

NAG C Library Function Document

nag_sum_sqs (g02buc)

1 Purpose

nag_sum_sqs (g02buc) calculates the sample means and sums of squares and cross-products, or sums of squares and cross-products of deviations from the mean, in a single pass for a set of data. The data may be weighted.

2 Specification

```
void nag_sum_sqs (Nag_OrderType order, Nag_SumSquare mean, Integer n, Integer m,
  const double x[], Integer pdx, const double wt[], double *sw, double wmean[],
  double c[], NagError *fail)
```

3 Description

nag_sum_sqs (g02buc) is an adaptation of West's WV2 algorithm; see West (1979). This routine calculates the (optionally weighted) sample means and (optionally weighted) sums of squares and cross-products or sums of squares and cross-products of deviations from the (weighted) mean for a sample of n observations on m variables X_j , for $j = 1, 2, \dots, m$. The algorithm makes a single pass through the data.

For the first $i - 1$ observations let the mean of the j th variable be $\bar{x}_j(i - 1)$, the cross-product about the mean for the j th and k th variables be $c_{jk}(i - 1)$ and the sum of weights be W_{i-1} . These are updated by the i th observation, x_{ij} , for $j = 1, 2, \dots, m$, with weight w_i as follows:

$$W_i = W_{i-1} + w_i \quad \bar{x}_j(i) = \bar{x}_j(i-1) + \frac{w_i}{W_i}(x_j - \bar{x}_j(i-1)), \quad j = 1, 2, \dots, m$$

and

$$c_{jk}(i) = c_{jk}(i-1) + \frac{w_i}{W_i}(x_j - \bar{x}_j(i-1))(x_k - \bar{x}_k(i-1))W_{i-1}, \quad j = 1, 2, \dots, m; \quad k = j, j+1, \dots, m.$$

The algorithm is initialised by taking $\bar{x}_j(1) = x_{1j}$, the first observation, and $c_{ij}(1) = 0.0$.

For the unweighted case $w_i = 1$ and $W_i = i$ for all i .

Note that only the upper triangle of the matrix is calculated and returned packed by column.

4 References

Chan T F, Golub G H and Leveque R J (1982) *Updating Formulae and a Pairwise Algorithm for Computing Sample Variances* Compstat, Physica-Verlag

West D H D (1979) Updating mean and variance estimates: An improved method *Comm. ACM* **22** 532–555

5 Parameters

1: **order** – Nag_OrderType *Input*

On entry: the **order** parameter specifies the two-dimensional storage scheme being used, i.e., row-major ordering or column-major ordering. C language defined storage is specified by **order = Nag_RowMajor**. See Section 2.2.1.4 of the Essential Introduction for a more detailed explanation of the use of this parameter.

Constraint: **order = Nag_RowMajor** or **Nag_ColMajor**.

- 2: **mean** – Nag_SumSquare *Input*
On entry: indicates whether nag_sum_sqs (g02buc) is to calculate sums of squares and cross-products, or sums of squares and cross-products of deviations from the mean.
 If **mean** = **Nag_AboutMean**, the sums of squares and cross-products of deviations from the mean are calculated.
 If **mean** = **Nag_AboutZero**, the sums of squares and cross-products are calculated.
Constraint: **mean** = **Nag_AboutMean** or **Nag_AboutZero**.
- 3: **n** – Integer *Input*
On entry: the number of observations in the data set, n .
Constraint: $n \geq 1$.
- 4: **m** – Integer *Input*
On entry: the number of variables, m .
Constraint: $m \geq 1$.
- 5: **x**[*dim*] – const double *Input*
Note: the dimension, *dim*, of the array **x** must be at least $\max(1, \mathbf{pdx} \times \mathbf{m})$ when **order** = **Nag_ColMajor** and at least $\max(1, \mathbf{pdx} \times \mathbf{n})$ when **order** = **Nag_RowMajor**.
 Where **X**(i, j) appears in this document, it refers to the array element
 if **order** = **Nag_ColMajor**, $\mathbf{x}[(j - 1) \times \mathbf{pdx} + i - 1]$;
 if **order** = **Nag_RowMajor**, $\mathbf{x}[(i - 1) \times \mathbf{pdx} + j - 1]$.
On entry: **X**(i, j) must contain the i th observation on the j th variable, for $i = 1, 2, \dots, n$;
 $j = 1, 2, \dots, m$.
- 6: **pdx** – Integer *Input*
On entry: the stride separating matrix row or column elements (depending on the value of **order**) in the array **x**.
Constraints:
 if **order** = **Nag_ColMajor**, $\mathbf{pdx} \geq \mathbf{n}$;
 if **order** = **Nag_RowMajor**, $\mathbf{pdx} \geq \mathbf{m}$.
- 7: **wt**[*dim*] – const double *Input*
Note: the dimension, *dim*, of the array **wt** must be at least **n**.
On entry: the optional weights of each observation. If weights are not provided then **wt** must be set to the **NULL** pointer, i.e., (double *)0, otherwise **wt**[i] must contain the weight for the $i - 1$ th observation.
Constraint: if **wt** is not **NULL**, $\mathbf{wt}[i] \geq 0.0$ for $i = 0, 1, \dots, n - 1$.
- 8: **sw** – double * *Output*
On exit: the sum of weights.
 If **wt** is **NULL**, then **sw** contains the number of observations, n .
- 9: **wmean**[**m**] – double *Output*
On exit: the sample means. **wmean**[$j - 1$] contains the mean for the j th variable.

- 10: **c**[*dim*] – double *Output*
Note: the dimension, *dim*, of the array **c** must be at least $(\mathbf{m} \times \mathbf{m} + \mathbf{m})/2$.
On exit: the cross-products.
 If **mean** = **Nag_AboutMean**, then **c** contains the upper triangular part of the matrix of (weighted) sums of squares and cross-products of deviations about the mean.
 If **mean** = **Nag_AboutZero**, then **c** contains the upper triangular part of the matrix of (weighted) sums of squares and cross-products.
 These are stored packed by columns, i.e., the cross-product between the *j*th and *k*th variable, $k \geq j$, is stored in $\mathbf{c}(k \times (k - 1)/2 + j)$.
- 11: **fail** – NagError * *Input/Output*
 The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT

On entry, **n** = *value*.
 Constraint: **n** \geq 1.

On entry, **pdx** = *value*.
 Constraint: **pdx** $>$ 0.

On entry, **m** = *value*.
 Constraint: **m** \geq 1.

NE_INT_2

On entry, **pdx** = *value*, **n** = *value*.
 Constraint: **pdx** \geq **n**.

On entry, **pdx** = *value*, **m** = *value*.
 Constraint: **pdx** \geq **m**.

NE_REAL_ARRAY_ELEM_CONS

On entry, **wt**[*value*] $<$ 0.0.

NE_ALLOC_FAIL

Memory allocation failed.

NE_BAD_PARAM

On entry, parameter *value* had an illegal value.

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.

7 Accuracy

For a detailed discussion of the accuracy of this algorithm see Chan *et al.* (1982) or West (1979).

8 Further Comments

nag_cov_to_corr (g02buc) may be used to calculate the correlation coefficients from the cross-products of deviations about the mean. The cross-products of deviations about the mean may be scaled to give a variance-covariance matrix.

The means and cross-products produced by nag_sum_sqs (g02buc) may be updated by adding or removing observations using nag_sum_sqs_update (g02btc).

9 Example

None.
