIMATES WITH:
XO(1) = MLE OF SHAPE PAR. OF THE GAMMA MIXING DISTRIBUTION.
XO(2) = MLE OF MEAN OF THE GAMMA MIXING DISTRIBUTION.
XO(J) = MLE OF THE .1030\n\n022 REGRESSION COEFFICIENT IN THE
REGRESSION MODEL.
XINC(25) = NEGATIVE OF THE SCORE VECTOR
DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION.
OBJF = THE OBJECTIVE FUNCTION TO BE MAXIMIZED
= LOG LIKELIHOOD FUNCTION.
MDL = A SCALAR INDICATOR OF MODEL TO BE USED IN ANALYSIS WITH
VALUES: 1=PARETO MODEL 2=MODEL-I 3=MODEL-II.
START = VECTOR OF STARTING VALUES FOR THE BASELINE PARAMETERS.

GI = T(1,2) + T(1,3) - 1 = NUMBER OF TUMOR VISITS FOR THE
1ST PATIENT.
XB1 = Z1*BETA
XD = EXP(XB1)
SM1 = T(1,4) + T(1,5) = TOTAL FOLLOW UP TIME.
DEN = XO(2) + XB*SM1 + XO(1)
GB = GI + XO(1)

REAL*8 SM1,SM2,SM3,DEN,GI,OBJF,XB,XB1,MAT(25,25),
+ SM1,A(25,25),XD,XINC(25),GB
INTEGER COV,2,M
COMMON XX(3000,30),XX(3000,30),P(25,25),
+ MDL, NSUBS, START(5), COVTEMP

CHECK IF THE UPDATED ESTIMATE OF THE SHAPE OR SCALE PARAMETERS IS
ZERO, IF SO THEN REPLACE THE UPDATED VALUE BY THE STARTING VALUE.
IF(XO(1).LE.0.0) XO(1) = START(1)
IF(XO(2).LE.0.0) XO(2) = START(2)

INITIALIZE THE SCORE VECTOR AND THE MATRIX OF SECOND PARTIAL
DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION.
OBJF = 0.000
DO 4 I = 1, 25
XINC(I) = 0.000
DO 4 J = 1, 25
P(I,J) = 0.000
4 CONTINUE

INITIALIZE THE DIFFERENT SCALARS NEEDED IN CALCULATIONS.
SM1=0.000
SM2=0.000
SM3=0.000
DEN=0.000
GI = 0.000
GB = 0.000
SM1 = 0.000
SM2 = 0.000
XB = 1.000
XBI=0.000

SET UP THE VALUES OF GI AND SM1. THE VALUES OF THESE
QUANTITIES DEPEND ON THE MODEL CHOSEN.

GI = XXI(1,1)
SM1 = XXI(1,2)

IF THE NUMBER OF FAILURES FOR THE ith PATIENT IS > 0 THEN
COMPUTE SM1, SM2 AND SM3.

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
SUBROUTINE ARINV ( M, MAT )

INTEGER IJOB, IA, IER
REAL *8 MAT(M, M), D1, D2, WKAREA(30), B(1)

N = M
IA = 25
IJOB = 1
D1 = -1.0

CALL LINV3P( MAT, B, IJOB, N, IA, D1, D2, WKAREA, IER )

RETURN
END

SUBROUTINE LINV3P ( A, B, IJOB, N, IA, D1, D2, WKAREA, IER )

DOUBLE PRECISION A(IA, IA, N(1), WKAREA(1), C1, C2, D1, D2, WA, ZERO, ONE, SUM, C
DATA ZERO, 0.0D0, ONE, 1.0D0/

FIRST EXECUTABLE STATEMENT
LU DECOMPOSITION OF A

CALL LUDEATN ( A, IA, N, IA, IA, IA, 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. 120 ) GO TO 60
IF ( IJOB .LE. 0 .OR. IJOB .GT. 4 ) GO TO 55
SOLVE AX = B
IF ( IJOB .EQ. 2 .OR. IJOB .EQ. 3 ) CALL LWEQN3N ( A, IA, N, B, WKAREA, B)
IF ( IJOB .NE. 1 .AND. IJOB .NE. 3 ) GO TO 9005

MATRIX INVERSION

A(N, N) = ONE / A(N, N)
NMI = N - 1
IF ( NMI .LT. 1 ) GO TO 9005
DO 40 I = 1, NMI
L = N - I
M = L + 1
DO 15 J = 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(I, 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
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 = L, N
C = A(J, K)
C = A(K, JP)
50 CONTINUE
A(K,J) = A(K,J)
A(K,J) = C
45 CONTINUE
50 CONTINUE
GO TO 9005
55 CONTINUE

C WARNING WITH FIX - IJOB WAS SET INCORRECTLY
IJOB = 65
GO TO 9000

C TERMINAL ERROR - MATRIX A IS ALGORITHMICALLY SINGULAR
IJOB = 130
9000 CONTINUE
CALL UERROR(IJOB,64L1N1F)
9005 RETURN

SUBROUTINE LUATNM(A,IA,N,LU,ILU,LDLT,D1,D2,APVT,EQUIL,WAI,IER)

DIMENSION A(IA,1),LU(1,LU,1),APVT(1),EQUIL(1)
DOUBLE PRECISION A,LU,DL,D2,EQUIW,AO,ONE,P,F,SM,LDLT,F
DATA ONE,P,F,SM,LDLT,F

C FIRST EXECUTABLE STATEMENT
INITIALIZATION

C 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. BIGA) BIG = P
5 CONTINUE
IF (BIG .GT. BIGA) BIGA = BIG
IF (BIG .EQ. ZERO) GO TO 110
EQUIL(I) = ONE/BIGA
10 CONTINUE
COMPUTE U(I,J), I=1,...,J-1

C DO 35 I=1,1,J1
SUM = LU(I,J)
J1 = I-1
IF (IELD .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
15 CONTINUE
LU(I,J) = SUM

C WITH OUT ACCURACY TEST

20 WI = WI+DABS(SUM)
IF (AI .EQ. ZERO) AI = BIGA
TEST = WI/AI
IF (TEST .GT. WREL) WREL = TEST
GO TO 35

C

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
COMPUTE U(J,J) AND L(I,J), I=J+1,...
DO 70 I=J,N
SUM = LU(I,J)
IF (ILD .EQ. 0) GO TO 55
WITH ACCURACY TEST
45 AI = DABS(SUM)
WI = ZERO
IF (IM1 .LT. 1) GO TO 50
DO 45 K=1,IM1
T = LU(I,K)*LU(K,J)
SUM = SUM+T
WI = WI+DABS(T)
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 (IM1 .LT. 1) GO TO 65
DO 60 K=1,IM1
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
C 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*6TH
D2 = D2*4TH
GO TO 95
90 IF (DABS(D1) .GE. 6TH) GO TO 95
D1 = D1*6TH
D2 = D2*4TH
GO TO 90
95 CONTINUE
J1 = J+1
IF (J1 .LT. N) GO TO 105
C DIVIDE BY Pivot Element U(J,J)
P = LU(J,J)
DO 100 I=J1,N
LU(I,J) = LU(I,J)/P
100 CONTINUE
105 CONTINUE
C
PERFORM ACCURACY TEST
IF (IDT .EQ. 0) GO TO 9005
P = 3*N+3
WA = P*WREL
IF (WA+10.0D0*(~IDT) .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
9005 RETURN
END
C
SUBROUTINE LUDEMN (A,A,1,N,MAPVT)
C
C DIMENSION A(A,1),B(1),MAPVT(1),X(1)
C DOUBLE PRECISION A,X,SUM,APVT
C FIRST EXECUTABLE STATEMENT
SOLVE LY = B FOR Y
C
5 DO 1 I=1,N
5 X(I) = B(I)
10 IM = 0
DO 20 I=1,N
IF = APVT(I)
SUM = X(IF)
X(IF) = X(I)
IF (IM .EQ. 0) GO TO 15
IM = I-1
DO 10 J=IM,1
SUM = SUM-A(I,J)*X(J)
10 CONTINUE
GO TO 30
15 IF (SUM .NE. 0.0D0) IM = I
20 X(I) = SUM
DO 30 J=1,N
I = I+1
IPI = I+1
SUM = X(I)
IF (IPI .GT. N) GO TO 30
DO 25 J=IPI,N
SUM = SUM-A(I,J)*X(J)
25 CONTINUE
30 X(I) = SUM/A(I,1)
RETURN
END
C
SUBROUTINE UBERTST (IER,NAMBP)
C
C SPECIFICATIONS FOR ARGUMENTS
INTEGER IER
INTEGER NAME(1)
C
SPECIFICATIONS FOR LOCAL VARIABLES
INTEGER 1,I8Q,IEQDF,IOUNIT,LEVEL,LEVOLD,NAMBP(6),
NAMES(6),NAMURP(6),NIN,NMTB
DATA NAMSET/I0H,1HE,1HR,1HS,1HE,1HT/
DATA NAMES/6*1H /
DATA LEVEL/4,1EQDF/0,1EQH/=
C
CALL USKPD (NAME,6,NAMURP,NMTB)
C
CALL UGETIO(1,NIN,IOUNIT)
C
CHECK IER
C
IF (IER .GT. 999) GO TO 25
IF (IER .LT. 32) GO TO 25
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,NAMBP,IEQ,NAMURP
IF (IEQDF .EQ. 0) WRITE(IOUNIT,35) IER,NAMURP
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,NAMBP,IEQ,NAMURP
IF (IEQDF .EQ. 0) WRITE(IOUNIT,40) IER,NAMURP
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,NAMBP,IEQ,NAMURP
IF (IEQDF .EQ. 0) WRITE(IOUNIT,45) IER,NAMURP
GO TO 30
15 CONTINUE
C
CHECK FOR UERSET CALL
DO 20 I=1,6
IF (NAMURP(I) .NE. NAMSET(I)) GO TO 25
20 CONTINUE
C
LEVEL = LEVEL
IF (LEVEL .LT. 4) LEVEL = 4
IF (LEVEL .LT. 4) LEVEL = 4
GO TO 30
25 CONTINUE
C
PRINT NON-DEFINED MESSAGE
IF (IEQDF .EQ. 1) WRITE(IOUNIT,50) IER,NAMBP,IEQ,NAMURP
IF (IEQDF .EQ. 0) WRITE(IOUNIT,50) IER,NAMURP
30 IEQDF = 0
RETURN
C
FORMAT(19H *** TERMINAL ERROR,10X,7H(IER = ,13, 1 20H FROM IMS; ROUTINE ,6AL,6A1)
40 FORMAT(27H *** WARNING WITH FIX ERROR,2X,7H(IER = ,13, 1 20H FROM IMS; ROUTINE ,6AL,6A1)
45 FORMAT(18H *** WARNING ERROR,11X,7H(IER = ,13, 1 20H FROM IMS; ROUTINE ,6AL,6A1)
50 FORMAT(10H *** UNDEFINED ERROR,5X,7H(IER = ,15, 1 20H FROM IMS; ROUTINE ,6AL,6A1)
C
SAVE P FOR R CASE
C
P IS THE PAGE NAMURP
R IS THE ROUTINE NAMURP
55 IEQDF = 1
DO 60 I=1,6
IF (I .EQ. 3) GO TO 10
IF (I .EQ. 2) GO TO 5
IF (IEQDF .EQ. 0) GO TO 9005
NIN = NIND
C
SUBROUTINE UGETIO(IOPT,NIN,LOCAL)
C
C SPECIFICATIONS FOR ARGUMENTS
INTEGER IOPT,NIN,LOCAL
C
C SPECIFICATIONS FOR LOCAL VARIABLES
INTEGER MIND,OUTD
DATA MIND/5,OUTD/6/
C
FIRST EXECUTABLE STATEMENT
IF (IOPT .EQ. 3) GO TO 10
IF (IOPT .EQ. 2) GO TO 5
IF (IOPT .EQ. 1) GO TO 9005