NAG Toolbox for MATLAB

g02hk

1 Purpose

g02hk computes a robust estimate of the covariance matrix for an expected fraction of gross errors.

2 Syntax

```
[cov, theta, nit, ifail] = g02hk(n, x, eps, nitmon, tol, 'm', m,
'maxit', maxit)
```

3 Description

For a set of n observations on m variables in a matrix X, a robust estimate of the covariance matrix, C, and a robust estimate of location, θ, are given by

$$C = \tau^2 \left(A^{\mathrm{T}} A\right)^{-1},$$

where τ^2 is a correction factor and A is a lower triangular matrix found as the solution to the following equations:

$$z_i = A(x_i - \theta),$$

$$\frac{1}{n} \sum_{i=1}^{n} w\left(\|z_i\|_2\right) z_i = 0,$$

and

$$\frac{1}{n} \sum_{i=1}^{n} u\left(\|z_i\|_2\right) z_i z_i^{\mathrm{T}} - I = 0,$$

where x_i is a vector of length m containing the elements of the ith row of $\mathbf{x}$,

z_i is a vector of length m,

I is the identity matrix and 0 is the zero matrix,

and w and u are suitable functions.

g02hk uses weight functions:

$$u(t) = \frac{a_u}{t^2}, \quad \text{if } t < a_u^2$$

$$u(t) = 1, \quad \text{if } a_u^2 \leq t \leq b_u^2$$

$$u(t) = \frac{b_u}{t^2}, \quad \text{if } t > b_u^2$$

and

$$w(t) = 1, \quad \text{if } t \leq c_w$$

$$w(t) = \frac{c_w}{t}, \quad \text{if } t > c_w$$

for constants a_u, b_u and c_w.

These functions solve a minimax problem considered by Huber (see Huber 1981). The values of a_u, b_u and c_w are calculated from the expected fraction of gross errors, ϵ (see Huber 1981 and Marazzi 1987a). The expected fraction of gross errors is the estimated proportion of outliers in the sample.

In order to make the estimate asymptotically unbiased under a Normal model a correction factor, τ^2, is calculated, (see Huber 1981 and Marazzi 1987a).

The matrix C is calculated using g02hl. Initial estimates of θ_j, for $j = 1, 2, \ldots, m$, are given by the median of the jth column of X and the initial value of A is based on the median absolute deviation (see Marazzi 1987a). g02hk is based on routines in ROBETH; see Marazzi 1987a.

4 References

Huber P J 1981 *Robust Statistics* Wiley

Marazzi A 1987a Weights for bounded influence regression in ROBETH *Cah. Rech. Doc. IUMSP, No. 3 ROB 3* Institut Universitaire de Médecine Sociale et Préventive, Lausanne

5 Parameters

5.1 Compulsory Input Parameters

1: **n – int32 scalar**

n, the number of observations.

Constraint: $\mathbf{n} > 1$.

2: **x(ldx,m) – double array**

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

$\mathbf{x}(i,j)$ must contain the ith observation for the jth variable, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.

3: **eps – double scalar**

ϵ, the expected fraction of gross errors expected in the sample.

Constraint: $0.0 \leq \mathbf{eps} < 1.0$.

4: **nitmon – int32 scalar**

Indicates the amount of information on the iteration that is printed.

nitmon > 0

The value of A, θ and δ (see Section 7) will be printed at the first and every **nitmon** iterations.

nitmon ≤ 0

No iteration monitoring is printed.

When printing occurs the output is directed to the current advisory message unit (see x04ab).

5: **tol – double scalar**

The relative precision for the final estimates of the covariance matrix.

Constraint: $\mathbf{tol} > 0.0$.

5.2 Optional Input Parameters

1: **m – int32 scalar**

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

m, the number of columns of the matrix X, i.e., number of independent variables.

Constraint: $1 \leq \mathbf{m} \leq \mathbf{n}$.

2: **maxit – int32 scalar**

The maximum number of iterations that will be used during the calculation of the covariance matrix.

Constraint: **maxit** > 0.

5.3 Input Parameters Omitted from the MATLAB Interface

ldx, wk

5.4 Output Parameters

1: **cov**$(\mathbf{m} \times (\mathbf{m}+\mathbf{1})/\mathbf{2})$ **– double array**

A robust estimate of the covariance matrix, C. The upper triangular part of the matrix C is stored packed by columns. C_{ij} is returned in **cov**$(j \times (j-1)/2 + i)$, $i \leq j$.

2: **theta**$(\mathbf{m})$ **– double array**

The robust estimate of the location parameters θ_j, for $j = 1, 2, \ldots, m$.

3: **nit – int32 scalar**

The number of iterations performed.

4: **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, **n** ≤ 1,
or **m** < 1,
or **n** $<$ **m**,
or **ldx** $<$ **n**,
or **eps** < 0.0,
or **eps** ≥ 1.0,
or **tol** ≤ 0.0,
or **maxit** ≤ 0.

ifail $= 2$

On entry, a variable has a constant value, i.e., all elements in a column of X are identical.

ifail $= 3$

The iterative procedure to find C has failed to converge in **maxit** iterations.

ifail $= 4$

The iterative procedure to find C has become unstable. This may happen if the value of **eps** is too large for the sample.

7 Accuracy

On successful exit the accuracy of the results is related to the value of **tol**; see Section 5. At an iteration let

(i) $d1 =$ the maximum value of the absolute relative change in A

(ii) $d2 =$ the maximum absolute change in $u\left(\|z_i\|_2\right)$

(iii) $d3 =$ the maximum absolute relative change in θ_j

and let $\delta = \max(d1, d2, d3)$. Then the iterative procedure is assumed to have converged when $\delta <$ **tol**.

8 Further Comments

The existence of A, and hence c, will depend upon the function u (see Marazzi 1987a); also if X is not of full rank a value of A will not be found. If the columns of X are almost linearly related, then convergence will be slow.

9 Example

```
n = int32(10);
x = [3.4, 6.9, 12.2;
     6.4, 2.5, 15.1;
     4.9, 5.5, 14.2;
     7.3, 1.9, 18.2;
     8.800000000000001, 3.6, 11.7;
     8.4, 1.3, 17.9;
     5.3, 3.1, 15;
     2.7, 8.1, 7.7;
     6.1, 3, 21.9;
     5.3, 2.2, 13.9];
eps = 0.1;
nitmon = int32(0);
tol = 5e-05;
[cov, theta, nit, ifail] = g02hk(n, x, eps, nitmon, tol)
```

```
cov =
    3.4611
   -3.6806
    5.3477
    4.6818
   -6.6445
   14.4386
theta =
    5.8178
    3.6813
   15.0369
nit =
          23
ifail =
           0
```