NAG Toolbox for MATLAB

g08ea

1 Purpose

g08ea performs a runs up (or a runs down) test on a sequence of observations.

2 Syntax

```
[nruns, ncount, ex, cov, chi, df, prob, ifail] = g08ea(cl, x, m, nruns,
ncount, 'n', n, 'maxr', maxr)
```

3 Description

Runs tests may be used to investigate for trends in a sequence of observations. g08ea computes statistics for the runs up test. If the runs down test is desired then each observation must be multiplied by -1 before g08ea is called with the modified vector of observations. g08ea may be used in two different modes:

(i) a single call to g08ea which computes all test statistics after counting the runs;

(ii) multiple calls to g08ea with the final test statistics only being computed in the last call.

The second mode is necessary if all the data do not fit into the memory. See parameter **cl** in Section 5 for details on how to invoke each mode.

A run up is a sequence of numbers in increasing order. A run up ends at x_k when $x_k > x_{k+1}$ and the new run then begins at x_{k+1}. g08ea counts the number of runs up of different lengths. Let c_i denote the number of runs of length i, for $i = 1, 2, \ldots, r - 1$. The number of runs of length r or greater is then denoted by c_r.

An unfinished run at the end of a sequence is not counted unless the sequence is part of an initial or intermediate call to g08ea (i.e., unless there is another call to g08ea to follow) in which case the unfinished run is used together with the beginning of the next sequence of numbers input to g08ea in the next call. The following is a trivial example.

Suppose we called g08ea twice with the following two sequences:

(0.20 0.40 0.45 0.40 0.15 0.75 0.95 0.23) and

(0.27 0.40 0.25 0.10 0.34 0.39 0.61 0.12).

Then after the second call g08ea would have counted the runs up of the following lengths:

3, 1, 3, 3, 1, and 4.

When the counting of runs is complete g08ea computes the expected values and covariances of the counts, c_i. For the details of the method used see Knuth 1981. An approximate χ^2 statistic with r degrees of freedom is computed, where

$$X^2 = (c - \mu_c)^{\mathrm{T}} \Sigma_c^{-1} (c - \mu_c),$$

where

c is the vector of counts, c_i, for $i = 1, 2, \ldots, r$,
μ_c is the vector of expected values,
e_i, for $i = 1, 2, \ldots, r$, where e_i is the expected value for c_i under the null hypothesis of randomness, and
Σ_c is the covariance matrix of c under the null hypothesis.

The use of the χ^2-distribution as an approximation to the exact distribution of the test statistic improves as the expected values increase.

You may specify the total number of runs to be found. If the specified number of runs is found before the end of a sequence g08ea will exit before counting any further runs. The number of runs actually counted and used to compute the test statistic is returned via **nruns**.

4 References

Dagpunar J 1988 *Principles of Random Variate Generation* Oxford University Press

Knuth D E 1981 *The Art of Computer Programming (Volume 2)* (2nd Edition) Addison–Wesley

Morgan B J T 1984 *Elements of Simulation* Chapman and Hall

Ripley B D 1987 *Stochastic Simulation* Wiley

5 Parameters

5.1 Compulsory Input Parameters

1: **cl – string**

Must specify the type of call to g08ea.

cl = 'S'

> This is the one and only call to g08ea (single call mode). All data are to be input at once. All test statistics are computed after the counting of runs is complete.

cl = 'F'

> This is the first call to the function. All initializations are carried out and the counting of runs begins. The final test statistics are not computed since further calls will be made to g08ea.

cl = 'I'

> This is an intermediate call during which the counts of runs are updated. The final test statistics are not computed since further calls will be made to g08ea.

cl = 'L'

> This is the last call to g08ea. The test statistics are computed after the final counting of runs is completed.

Constraint: **cl** = 'S', 'F', 'I' or 'L'.

2: **x**(**n**) **– double array**

The sequence of observations.

3: **m – int32 scalar**

The maximum number of runs to be sought. If $\mathbf{m} \leq 0$ then no limit is placed on the number of runs that are found.

m must not be changed between calls to g08ea.

Constraint: if $\mathbf{m} \leq \mathbf{n}$, **cl** = 'S'.

4: **nruns – int32 scalar**

If **cl** = 'S' or 'F', **nruns** need not be set.

If **cl** = 'I' or 'L', **nruns** must contain the value returned by the previous call to g08ea.

5: **ncount**(**maxr**) **– int32 array**

If **cl** = 'S' or 'F', **ncount** need not be set.

If **cl** = 'I' or 'L', **ncount** must contain the values returned by the previous call to g08ea.

5.2 Optional Input Parameters

1: **n – int32 scalar**

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

n, the length of the current sequence of observations.

Constraints:

if **cl** = 'S', **n** ≥ 3;
n ≥ 1 otherwise.

2: **maxr – int32 scalar**

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

r, the length of the longest run for which tabulation is desired. That is, all runs with length greater than or equal to *r* are counted together.

maxr must not be changed between calls to g08ea.

Constraint: **maxr** ≥ 1 and if **cl** = 'S', **maxr** < **n**.

5.3 Input Parameters Omitted from the MATLAB Interface

ldcov, wrk, lwrk

5.4 Output Parameters

1: **nruns – int32 scalar**

The number of runs actually found.

2: **ncount(maxr) – int32 array**

The counts of runs of the different lengths, c_i, for $i = 1, 2, \ldots, r$.

3: **ex(maxr) – double array**

If **cl** = 'S' or 'L', (i.e., if it is the final exit) then **ex** contains the expected values of the counts, e_i, for $i = 1, 2, \ldots, r$.

Otherwise the elements of **ex** are not set.

4: **cov(ldcov,maxr) – double array**

If **cl** = 'S' or 'L' (i.e., if it is the final exit) then **cov** contains the covariance matrix of the counts, Σ_c.

Otherwise the elements of **cov** are not set.

5: **chi – double scalar**

If **cl** = 'S' or 'L' (i.e., if it is the final exit) then **chi** contains the approximate χ^2 test statistic, X^2.

Otherwise **chi** is not set.

6: **df – double scalar**

If **cl** = 'S' or 'L' (i.e., if it is the final exit) then **df** contains the degrees of freedom of the χ^2 statistic.

Otherwise **df** is not set.

7: **prob – double scalar**

If **cl** = 'S' or 'L', (i.e., if it is the final exit) then **prob** contains the upper tail probability corresponding to the χ^2 test statistic, i.e., the significance level.

Otherwise **prob** is not set.

8: **ifail – int32 scalar**

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

6 Error Indicators and Warnings

Note: g08ea may return useful information for one or more of the following detected errors or warnings.

ifail = 1

On entry, **cl** $\neq$ 'S', 'F', 'I' or 'L'.

ifail = 2

On entry, **n** < 3 if **cl** = 'S',
or **n** < 1 otherwise.

ifail = 3

On entry, with **cl** = 'S', **m** $>$ **n**.

ifail = 4

On entry, **maxr** < 1,
or **maxr** $\geq$ **n** and **cl** = 'S'.

ifail = 5

On entry, **ldcov** $<$ **maxr**.

ifail = 6

On entry, **lwrk** $<$ **maxr** $\times$ $($**maxr** $+ 5)/2 + 1$.

ifail = 7

There is a tie in the sequence of observations.

ifail = 8

The total length of the runs found is less than **maxr**.

ifail = 9

The covariance matrix stored in **cov** is not positive-definite. This may be because the value of **maxr** is too large relative to the full length of the series. Thus the approximate χ^2 test statistic cannot be computed.

ifail = 10

The number of runs requested were not found. All statistics are still computed and the information returned may still be of use.

7 Accuracy

The computations are believed to be stable. The computation of **prob** given the values of **chi** and **df** will obtain a relative accuracy of five significant figures for most cases.

8 Further Comments

The time taken by g08ea increases with the number of observations *n*, and also depends to some extent on whether the call to g08ea is an only, first, intermediate or last call.

9 Example

```
cl = ’F’;
m = int32(0);
nruns = int32(0);
ncount = [int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0)];
g05cb(int32(0));
[x] = g05fa(0, 1, int32(1000));
[nrunsOut, ncountOut, ex, cov, chi, df, prob, ifail] = g08ea(cl, x, m,
nruns, ncount)
```

```
nrunsOut =
         495
ncountOut =
         162
         209
          86
          32
           4
           2
ex =
     0
     0
     0
     0
     0
     0
cov =
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
chi =
     0
df =
     0
prob =
     0
ifail =
           0
```