

NAG Toolbox

nag_stat_quantiles (g01am)

1 Purpose

nag_stat_quantiles (g01am) finds specified quantiles from a vector of unsorted data.

2 Syntax

```
[qv, ifail] = nag_stat_quantiles(rv, q, 'n', n, 'nq', nq)
[qv, ifail] = g01am(rv, q, 'n', n, 'nq', nq)
```

Note: the interface to this routine has changed since earlier releases of the toolbox:

At Mark 23: **rv** is no longer an output parameter.

3 Description

A quantile is a value which divides a frequency distribution such that there is a given proportion of data values below the quantile. For example, the median of a dataset is the 0.5 quantile because half the values are less than or equal to it; and the 0.25 quantile is the 25th percentile.

nag_stat_quantiles (g01am) uses a modified version of Singleton's 'median-of-three' Quicksort algorithm (Singleton (1969)) to determine specified quantiles of a vector of real values. The input vector is partially sorted, as far as is required to compute desired quantiles; for a single quantile, this is much faster than sorting the entire vector. Where necessary, linear interpolation is also carried out to return the values of quantiles which lie between two data points.

4 References

Singleton R C (1969) An efficient algorithm for sorting with minimal storage: Algorithm 347 *Comm. ACM* **12** 185–187

5 Parameters

5.1 Compulsory Input Parameters

1: **rv(n)** – REAL (KIND=nag_wp) array

The vector whose quantiles are to be determined.

2: **q(nq)** – REAL (KIND=nag_wp) array

The quantiles to be calculated, in ascending order. Note that these must be between 0.0 and 1.0, with 0.0 returning the smallest element and 1.0 the largest.

Constraints:

$$0.0 \leq \mathbf{q}(i) \leq 1.0, \text{ for } i = 1, 2, \dots, \mathbf{nq};$$

$$\mathbf{q}(i) \leq \mathbf{q}(i + 1), \text{ for } i = 1, 2, \dots, \mathbf{nq} - 1.$$

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the dimension of the array **rv**.

The number of elements in the input vector **rv**.

Constraint: **n** > 0.

2: **nq** – INTEGER

Default: the dimension of the array **q**.

The number of quantiles requested.

Constraint: **nq** > 0.

5.3 Output Parameters

1: **qv(nq)** – REAL (KIND=nag_wp) array

qv(*i*) contains the quantile specified by the value provided in **q**(*i*), or an interpolated value if the quantile falls between two data values.

2: **ifail** – INTEGER

ifail = 0 unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

Constraint: **n** > 0.

ifail = 2

Constraint: **nq** > 0.

ifail = 3

On entry, an element of **q** was less than 0.0 or greater than 1.0.

ifail = 4

On entry, **q** was not in ascending order.

ifail = 5

Internal error. Please contact NAG.

ifail = -99

An unexpected error has been triggered by this routine. Please contact NAG.

ifail = -399

Your licence key may have expired or may not have been installed correctly.

ifail = -999

Dynamic memory allocation failed.

7 Accuracy

Not applicable.

8 Further Comments

The average time taken by `nag_stat_quantiles` (`g01am`) is approximately proportional to $n \times (1 + \log(nq))$. The worst case time is proportional to n^2 but this is extremely unlikely to occur.

9 Example

This example computes a list of quantiles from an array of doubles and an array of point values.

9.1 Program Text

```
function g01am_example
fprintf('g01am example results\n\n');

rv = [4.9  7  3.9  9.5  1.3  3.1  9.7  0.3  8.5  0.6  6.2];
q  = [0.25 0.5 1];

[qv, ifail] = g01am(rv, q);

fprintf('Quantile    Result\n');
fprintf('%7.2f    %7.4f\n',[q; qv]);
```

9.2 Program Results

```
g01am example results

Quantile    Result
  0.25      2.2000
  0.50      4.9000
  1.00      9.7000
```
