NAG Toolbox for MATLAB

g11bb

1 Purpose

g11bb computes a table from a set of classification factors using a given percentile or quantile, for example the median.

2 Syntax

```
[table, ncells, ndim, idim, icount, ifail] = g11bb(typ, weight, isf, lfac, ifac, percnt, y, wt, maxt, 'n', n, 'nfac', nfac)
```

3 Description

A data set may include both classification variables and general variables. The classification variables, known as factors, take a small number of values known as levels. For example, the factor sex would have the levels male and female. These can be coded as 1 and 2 respectively. Given several factors, a multi-way table can be constructed such that each cell of the table represents one level from each factor. For example, the two factors sex and habitat, habitat having three levels (inner-city, suburban and rural) define the 2×3 contingency table

Sex	**Habitat**		
	Inner-city	Suburban	Rural
Male			
Female			

For each cell statistics can be computed. If a third variable in the data set was age then for each cell the median age could be computed:

Sex	**Habitat**		
	Inner-city	Suburban	Rural
Male	24	31	37
Female	21.5	28.5	33

That is, the median age for all observations for males living in rural areas is 37, the median being the 50% quantile. Other quantiles can also be computed: the p percent quantile or percentile, q_p, is the estimate of the value such that p percent of observations are less than q_p. This is calculated in two different ways depending on whether the tabulated variable is continuous or discrete. Let there be m values in a cell and let $y_{(1)}, y_{(2)}, \ldots, y_{(m)}$ be the values for that cell sorted into ascending order. Also, associated with each value there is a weight, $w_{(1)}, w_{(2)}, \ldots, w_{(m)}$, which could represent the observed frequency for that value, with $W_j = \sum_{i=1}^{j} w_{(i)}$ and $W'_j = \sum_{i=1}^{j} w_{(i)} - \frac{1}{2}w_{(j)}$. For the p percentile let $p_w = (p/100)W_m$ and $p'_w = (p/100)W'_m$, then the percentiles for the two cases are as given below.

If the variable is discrete, that is, it takes only a limited number of (usually integer) values, then the percentile is defined as

$$\begin{cases} y_{(j)} & \text{if } W_{j-1} < p_W < W_j \\ \dfrac{y_{(j+1)} + y_{(j)}}{2} & \text{if } p_w = W_j. \end{cases}$$

If the data is continuous then the quantiles are estimated by linear interpolation.

$$\begin{cases} y_{(1)} & \text{if } p'_w \leq W'_1 \\ (1-f)y_{(j-1)} + f y_{(j)} & \text{if } W'_{j-1} < p'_w \leq W'_j \\ y_{(m)} & \text{if } p'_w > W'_m, \end{cases}$$

where $f = \left(p'_w - W'_{j-1}\right)/\left(W'_j - W'_{j-1}\right)$.

4 References

John J A and Quenouille M H 1977 *Experiments: Design and Analysis* Griffin

Kendall M G and Stuart A 1969 *The Advanced Theory of Statistics (Volume 1)* (3rd Edition) Griffin

5 Parameters

5.1 Compulsory Input Parameters

1: **typ – string**

Indicates if the variable to be tabulated is discrete or continuous.

If **typ** = 'D', the percentiles are computed for a discrete variable.

If **typ** = 'C', the percentiles are computed for a continuous variable using linear interpolation.

Constraint: **typ** = 'D' or 'C'.

2: **weight – string**

Indicates if there are weights associated with the variable to be tabulated.

If **weight** = 'U', weights are not input and unit weights are assumed.

If **weight** = 'W', weights must be supplied in **wt**.

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

3: **isf**(**nfac**) **– int32 array**

Indicates which factors in **ifac** are to be used in the tabulation.

If **isf**$(i) > 0$, the ith factor in **ifac** is included in the tabulation.

Note that if **isf**$(i) \leq 0$, for $i = 1, 2, \ldots,$ **nfac** then the statistic for the whole sample is calculated and returned in a 1×1 table.

4: **lfac**(**nfac**) **– int32 array**

The number of levels of the classifying factors in **ifac**.

Constraint: **lfac**$(i) \geq 2$ if **isf**$(i) > 0$, for $i = 1, 2, \ldots,$ **nfac**.

5: **ifac**(**ldf,nfac**) **– int32 array**

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

The **nfac** coded classification factors for the **n** observations.

Constraint: $1 \leq \mathbf{ifac}(i,j) \leq \mathbf{lfac}(j)$ for $i = 1,2,\ldots,\mathbf{n}$ and $j = 1,2,\ldots,\mathbf{nfac}$.

6: **percnt – double scalar**

p, the percentile to be tabulated.

Constraint: $0.0 < p < 100.0$.

7: **y(n) – double array**

The variable to be tabulated.

8: **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 **n** weights. Otherwise **wt** is not referenced.

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

9: **maxt – int32 scalar**

the maximum size of the table to be computed.

Constraint: **maxt** $\geq$ product of the levels of the factors included in the tabulation.

5.2 Optional Input Parameters

1: **n – int32 scalar**

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

the number of observations.

Constraint: $\mathbf{n} \geq 2$.

2: **nfac – int32 scalar**

Default: The dimension of the arrays **isf**, **lfac**, **ifac**, **idim**. (An error is raised if these dimensions are not equal.)

the number of classifying factors in **ifac**.

Constraint: $\mathbf{nfac} \geq 1$.

5.3 Input Parameters Omitted from the MATLAB Interface

ldf, iwk, wk

5.4 Output Parameters

1: **table(maxt) – double array**

The computed table. The **ncells** cells of the table are stored so that for any two factors the index relating to the factor occurring later in **lfac** and **ifac** changes faster. For further details see Section 8.

2: **ncells – int32 scalar**

The number of cells in the table.

3: **ndim – int32 scalar**

The number of factors defining the table.

4: **idim**(**nfac**) – **int32 array**

The first **ndim** elements contain the number of levels for the factors defining the table.

5: **icount**(**maxt**) – **int32 array**

A table containing the number of observations contributing to each cell of the table, stored identically to **table**.

6: **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** < 2,
or **nfac** < 1,
or **ldf** $<$ **n**,
or **typ** $\neq$ 'D' or 'C',
or **weight** $\neq$ 'U' or 'W',
or **percnt** ≤ 0.0,
or **percnt** ≥ 100.0.

ifail $= 2$

On entry, **isf**$(i) > 0$ and **lfac**$(i) \leq 1$, for some i,
or **ifac**$(i,j) < 1$, for some i,j,
or **ifac**$(i,j) >$ **lfac**(j), for some i,j,
or **maxt** is too small,
or **weight** $=$ 'W' and **wt**$(i) < 0.0$, for some i.

ifail $= 3$

At least one cell is empty.

7 Accuracy

Not applicable.

8 Further Comments

The tables created by g11bb and stored in **table** and **icount** are stored in the following way. Let there be n factors defining the table with factor k having l_k levels, then the cell defined by the levels $i_1, i_2, \ldots, i_n$ of the factors is stored in the mth cell given by:

$$m = 1 + \sum_{k=1}^{n} [(i_k - 1)c_k],$$

where $c_j = \prod_{k=j+1}^{n} l_k$, for $j = 1, 2, \ldots, n-1$ and $c_n = 1$.

9 Example

```
typ = 'C';
```

```
weight = 'U';
isf = [int32(0);
     int32(1);
     int32(1)];
lfac = [int32(3);
     int32(3);
     int32(6)];
ifac = [int32(1), int32(1), int32(1);
     int32(1), int32(2), int32(1);
     int32(1), int32(3), int32(1);
     int32(1), int32(1), int32(2);
     int32(1), int32(2), int32(2);
     int32(1), int32(3), int32(2);
     int32(1), int32(1), int32(3);
     int32(1), int32(2), int32(3);
     int32(1), int32(3), int32(3);
     int32(1), int32(1), int32(4);
     int32(1), int32(2), int32(4);
     int32(1), int32(3), int32(4);
     int32(1), int32(1), int32(5);
     int32(1), int32(2), int32(5);
     int32(1), int32(3), int32(5);
     int32(1), int32(1), int32(6);
     int32(1), int32(2), int32(6);
     int32(1), int32(3), int32(6);
     int32(2), int32(1), int32(1);
     int32(2), int32(2), int32(1);
     int32(2), int32(3), int32(1);
     int32(2), int32(1), int32(2);
     int32(2), int32(2), int32(2);
     int32(2), int32(3), int32(2);
     int32(2), int32(1), int32(3);
     int32(2), int32(2), int32(3);
     int32(2), int32(3), int32(3);
     int32(2), int32(1), int32(4);
     int32(2), int32(2), int32(4);
     int32(2), int32(3), int32(4);
     int32(2), int32(1), int32(5);
     int32(2), int32(2), int32(5);
     int32(2), int32(3), int32(5);
     int32(2), int32(1), int32(6);
     int32(2), int32(2), int32(6);
     int32(2), int32(3), int32(6);
     int32(3), int32(1), int32(1);
     int32(3), int32(2), int32(1);
     int32(3), int32(3), int32(1);
     int32(3), int32(1), int32(2);
     int32(3), int32(2), int32(2);
     int32(3), int32(3), int32(2);
     int32(3), int32(1), int32(3);
     int32(3), int32(2), int32(3);
     int32(3), int32(3), int32(3);
     int32(3), int32(1), int32(4);
     int32(3), int32(2), int32(4);
     int32(3), int32(3), int32(4);
     int32(3), int32(1), int32(5);
     int32(3), int32(2), int32(5);
     int32(3), int32(3), int32(5);
     int32(3), int32(1), int32(6);
     int32(3), int32(2), int32(6);
     int32(3), int32(3), int32(6)];
percnt = 50;
y = [274;
     361;
     253;
     325;
     317;
     339;
     326;
     402;
```

```
     336;
     379;
     345;
     361;
     352;
     334;
     318;
     339;
     393;
     358;
     350;
     340;
     203;
     397;
     356;
     298;
     382;
     376;
     355;
     418;
     387;
     379;
     432;
     339;
     293;
     322;
     417;
     342;
     82;
     297;
     133;
     306;
     352;
     361;
     220;
     333;
     270;
     388;
     379;
     274;
     336;
     307;
     266;
     389;
     333;
     353];
wt = [0];
maxt = int32(18);
[table, ncells, ndim, idim, icount, ifail] = ...
    g11bb(typ, weight, isf, lfac, ifac, percnt, y, wt, maxt)
```

```
table =
  226.0000
  320.2500
  299.5000
  385.7500
  348.0000
  334.7500
  329.2500
  343.2500
  365.2500
  370.5000
  327.2500
  378.0000
  185.5000
  328.7500
  319.5000
  339.2500
  286.2500
  350.2500
```

```
ncells =
          18
ndim =
           2
idim =
           3
           6
           0
icount =
           3
           3
           3
           3
           3
           3
           3
           3
           3
           3
           3
           3
           3
           3
           3
           3
           3
           3
ifail =
           0
```