NAG Toolbox

nag_stat_quantiles_stream_arbitrary (g01ap)

1 Purpose

nag_stat_quantiles_stream_arbitrary (g01ap) finds approximate quantiles from a large arbitrary-sized data stream using an out-of-core algorithm.

2 Syntax

```
[ind, np, qv, rcomm, icomm, ifail] = nag_stat_quantiles_stream_arbitrary(ind, rv, eps, q, rcomm, icomm, 'nb', nb, 'nq', nq, 'lrcomm', lrcomm, 'licomm', licomm)
[ind, np, qv, rcomm, icomm, ifail] = g01ap(ind, rv, eps, q, rcomm, icomm, 'nb', nb, 'nq', nq, 'lrcomm', lrcomm, 'licomm', licomm)
```

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.

nag_stat_quantiles_stream_arbitrary (g01ap) uses a slightly modified version of an algorithm described in a paper by Zhang and Wang (2007) to determine ϵ-approximate quantiles of a large arbitrary-sized data stream of real values, where ϵ is a user-defined approximation factor. Let m denote the number of data elements processed so far then, given any quantile $q \in [0.0, 1.0]$, an ϵ-approximate quantile is defined as an element in the data stream whose rank falls within $[(q-\epsilon)m, (q+\epsilon)m]$. In case of more than one ϵ-approximate quantile being available, the one closest to qm is used.

4 References

Zhang Q and Wang W (2007) A fast algorithm for approximate quantiles in high speed data streams *Proceedings of the 19th International Conference on Scientific and Statistical Database Management* IEEE Computer Society 29

5 Parameters

5.1 Compulsory Input Parameters

1: **ind** – INTEGER

On initial entry: must be set to 0.

Indicates the action required in the current call to nag_stat_quantiles_stream_arbitrary (g01ap).

ind $= 0$
Initialize the communication arrays and attempt to process the first **nb** values from the data stream. **eps**, **rv** and **nb** must be set and **licomm** must be at least 10.

ind $= 1$
Attempt to process the next block of **nb** values from the data stream. The calling program must update **rv** and (if required) **nb**, and re-enter nag_stat_quantiles_stream_arbitrary (g01ap) with all other parameters unchanged.

ind $= 2$
Continue calculation following the reallocation of either or both of the communication arrays **rcomm** and **icomm**.

ind $= 3$
Calculate the **nq** ϵ-approximate quantiles specified in **q**. The calling program must set **q** and **nq** and re-enter nag_stat_quantiles_stream_arbitrary (g01ap) with all other parameters unchanged. This option can be chosen only when **np** $\geq \lceil \exp(1.0)/\textbf{eps} \rceil$.

Constraint: **ind** $= 0$, 1, 2 or 3.

2: **rv**(:) – REAL (KIND=nag_wp) array

The dimension of the array **rv** must be at least **nb** if **ind** $= 0$, 1 or 2

If **ind** $= 0$, 1 or 2, the vector containing the current block of data, otherwise **rv** is not referenced.

3: **eps** – REAL (KIND=nag_wp)

Approximation factor ϵ.

Constraint: **eps** > 0.0 and **eps** ≤ 1.0.

4: **q**(:) – REAL (KIND=nag_wp) array

The dimension of the array **q** must be at least **nq** if **ind** $= 3$

If **ind** $= 3$, the quantiles to be calculated, otherwise **q** is not referenced. Note that $\textbf{q}(i) = 0.0$, corresponds to the minimum value and $\textbf{q}(i) = 1.0$ to the maximum value.

Constraint: if **ind** $= 3$, $0.0 \leq \textbf{q}(i) \leq 1.0$, for $i = 1, 2, \ldots,$ **nq**.

5: **rcomm**(**lrcomm**) – REAL (KIND=nag_wp) array

If **ind** $= 1$ or 2 then the first l elements of **rcomm** as supplied to nag_stat_quantiles_stream_arbitrary (g01ap) must be identical to the first l elements of **rcomm** returned from the last call to nag_stat_quantiles_stream_arbitrary (g01ap), where l is the value of **lrcomm** used in the last call. In other words, the contents of **rcomm** must not be altered between calls to this function. If **rcomm** needs to be reallocated then its contents must be preserved. If **ind** $= 0$ then **rcomm** need not be set.

6: **icomm**(**licomm**) – INTEGER array

If **ind** $= 1$ or 2 then the first l elements of **icomm** as supplied to nag_stat_quantiles_stream_arbitrary (g01ap) must be identical to the first l elements of **icomm** returned from the last call to nag_stat_quantiles_stream_arbitrary (g01ap), where l is the value of **licomm** used in the last call. In other words, the contents of **icomm** must not be altered between calls to this function. If **icomm** needs to be reallocated then its contents must be preserved. If **ind** $= 0$ then **icomm** need not be set.

5.2 Optional Input Parameters

1: **nb** – INTEGER

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

If **ind** $= 0$, 1 or 2, the size of the current block of data. The size of blocks of data in array **rv** can vary; therefore **nb** can change between calls to nag_stat_quantiles_stream_arbitrary (g01ap).

Constraint: if **ind** $= 0$, 1 or 2, **nb** > 0.

2: **nq** – INTEGER

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

If **ind** $= 3$, the number of quantiles requested, otherwise **nq** is not referenced.

Constraint: if **ind** $= 3$, **nq** > 0.

3: **lrcomm** – INTEGER

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

The dimension of the array **rcomm**.

Constraints:

if **ind** $= 0$, **lrcomm** ≥ 1;
otherwise **lrcomm** $\geq$ **icomm**(1).

4: **licomm** – INTEGER

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

The dimension of the array **icomm**.

Constraints:

if **ind** $= 0$, **licomm** ≥ 10;
otherwise **licomm** $\geq$ **icomm**(2).

5.3 Output Parameters

1: **ind** – INTEGER

Indicates output from the call.

ind $= 1$
nag_stat_quantiles_stream_arbitrary (g01ap) has processed **np** data points and expects to be called again with additional data.

ind $= 2$
Either one or more of the communication arrays **rcomm** and **icomm** is too small. The new minimum lengths of **rcomm** and **icomm** have been returned in **icomm**(1) and **icomm**(2) respectively. If the new minimum length is greater than the current length then the corresponding communication array needs to be reallocated, its contents preserved and nag_stat_quantiles_stream_arbitrary (g01ap) called again with all other parameters unchanged.

If there is more data to be processed, it is recommended that **lrcomm** and **licomm** are made significantly bigger than the minimum to limit the number of reallocations.

ind $= 3$
nag_stat_quantiles_stream_arbitrary (g01ap) has returned the requested ϵ-approximate quantiles in **qv**. These quantiles are based on **np** data points.

2: **np** – INTEGER

m, the number of elements processed so far.

3: **qv**$(:)$ – REAL (KIND=nag_wp) array

The dimension of the array **qv** will be **nq** if **ind** $= 3$

If **ind** $= 3$, **qv**(i) contains the ϵ-approximate quantiles specified by the value provided in **q**(i).

4: **rcomm**(**lrcomm**) – REAL (KIND=nag_wp) array

rcomm holds information required by subsequent calls to nag_stat_quantiles_stream_arbitrary (g01ap)

5: **icomm**(**licomm**) – INTEGER array

icomm(1) holds the minimum required length for **rcomm** and **icomm**(2) holds the minimum required length for **icomm**. The remaining elements of **icomm** are used for communication between subsequent calls to nag_stat_quantiles_stream_arbitrary (g01ap).

6: **ifail** – INTEGER

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

As an out-of-core function nag_stat_quantiles_stream_arbitrary (g01ap) will only perform certain argument checks when a data checkpoint (including completion of data input) is signaled. As such it will usually be inappropriate to halt program execution when an error is detected since any errors may be subsequently resolved without losing any processing already carried out. Therefore setting **ifail** to a value of -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. **When the value -1 or 1 is used it is essential to test the value of ifail on exit.**

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

Constraint: **ind** $= 0$, 1, 2 or 3.

ifail $= 2$

Constraint: $0.0 <$ **eps** ≤ 1.0.

ifail $= 3$

Constraint: if **ind** $= 0$, 1 or 2 then **nb** > 0.

ifail $= 4$

Constraint: **licomm** ≥ 10.

ifail $= 5$

Constraint: **lrcomm** ≥ 1.

ifail $= 6$

The contents of **icomm** have been altered between calls to this function.

ifail $= 7$

The contents of **rcomm** have been altered between calls to this function.

ifail $= 8$

Number of data elements streamed, $\langle value \rangle$ is not sufficient for a quantile query when . Supply more data or reprocess the data with a higher **eps** value.

ifail $= 9$

Constraint: if **ind** $= 3$ then **nq** > 0.

ifail $= 10$

Constraint: if **ind** $= 3$ then $0.0 \leq \mathbf{q}(i) \leq 1.0$ for all i.

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_stream_arbitrary (g01ap) scales as $\mathbf{np}\log(1/\epsilon\log(\epsilon\mathbf{np}))$.

It is not possible to determine in advance the final size of the communication arrays **rcomm** and **icomm** without knowing the size of the dataset. However, if a rough size (n) is known, the speed of the computation can be increased if the sizes of the communication arrays are not smaller than

$$\begin{aligned}\mathbf{lrcomm} &= (\log_2(n\times\mathbf{eps}+1.0)-2)\times\lceil 1.0/\mathbf{eps}\rceil+1+x+2\times\min(x,\lceil x/2.0\rceil+1)\times y+1\\ \mathbf{licomm} &= (\log_2(n\times\mathbf{eps}+1.0)-2)\times(2\times(\lceil 1.0/\mathbf{eps}\rceil+1)+1)+\\ &\quad 2\times(x+2\times\min(x,\lceil x/2.0\rceil+1)\times y)+y+11\end{aligned}$$

where

$$\begin{aligned}x &= \max(1,\lfloor\log(\mathbf{eps}\times n)/\mathbf{eps}\rfloor)\\ y &= \log_2(n/x+1.0)+1.\end{aligned}$$

9 Example

This example computes a list of ϵ-approximate quantiles. The data is processed in blocks of 20 observations at a time to simulate a situation in which the data is made available in a piecemeal fashion.

9.1 Program Text

```
function g01ap_example

fprintf('g01ap example results\n\n');

n   = 0;
tol = 0.2;
q   = [0.25 0.5 1.0];
nb  = 20;
% For this example we are using a string as the source of data.
datastream = ['34.01 57.95 44.88 22.04 28.84  4.43  0.32 20.82 ' ...
              '20.53 13.08  7.99 54.03 23.21 26.73 39.72  0.97 ' ...
              '39.05 38.78 19.38 51.34 24.08 12.41 58.11 35.90 ' ...
              '40.38 27.41 19.80  6.02 45.33 36.34 43.14 53.84 ' ...
       '39.49  9.04 36.74 58.72 59.95 15.41 33.05 39.54 ' ...
              '33.24 58.67 54.12 39.48 43.73 24.15 55.72  8.87 ' ...
              '40.47 46.18 20.36  6.95 36.86 49.24 56.83 43.87 ' ...
              '29.86 22.49 25.29 33.17'];

rcomm = zeros(100, 1);
icomm = zeros(400, 1, nag_int_name);
ind   = nag_int(0);
repeat = true;
pos    = 0;

while (repeat)
  if (ind==0 || ind==1)
    [rv, new_pos] = textscan(datastream(pos+1:end), '%f', nb);
    pos = pos+new_pos;

    nd = numel(rv{1});
    if nd == 0
      break;
    elseif nd < nb
      nb = nd;
```

```
      repeat = false;
    elseif pos == numel(datastream)
      repeat = false;
    end
    n = n+nb;
  end

  [ind, np, qv, rcomm, icomm, ifail] = ...
    g01ap(ind, rv{1}, tol, q, rcomm, icomm);

  % If ind=2, the communication arrays are too small, so extend them
  % and call the routine again with the same rv
  if ind == 2
    if numel(rcomm) < icomm(1)
      rcomm(icomm(1)) = 0;
    end
    if numel(icomm) < icomm(2)
      icomm(icomm(2)) = 0;
    end
  end
end

% Call again to calculate quantiles q
ind = nag_int(3);
[ind, np, qv, rcomm, icomm, ifail] = ...
  g01ap(ind, 0, tol, q, rcomm, icomm);

% Display results
fprintf('\nInput Data:\n %d observations\n eps = %5.2f\n\n', n, tol);
fprintf('Quantile     Result\n');
fprintf('%7.2f     %7.2f\n', [q; qv']);
```

9.2 Program Results

```
     g01ap example results

Input Data:
 60 observations
 eps =  0.20

Quantile     Result
   0.25       22.49
   0.50       39.54
   1.00       59.95
```