NAG C Library Function Document

nag_tabulate_percentile (g11bbc)

1 Purpose

nag_tabulate_percentile (g11bbc) computes a table from a set of classification factors using a given percentile or quantile, for example the median.

2 Specification

void nag_tabulate_percentile (Nag_TabulateVar type, Integer n, Integer nfac,
const Integer sf[], const Integer lfac[], const Integer factor[],
Integer tdf, double perct, const double y[], const double wt[],
double table[], Integer maxat, Integer *nells, Integer *ndim,
Integer idim[], Integer count[], NagError *fail)

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 by 3 contingency table:

<table>
<thead>
<tr>
<th>Sex</th>
<th>Habitat</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inner-city</td>
</tr>
<tr>
<td>Male</td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Sex</th>
<th>Habitat</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inner-city</td>
</tr>
<tr>
<td>Male</td>
<td>24</td>
</tr>
<tr>
<td>Female</td>
<td>21.5</td>
</tr>
</tbody>
</table>

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} \left( w_{(i)} - \frac{1}{2} w_{(j)} \right). 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 takes only a limited number of (usually integer) values then the percentile is defined as:

\[
y_{(j)} \quad \text{if } W_{j-1} < p_{w} < W_{j} \\
\frac{y_{(j)} + y_{(j+1)}}{2} \quad \text{if } p_{w} = W_{j}
\]
If the data is continuous then the quantiles are estimated by linear interpolation.

\[
y(i) = \begin{cases} 
  y_{(1)} & \text{if } p'_w \leq W'_i \\
  (1 - f)y_{(j-1)} + fy_{(j)} & \text{if } W'_{j-1} < p'_w \leq W'_j \\
  y_{(m)} & \text{if } p'_w > W'_m 
\end{cases}
\]

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

4 Parameters

1. \textbf{type} – \texttt{Nag_TabulateVar} \hspace{1cm} \textit{Input}

   \textit{On entry:} indicates whether the variable to be tabulated is discrete or continuous.

   If \textbf{type} = \texttt{Nag_TabulateVarDiscr}, the percentiles are computed for a discrete variable.

   If \textbf{type} = \texttt{Nag_TabulateVarCont}, the percentiles are computed for a continuous variable
   using linear interpolation.

   \textit{Constraint:} \textbf{type} = \texttt{Nag_TabulateVarDiscr} or \texttt{Nag_TabulateVarCont}.

2. \textbf{n} – Integer \hspace{1cm} \textit{Input}

   \textit{On entry:} the number of observations.

   \textit{Constraint:} \textbf{n} \geq 2.

3. \textbf{nfac} – Integer \hspace{1cm} \textit{Input}

   \textit{On entry:} the number of classifying factors in \textbf{factor}.

   \textit{Constraint:} \textbf{nfac} \geq 1.

4. \textbf{sf[nfac]} – const Integer \hspace{1cm} \textit{Input}

   \textit{On entry:} indicates which factors in \textbf{factor} are to be used in the tabulation.

   If \textbf{sf}[i-1] > 0 the \textit{i}th factor in \textbf{factor} is included in the tabulation.

   Note that if \textbf{sf}[i-1] \leq 0, for \( i = 1, 2, \ldots, \textbf{nfac} \) then the statistic for the whole sample is calculated
   and returned in a 1 by 1 table.

5. \textbf{lfac[nfac]} – const Integer \hspace{1cm} \textit{Input}

   \textit{On entry:} the number of levels of the classifying factors in \textbf{factor}.

   \textit{Constraint:} if \textbf{sf}[i-1] > 0, \textbf{lfac}[i-1] \geq 2, for \( i = 1, 2, \ldots, \textbf{nfac} \).

6. \textbf{factor[n][tdf]} – const Integer \hspace{1cm} \textit{Input}

   \textit{On entry:} the \textbf{nfac} coded classification factors for the \textbf{n} observations.

   \textit{Constraint:} if \textbf{sf}[i-1] > 0, \( 1 \leq \textbf{factor}[i-1][j-1] \leq \textbf{lfac}[j-1] \), for \( i = 1, 2, \ldots, \textbf{n} ; j = 1, 2, \ldots, \textbf{nfac} \).

7. \textbf{tdf} – Integer \hspace{1cm} \textit{Input}

   \textit{On entry:} the second dimension of the array \textbf{factor} as declared in the function from which
   \texttt{nag_tabulate_percentile} is called.

   \textit{Constraint:} \textbf{tdf} \geq \textbf{nfac}.

8. \textbf{percent} – double \hspace{1cm} \textit{Input}

   \textit{On entry:} the percentile to be tabulated, \( p \).

   \textit{Constraint:} 0.0 \leq \textbf{percent} < 100.0.
9: \( y[n] \) – const double
\( Input \)
\( On \ entry: \) the variable to be tabulated.

10: \( wt[n] \) – const double
\( Input \)
\( On \ entry: \) \( wt \) must contain the \( n \) weights. Otherwise \( wt \) must be set to null pointer (double *)0.
\( Constraint: \) \( wt[i - 1] \geq 0.0, \) for \( i = 1, 2, \ldots, n. \)

11: \( table[max] \) – double
\( Output \)
\( On \ exit: \) 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 \( factor \) changes faster. For further details see Section 6.

12: \( max \) – Integer
\( Input \)
\( On \ entry: \) the maximum size of the table to be computed.
\( Constraint: \) \( max \geq \) product of the levels of the factors included in the tabulation.

13: \( ncells \) – Integer *
\( Output \)
\( On \ exit: \) the number of cells in the table.

14: \( ndim \) – Integer *
\( Output \)
\( On \ exit: \) the number of factors defining the table.

15: \( idim[nfac] \) – Integer
\( Output \)
\( On \ exit: \) the first \( ndim \) elements contain the number of levels for the factors defining the table.

16: \( count[max] \) – Integer
\( Output \)
\( On \ exit: \) a table containing the number of observations contributing to each cell of the table, stored identically to \( table. \)

17: \( fail \) – NagError *
\( Input/Output \)
The NAG error parameter (see the Essential Introduction).

5 Error Indicators and Warnings

**NE_INT_ARG_LT**
\( On \ entry, n \) must not be less than 2: \( n = <value>. \)
\( On \ entry, nfac \) must not be less than 1: \( nfac = <value>. \)

**NE_2_INT_ARG_LT**
\( On \ entry, tdf = <value> \) while \( nfac = <value>. \)
These parameters must satisfy \( tdf \geq nfac. \)

**NE_REAL**
\( On \ entry, percent = <value>. \)
\( Constraint: \) \( 0.0 < percent < 100.0. \)

**NE_BAD_PARAM**
\( On \ entry, \) parameter \( type \) had an illegal value.
NE_REAL_ARRAY_CONS

On entry, \( \text{wt}[^{\text{value}}] = ^{\text{value}} \).
Constraint: \( \text{wt}[i] \geq 0 \), for \( i = 0, 1, \ldots, n - 1 \).

NE_2_INT_ARRAY_CONS

On entry, \( sf[^{\text{value}}] = ^{\text{value}} \) while \( lfac[^{\text{value}}] = ^{\text{value}} \).
Constraint: if \( sf[i] > 0 \), \( lfac[i] \geq 2 \), for \( i = 0, 1, \ldots, nfac - 1 \).

NE_2D_INT_ARRAY_CONS

On entry, \( \text{factor}[^{\text{value}}][^{\text{value}}] = ^{\text{value}} \).
Constraint: \( \text{factor}[i][j] \geq 1 \), for \( i = 0, 1, \ldots, n - 1 \); \( j = 0, 1, \ldots, nfac - 1 \).

NE_2D_1D_INT ARRAYS_CONS

On entry, \( \text{factor}[^{\text{value}}][^{\text{value}}] = ^{\text{value}} \) while \( lfac[^{\text{value}}] = ^{\text{value}} \).
Constraint: \( \text{factor}[i][j] \leq lfac[j] \), for \( i = 0, 1, \ldots, n - 1 \); \( j = 0, 1, \ldots, nfac - 1 \).

NE_MAXT

The maximum size of the table to be computed, \texttt{maxt} is too small.

NE_CELL_EMPTY

At least one cell is empty.

NE_ALLOC_FAIL

Memory allocation failed.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please consult NAG for assistance.

6 Further Comments

The tables created by \texttt{nag_tabulate_percentile} and stored in \texttt{table} and \texttt{count} 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 \( m \)th cell given by:

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

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

6.1 Accuracy

Not applicable.

6.2 References

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

7 See Also

None.
8 Example
The data, given by John and Quenouille (1977), are for a 3 by 6 factorial experiment in 3 blocks of 18 units. The data is input in the order: blocks, factor with 3 levels, factor with 6 levels, yield, and the 3 by 6 table of treatment medians for yield over blocks is computed and printed.

8.1 Program Text
/* nag_tabulate_percentile (gllbcc) Example Program. */
*/

#include <stdio.h>
#include <nag.h>
#include <nag_stddlib.h>
#include <naggl1l.h>

int main (void)
{
    char type[2], weight[2];
    double percnt, *table=0, *wt=0, *wt.ptr, *y=0;
    Integer items, i, *count=0, *idim=0, *factor=0, ifail, *sf=0, j, k, tdf;
    Integer lmax, *lfac=0, maxt, n, ncells, ncol, ndim, nfac, nrow;
    Integer exit_status=0;
    Nag_TabulateVar type_enum;
    NagError fail;

#define FACTOR(I,J) factor[((I)-1)*nfac +(J)-1]

INIT.FAIL(fail);
Vprintf("gllbcc Example Program Results\n");

    /* Skip heading in data file */
Vscanf("%*[\n]");

Vscanf(" %s %s %ld %ld %lf", type, weight, &n, &nfac, &percnt);
lmax = 18;
maxt = lmax;
if (!(!sf = NAG_ALLOC(nfac, Integer))
    || !(!lfac = NAG_ALLOC(nfac, Integer))
    || !(!idim = NAG_ALLOC(nfac, Integer))
    || !(!factor = NAG_ALLOC(n*nfac, Integer))
    || !(!count = NAG_ALLOC(maxt, Integer))
    || !(!y = NAG_ALLOC(n, double))
    || !(!table = NAG_ALLOC(maxt, double))
    || !(!wt = NAG_ALLOC(n, double)))
{
    Vprintf("Allocation failure\n");
    exit_status = -1;
    goto END;
}

if (*weight == 'W' || *weight == 'V')
{
for (i = 1; i <= n; ++i)
{
    for (j = 1; j <= nfac; ++j)
        Vscanf("%ld", &FACTOR(i,j));
    Vscanf("%lf %lf ", &y[i - 1], &wt[i - 1]);
}  
if (otype == 'D')
    type_enum = Nag_TabulateVarDiscr;
else if (otype == 'C')
    type_enum = Nag_TabulateVarCont;
else
    type_enum = (Nag_TabulateVar)-999;
gllbbc(type_enum, n, nfac, sf, lfac, factor, tdf, percnt, y, wtptr, table, maxt, &ncells, &ndim, idim, count, &fail);
if (fail.code != NE_NOERROR)
{
    Vprintf("Error from gllbbc.
");
    exit_status = 1;
    goto END;
}
Vprintf("\n");
Vprintf("%s4.0f%s
", " Table for ", percnt, "th percentile");
Vprintf("\n");
ncol = idim[ndim - 1];
nrow = ncells / ncol;
k = 1;
for (i = 1; i <= nrow; ++i)
{
    for (items = 1, j = k; j <= k + ncol - 1; ++j, items++)
    {
        Vprintf("%8.2f(\%ld)\%s", table[j - 1],
            count[j - 1], items%6?":":"
        );
    }
    k += ncol;
}
END:
if (sf) NAG_FREE(sf);
if (lfac) NAG_FREE(lfac);
if (idim) NAG_FREE(idim);
8.2 Program Data

g11bbc Example Program Data

C U 54 3 50.0

1 1 1 274
1 2 1 361
1 3 1 253
1 1 2 325
1 2 2 317
1 3 2 339
1 1 3 326
1 2 3 402
1 3 3 336
1 1 4 379
1 2 4 345
1 3 4 361
1 1 5 352
1 2 5 334
1 3 5 318
1 1 6 339
1 2 6 393
1 3 6 358
2 1 1 350
2 2 1 340
2 3 1 203
2 1 2 397
2 2 2 356
2 3 2 298
2 1 3 382
2 2 3 376
2 3 3 355
2 1 4 418
2 2 4 387
2 3 4 379
2 1 5 432
2 2 5 339
2 3 5 293
2 1 6 322
2 2 6 417
2 3 6 342
3 1 1 82
3 2 1 297
3 3 1 133
3 1 2 306
3 2 2 352
3 3 2 361
3 1 3 220
3 2 3 333
3 3 3 270
8.3 Program Results

gllbbc Example Program Results

Table for 50th percentile

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>226.00</td>
<td>320.25</td>
<td>299.50</td>
<td>385.75</td>
<td>348.00</td>
<td>334.75</td>
</tr>
<tr>
<td>329.25</td>
<td>343.25</td>
<td>365.25</td>
<td>370.50</td>
<td>327.25</td>
<td>378.00</td>
</tr>
<tr>
<td>185.50</td>
<td>328.75</td>
<td>319.50</td>
<td>339.25</td>
<td>286.25</td>
<td>350.25</td>
</tr>
</tbody>
</table>