NAG Toolbox for MATLAB

g02dg

1 Purpose

g02dg calculates the estimates of the parameters of a general linear regression model for a new dependent variable after a call to g02da.

2 Syntax

```
[rss, cov, q, b, se, res, ifail] = g02dg(weight, wt, rss, ip, irank,
cov, q, svd, p, y, wk, 'n', n)
```

3 Description

g02dg uses the results given by g02da to fit the same set of independent variables to a new dependent variable.

g02da computes a QR decomposition of the matrix of p independent variables and also, if the model is not of full rank, a singular value decomposition (**svd**). These results can be used to compute estimates of the parameters for a general linear model with a new dependent variable. The QR decomposition leads to the formation of an upper triangular p by p matrix R and an n by n orthogonal matrix Q. In addition the vector $c = Q^{\mathrm{T}}y$ (or $Q^{\mathrm{T}}W^{1/2}y$) is computed. For a new dependent variable, y_{new}, g02dg computes a new value of $c = Q^{\mathrm{T}}y_{\mathrm{new}}$ or $Q^{\mathrm{T}}W^{1/2}y_{\mathrm{new}}$.

If R is of full rank, then the least-squares parameter estimates, $\hat{\beta}$, are the solution to

$$R\hat{\beta} = c_1,$$

where c_1 is the first p elements of c.

If R is not of full rank, then g02da will have computed an **svd** of R,

$$R = Q_* \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} P^{\mathrm{T}},$$

where D is a k by k diagonal matrix with nonzero diagonal elements, k being the rank of R, and Q_* and P are p by p orthogonal matrices. This gives the solution

$$\hat{\beta} = P_1 D^{-1} Q_{*_1}^{\mathrm{T}} c_1,$$

P_1 being the first k columns of P, i.e., $P = (P_1 P_0)$, and Q_{*_1} being the first k columns of Q_*. Details of the **svd** are made available by g02da in the form of the matrix P^*:

$$P^* = \begin{pmatrix} D^{-1}P_1^{\mathrm{T}} \\ P_0^{\mathrm{T}} \end{pmatrix}.$$

The matrix Q_* is made available through the workspace of g02da.

In addition to parameter estimates, the new residuals are computed and the variance-covariance matrix of the parameter estimates are found by scaling the variance-covariance matrix for the original regression.

4 References

Golub G H and Van Loan C F 1996 *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

Hammarling S 1985 The singular value decomposition in multivariate statistics *SIGNUM Newsl.* **20 (3)** 2–25

Searle S R 1971 *Linear Models* Wiley

5 Parameters

5.1 Compulsory Input Parameters

1: **weight – string**

Indicates if weights are to be used.

weight = 'U' (Unweighted)

Least-squares estimation is used.

weight = 'W' (Weighted)

Weighted least-squares is used and weights must be supplied in array **wt**.

Constraint: **weight** = 'U' or 'W'.

2: **wt**$(*)$ **– double array**

Note: the dimension of the array **wt** must be at least **n** if **weight** = 'W', and at least 1 otherwise.

If **weight** = 'W', **wt** must contain the weights to be used in the weighted regression.

If **wt**$(i) = 0.0$, the ith observation is not included in the model, in which case the effective number of observations is the number of observations with nonzero weights.

If **weight** = 'U', **wt** is not referenced and the effective number of observations is n.

Constraint: **wt**$(i) \geq 0.0$ if **weight** = 'W', for $i = 1, 2, \ldots, n$.

3: **rss – double scalar**

The residual sum of squares for the original dependent variable.

Constraint: **rss** > 0.0.

4: **ip – int32 scalar**

p, the number of independent variables (including the mean if fitted).

Constraint: $1 \leq$ **ip** $\leq$ **n**.

5: **irank – int32 scalar**

The rank of the independent variables, as given by g02da.

Constraint: **irank** > 0, and if **svd** = **false**, then **irank** = **ip**, else **irank** $\leq$ **ip**.

6: **cov**(**ip** $\times$ (**ip** + **1**)/**2**) **– double array**

The covariance matrix of the parameter estimates as given by g02da.

7: **q**(**ldq**,**ip** + **1**) **– double array**

ldq, the first dimension of the array, must be at least **n**.

The results of the QR decomposition as returned by g02da.

8: **svd – logical scalar**

Indicates if a singular value decomposition was used by g02da.

svd = **true**

A singular value decomposition was used by g02da.

svd = **false**

A singular value decomposition was not used by g02da.

9: **p**(∗) **– double array**

Note: the dimension of the array **p** must be at least **ip** if **svd** = **false**, and at least **ip** × **ip** + 2 × **ip** otherwise.

Details of the *QR* decomposition and **svd**, if used, as returned in array **p** by g02da.

If **svd** = **false**, only the first **ip** elements of **p** are used; these contain the zeta values for the *QR* decomposition (see f08ae for details).

If **svd** = **true**, the first **ip** elements of **p** contain the zeta values for the *QR* decomposition (see f08ae for details) and the next **ip** × **ip** + **ip** elements of **p** contain details of the singular value decomposition.

10: **y**(**n**) **– double array**

The new dependent variable, y_{new}.

11: **wk**(**5** × (**ip** − **1**) + **ip** × **ip**) **– double array**

If **svd** = **true**, **wk** must be unaltered from the previous call to g02da or g02dg.

If **svd** = **false**, **wk** is used as workspace.

5.2 Optional Input Parameters

1: **n – int32 scalar**

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

n, the number of observations.

Constraint: **n** ≥ **ip**.

5.3 Input Parameters Omitted from the MATLAB Interface

ldq

5.4 Output Parameters

1: **rss – double scalar**

The residual sum of squares for the new dependent variable.

2: **cov**(**ip** × (**ip** + **1**)/**2**) **– double array**

The upper triangular part of the variance-covariance matrix of the **ip** parameter estimates given in **b**. They are stored packed by column, i.e., the covariance between the parameter estimate given in **b**(i) and the parameter estimate given in **b**(j), $j \geq i$, is stored in **cov**($j \times (j-1)/2 + i$).

3: **q**(**ldq**,**ip** + **1**) **– double array**

The first column of **q** contains the new values of c, the remainder of **q** will be unchanged.

4: **b**(**ip**) **– double array**

The least-squares estimates of the parameters of the regression model, $\hat{\beta}$.

5: **se(ip) – double array**

The standard error of the estimates of the parameters.

6: **res(n) – double array**

The residuals for the new regression model.

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, **ip** < 1,
or **n** $<$ **ip**,
or **irank** ≤ 0,
or **svd** $=$ **false** and **irank** $\neq$ **ip**,
or **svd** $=$ **true** and **irank** $>$ **ip**,
or **ldq** $<$ **n**,
or **rss** ≤ 0.0,
or **weight** $\neq$ 'U' or 'W'.

ifail $= 2$

On entry, **weight** $=$ 'W' and a value of **wt** < 0.0.

7 Accuracy

The same accuracy as g02da is obtained.

8 Further Comments

The values of the leverages, h_i, are unaltered by a change in the dependent variable so a call to g02fa can be made using the value of **h** from g02da.

9 Example

```
weight = 'U';
wt = [0];
rss = 22.22679999999987;
ip = int32(5);
irank = int32(4);
cov = [0.1481786666666657;
     0.03704466666666638;
     0.7038486666666622;
     0.03704466666666652;
     -0.2222679999999989;
     0.7038486666666626;
     0.03704466666666648;
     -0.2222679999999987;
     -0.2222679999999985;
     0.7038486666666621;
     0.03704466666666641;
     -0.2222679999999984;
     -0.2222679999999987;
     -0.2222679999999985;
```

```
     0.7038486666666622];
q   =   [-132.3142479415325,   -3.464101615137754,   -0.866025403784439,   -
0.8660254037844387, ...
     -0.8660254037844387, -0.8660254037844387;
      -4.384999999999992, 0.2542949013547401, 1.5, -0.5000000000000001, -
0.5000000000000001, -0.5000000000000001;
          3.450681092190354,  0.2542949013547401,  0.2464408878569322,  -
1.414213562373095, ...
     0.7071067811865476, 0.7071067811865477;
         -4.564215887385989,  0.2542949013547401,  0.2464408878569322,  -
0.1493782079840455, -1.224744871391589, 1.224744871391589;
         -0.5165745490611717,  0.2542949013547401,  0.2464408878569322,  -
0.1493782079840455, ...
     -0.2512250416211346, 3.684598669906146e-16;
           -1.786105166371516,   0.2542949013547401,   0.2464408878569322,
0.4236386677182777, ...
     0.04757984926531376, 0.05853513115087475;
         -2.040746763558067,  0.2542949013547401,  0.2464408878569322,  -
0.1493782079840455, ...
     -0.2512250416211346, 0.4717748148066523;
         1.699599417218472,  0.2542949013547401,  -0.3431208422711533,  -
0.05795805473930137, ...
     -0.09747415577321789, 0.2226984422646312;
          1.338178326423549,  0.2542949013547401,  0.2464408878569322,  -
0.1493782079840455, ...
     0.4136728082784363, 0.1941369084874573;
         2.919599417218471,  0.2542949013547401,  -0.3431208422711533,  -
0.05795805473930137, ...
     -0.09747415577321789, 0.2226984422646312;
         -1.151821673576453,  0.2542949013547401,  0.2464408878569322,  -
0.1493782079840455, ...
     0.4136728082784363, 0.1941369084874573;
           0.2738948336284861,   0.2542949013547401,   0.2464408878569322,
0.4236386677182777, ...
     0.04757984926531376, 0.05853513115087475];
svd = true;
p = [1.135198279858991;
     1.13078348304906;
     1.234006904805161;
     1.228003039942231;
     1.263332232672042;
     3.872983346207417;
     1.732050807568878;
     1.732050807568878;
     1.732050807568877;
     1.64780261914309e-16;
     0.2309401076758503;
     8.35089227579159e-18;
     -1.238223776129407e-17;
     7.900875770685859e-17;
     -0.447213595499958;
     0.0577350269189626;
     0.2871305767251709;
     0.2022129403845741;
     -0.3559016137228031;
     0.447213595499958;
     0.05773502691896256;
     0.03726461792122844;
     0.271286303307133;
     0.4183480487454616;
     0.447213595499958;
     0.05773502691896257;
     0.1515575558927682;
     -0.4677758485215264;
     0.0906424999201606;
     0.447213595499958;
     0.05773502691896258;
     -0.4759527505391677;
     -0.005723395170180857;
     -0.153088934942819;
```

```
     0.4472135954999578];
y = [63;
     69;
     68;
     71;
     68;
     65;
     65;
     66;
     72;
     67;
     70;
     67];
wk = [-1;
     4.888782090057172e-17;
     3.783865581974411e-17;
     3.205148747669446e-17;
     2.127304033926049e-17;
     -7.232084855102773e-18;
     0.574261153450342;
     -0.282082169979652;
     -0.7685400295478783;
     -1.766548161010173e-16;
     1.072333245697965e-17;
     0.4044258807691484;
     -0.7184712955241166;
     0.5658963725559621;
     -3.152388336389935e-17;
     -6.842359129558908e-17;
     -0.7118032274456063;
     -0.635789781993599;
     -0.2985088248453474;
     7.896994727506343e-17;
     2.12730403392605e-17;
     1.57267977670022e-16;
     -8.430959034105682e-17;
     -1.677854456096144e-16;
     0.9999999999999999;
     0.2886751345948128;
     0.03726461792122845;
     0.271286303307133;
     0.4183480487454616;
     5;
     3.25;
     5.25;
     4.25;
     5.25;
     6.070050924028785e-16;
     0.7071067811865474;
     -0.5700260616371526;
     0.9995120760870788;
     1;
     0.7071067811865475;
     0.8216266117004956;
     0.03123475237772124;
     5.777618349603308e-32;
     -0.7071067811865476;
     -0.5700260616371526];
[rssOut, covOut, qOut, b, se, res, ifail] = ...
    g02dg(weight, wt, rss, ip, irank, cov, q, svd, p, y, wk)
```

```
rssOut =
   24.0000
covOut =
    0.1600
    0.0400
    0.7600
    0.0400
   -0.2400
    0.7600
```

```
    0.0400
   -0.2400
   -0.2400
    0.7600
    0.0400
   -0.2400
   -0.2400
   -0.2400
    0.7600
qOut =
 -234.1155   -3.4641   -0.8660   -0.8660   -0.8660   -0.8660
   -4.5000    0.2543    1.5000   -0.5000   -0.5000   -0.5000
    3.5355    0.2543    0.2464   -1.4142    0.7071    0.7071
   -4.4907    0.2543    0.2464   -0.1494   -1.2247    1.2247
   -0.5880    0.2543    0.2464   -0.1494   -0.2512    0.0000
   -2.1727    0.2543    0.2464    0.4236    0.0476    0.0585
   -2.6415    0.2543    0.2464   -0.1494   -0.2512    0.4718
    1.6690    0.2543   -0.3431   -0.0580   -0.0975    0.2227
    0.9087    0.2543    0.2464   -0.1494    0.4137    0.1941
    2.6690    0.2543   -0.3431   -0.0580   -0.0975    0.2227
   -1.0913    0.2543    0.2464   -0.1494    0.4137    0.1941
   -0.1727    0.2543    0.2464    0.4236    0.0476    0.0585
b =
   54.0667
   11.2667
   12.6000
   16.9333
   13.2667
se =
    0.4000
    0.8718
    0.8718
    0.8718
    0.8718
res =
   -2.3333
    1.6667
    1.3333
   -0.0000
    0.6667
   -1.6667
   -2.3333
    0.6667
    1.0000
    1.6667
   -1.0000
    0.3333
ifail =
           0
```