NAG Toolbox for MATLAB

g08rb

1 Purpose

g08rb calculates the parameter estimates, score statistics and their variance-covariance matrices for the linear model using a likelihood based on the ranks of the observations when some of the observations may be right-censored.

2 Syntax

```
[prvr, irank, zin, eta, vapvec, parest, ifail] = g08rb(nv, y, x, icen,
gamma, nmax, tol, 'ns', ns, 'ip', ip)
```

3 Description

Analysis of data can be made by replacing observations by their ranks. The analysis produces inference for the regression model where the location parameters of the observations, θ_i, $i = 1, 2, \ldots, n$, are related by $\theta = X\beta$. Here X is an n by p matrix of explanatory variables and β is a vector of p unknown regression parameters. The observations are replaced by their ranks and an approximation, based on Taylor's series expansion, made to the rank marginal likelihood. For details of the approximation see Pettitt 1982.

An observation is said to be right-censored if we can only observe Y_j^* with $Y_j^* \leq Y_j$. We rank censored and uncensored observations as follows. Suppose we can observe Y_j, for $j = 1, 2, \ldots, n$, directly but Y_j^*, for $j = n+1, n+2, \ldots, q$; $n \leq q$, are censored on the right. We define the rank r_j of Y_j, for $j = 1, 2, \ldots, n$, in the usual way; r_j equals i if and only if Y_j is the ith smallest amongst the $Y_1, Y_2, \ldots, Y_n$. The right-censored Y_j^*, for $j = n+1, n+2, \ldots, q$, has rank r_j if and only if Y_j^* lies in the interval $\left[Y_{(r_j)}, Y_{(r_j+1)}\right]$, with $Y_0 = -\infty$, $Y_{(n+1)} = +\infty$ and $Y_{(1)} < \cdots < Y_{(n)}$ the ordered Y_j, for $j = 1, 2, \ldots, n$.

The distribution of the Y is assumed to be of the following form. Let $F_L(y) = e^y/(1+e^y)$, the logistic distribution function, and consider the distribution function $F_\gamma(y)$ defined by $1 - F_\gamma = [1 - F_L(y)]^{1/\gamma}$. This distribution function can be thought of as either the distribution function of the minimum, $X_{1,\gamma}$, of a random sample of size γ^{-1} from the logistic distribution, or as the $F_\gamma(y - \log\gamma)$ being the distribution function of a random variable having the F-distribution with 2 and $2\gamma^{-1}$ degrees of freedom. This family of generalized logistic distribution functions $\left[F_\gamma(.);0 \leq \gamma < \infty\right]$ naturally links the symmetric logistic distribution ($\gamma = 1$) with the skew extreme value distribution ($\lim\gamma \to 0$) and with the limiting negative exponential distribution ($\lim\gamma \to \infty$). For this family explicit results are available for right-censored data. See Pettitt 1983 for details.

Let l_R denote the logarithm of the rank marginal likelihood of the observations and define the $q \times 1$ vector a by $a = l_R'(\theta = 0)$, and let the q by q diagonal matrix B and q by q symmetric matrix A be given by $B - A = -l_R''(\theta = 0)$. Then various statistics can be found from the analysis.

(a) The score statistic $X^{\mathrm{T}}a$. This statistic is used to test the hypothesis $H_0 : \beta = 0$ (see (e)).

(b) The estimated variance-covariance matrix of the score statistic in (a).

(c) The estimate $\hat{\beta}_R = MX^{\mathrm{T}}a$.

(d) The estimated variance-covariance matrix $M = \left(X^{\mathrm{T}}(B-A)X\right)^{-1}$ of the estimate $\hat{\beta}_R$.

(e) The χ^2 statistic $Q = \hat{\beta}_R M^{-1}\hat{\beta}_r = a^{\mathrm{T}}X\left(X^{\mathrm{T}}(B-A)X\right)^{-1}X^{\mathrm{T}}a$, used to test $H_0 : \beta = 0$. Under H_0, Q has an approximate χ^2-distribution with p degrees of freedom.

(f) The standard errors $M_{ii}^{1/2}$ of the estimates given in (c).

(g) Approximate z-statistics, i.e., $Z_i = \hat{\beta}_{R_i}/se\left(\hat{\beta}_{R_i}\right)$ for testing $H_0 : \beta_i = 0$. For $i = 1, 2, \ldots, n$, Z_i has an approximate $N(0, 1)$ distribution.

In many situations, more than one sample of observations will be available. In this case we assume the model,

$$h_k(Y_k) = X_k^{\mathrm{T}}\beta + e_k, \qquad k = 1, 2, \ldots, \mathbf{ns},$$

where **ns** is the number of samples. In an obvious manner, Y_k and X_k are the vector of observations and the design matrix for the kth sample respectively. Note that the arbitrary transformation h_k can be assumed different for each sample since observations are ranked within the sample.

The earlier analysis can be extended to give a combined estimate of β as $\hat{\beta} = Dd$, where

$$D^{-1} = \sum_{k=1}^{\mathbf{ns}} X^{\mathrm{T}}(B_k - A_k)X_k$$

and

$$d = \sum_{k=1}^{\mathbf{ns}} X_k^{\mathrm{T}} a_k,$$

with a_k, B_k and A_k defined as a, B and A above but for the kth sample.

The remaining statistics are calculated as for the one sample case.

4 References

Kalbfleisch J D and Prentice R L 1980 *The Statistical Analysis of Failure Time Data* Wiley

Pettitt A N 1982 Inference for the linear model using a likelihood based on ranks *J. Roy. Statist. Soc. Ser. B* **44** 234–243

Pettitt A N 1983 Approximate methods using ranks for regression with censored data *Biometrika* **70** 121–132

5 Parameters

5.1 Compulsory Input Parameters

1: **nv(ns) – int32 array**

The number of observations in the ith sample, for $i = 1, 2, \ldots, \mathbf{ns}$.

Constraint: $\mathbf{nv}(i) \geq 1$, for $i = 1, 2, \ldots, \mathbf{ns}$.

2: **y(nsum) – double array**

The observations in each sample. Specifically, $\mathbf{y}\left(\sum_{k=1}^{i-1} \mathbf{nv}(k) + j\right)$ must contain the jth observation in the ith sample.

3: **x(ldx,ip) – double array**

ldx, the first dimension of the array, must be at least **nsum**.

The design matrices for each sample. Specifically, $\mathbf{x}\left(\sum_{k=1}^{i-1} \mathbf{nv}(k) + j, l\right)$ must contain the value of the lth explanatory variable for the jth observations in the ith sample.

Constraint: **x** must not contain a column with all elements equal.

4: **icen(nsum) – int32 array**

Defines the censoring variable for the observations in **y**.

icen$(i) = 0$

If **y**(i) is uncensored.

icen$(i) = 1$

If **y**(i) is censored.

Constraint: **icen**$(i) = 0$ or 1, for $i = 1, 2, \ldots,$ **nsum**.

5: **gamma – double scalar**

The value of the parameter defining the generalized logistic distribution. For **gamma** ≤ 0.0001, the limiting extreme value distribution is assumed.

Constraint: **gamma** > 0.0.

6: **nmax – int32 scalar**

the value of the largest sample size.

Constraint: **nmax** $= \max_{1 \leq i \leq \mathbf{ns}} (\mathbf{nv}(i))$ and **nmax** $>$ **ip**.

7: **tol – double scalar**

The tolerance for judging whether two observations are tied. Thus, observations Y_i and Y_j are adjudged to be tied if $\left|Y_i - Y_j\right| <$ **tol**.

Constraint: **tol** > 0.0.

5.2 Optional Input Parameters

1: **ns – int32 scalar**

Default: The dimension of the array **nv**.

the number of samples.

Constraint: **ns** ≥ 1.

2: **ip – int32 scalar**

Default: The dimension of the arrays **x**, **prvr**. (An error is raised if these dimensions are not equal.)

the number of parameters to be fitted.

Constraint: **ip** ≥ 1.

5.3 Input Parameters Omitted from the MATLAB Interface

nsum, ldx, ldprvr, work, lwork, iwa

5.4 Output Parameters

1: **prvr(ldprvr,ip) – double array**

The variance-covariance matrices of the score statistics and the parameter estimates, the former being stored in the upper triangle and the latter in the lower triangle. Thus for $1 \leq i \leq j \leq$ **ip**, **prvr**(i,j) contains an estimate of the covariance between the ith and jth score statistics. For $1 \leq j \leq i \leq$ **ip** $- 1$, **prvr**$(i+1,j)$ contains an estimate of the covariance between the ith and jth parameter estimates.

2: **irank(nmax) – int32 array**

For the one sample case, **irank** contains the ranks of the observations.

3: **zin(nmax) – double array**

For the one sample case, **zin** contains the expected values of the function g(.) of the order statistics.

4: **eta(nmax) – double array**

For the one sample case, **eta** contains the expected values of the function $g\prime(.)$ of the order statistics.

5: **vapvec(nmax $\times$ (nmax + 1)/2) – double array**

For the one sample case, **vapvec** contains the upper triangle of the variance-covariance matrix of the function g(.) of the order statistics stored column-wise.

6: **parest(4 $\times$ ip + 1) – double array**

The statistics calculated by the function as follows. The first **ip** components of **parest** contain the score statistics. The next **ip** elements contain the parameter estimates. **parest**$(2 \times$ **ip** $+ 1)$ contains the value of the χ^2 statistic. The next **ip** elements of **parest** contain the standard errors of the parameter estimates. Finally, the remaining **ip** elements of **parest** contain the z-statistics.

7: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **ns** < 1,
or **tol** ≤ 0.0,
or **nmax** $\leq$ **ip**,
or **ldprvr** $<$ **ip** $+ 1$,
or **ldx** $<$ **nsum**,
or **nmax** $\neq \max_{1 \leq i \leq \mathbf{ns}}(\mathbf{nv}(i))$,
or **nv**$(i) \leq 0$ for some i, $i = 1, 2, \ldots,$ **ns**,
or **nsum** $\neq \sum_{i=1}^{\mathbf{ns}} \mathbf{nv}(i)$,
or **ip** < 1,
or **gamma** < 0.0,
or **lwork** $<$ **nmax** $\times$ (**ip** $+ 1$).

ifail $= 2$

On entry, **icen**$(i) \neq 0$ or ,1 for some $1 \leq i \leq$ **nsum**.

ifail $= 3$

On entry, all the observations are adjudged to be tied. You are advised to check the value supplied for **tol**.

ifail $= 4$

The matrix $X^{\mathrm{T}}(B - A)X$ is either ill-conditioned or not positive-definite. This error should only occur with extreme rankings of the data.

ifail $= 5$

On entry, at least one column of the matrix X has all its elements equal.

7 Accuracy

The computations are believed to be stable.

8 Further Comments

The time taken by g08rb depends on the number of samples, the total number of observations and the number of parameters fitted.

In extreme cases the parameter estimates for certain models can be infinite, although this is unlikely to occur in practice. See Pettitt 1982 for further details.

9 Example

```
nv = [int32(40)];
y = [143;
     164;
     188;
     188;
     190;
     192;
     206;
     209;
     213;
     216;
     220;
     227;
     230;
     234;
     246;
     265;
     304;
     216;
     244;
     142;
     156;
     163;
     198;
     205;
     232;
     232;
     233;
     233;
     233;
     233;
     239;
     240;
     261;
     280;
     280;
     296;
     296;
     323;
     204;
     344];
x = [0;
     0;
     0;
     0;
     0;
     0;
```

```
      0;
      0;
      0;
      0;
      0;
      0;
      0;
      0;
      0;
      0;
      0;
      0;
      0;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1];
icen = [int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(1);
      int32(1);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
```

```
     int32(0);
     int32(0);
     int32(1);
     int32(1)];
gamma = 1e-05;
nmax = int32(40);
tol = 1e-05;
[parvar, irank, zin, eta, vapvec, parest, ifail] = ...
    g08rb(nv, y, x, icen, gamma, nmax, tol)
```

```
parvar =
    7.6526
    0.1307
irank =
           2
           5
           6
           7
           8
           9
          12
          13
          14
          15
          16
          17
          18
          25
          28
          30
          35
          15
          27
           1
           3
           4
          10
          11
          19
          20
          21
          22
          23
          24
          26
          27
          29
          31
          32
          33
          34
          36
          10
          36
zin =
   -0.9494
   -0.8682
   -0.8250
   -0.8250
   -0.7800
   -0.7487
   -0.6462
   -0.6092
   -0.5707
   -0.5307
   -0.4873
   -0.4418
   -0.3942
    0.0234
    0.2837
```

```
     0.5198
     1.6126
     0.4693
     1.1837
    -0.9750
    -0.9230
    -0.8960
    -0.7164
    -0.6820
    -0.3179
    -0.3179
    -0.1440
    -0.1440
    -0.1440
    -0.1440
     0.1003
     0.1837
     0.3948
     0.7460
     0.7460
     1.1543
     1.1543
     2.1126
     0.2836
     3.1126
eta =
     0.0506
     0.1318
     0.1750
     0.1750
     0.2200
     0.2513
     0.3538
     0.3908
     0.4293
     0.4693
     0.5127
     0.5582
     0.6058
     1.0234
     1.2837
     1.5198
     2.6126
     0.4693
     1.1837
     0.0250
     0.0770
     0.1040
     0.2836
     0.3180
     0.6821
     0.6821
     0.8560
     0.8560
     0.8560
     0.8560
     1.1003
     1.1837
     1.3948
     1.7460
     1.7460
     2.1543
     2.1543
     3.1126
     0.2836
     3.1126
vapvec =
      array elided
parest =
     4.5840
```

```
    0.5990
    2.7459
    0.3615
    1.6571
ifail =
           0
```