

## NAG C Library Function Document

### nag\_sum\_sqs\_update (g02btc)

#### 1 Purpose

nag\_sum\_sqs\_update (g02btc) updates the sample means and sums of squares and cross-products, or sums of squares and cross-products of deviations about the mean, for a new observation. The data may be weighted.

#### 2 Specification

```
void nag_sum_sqs_update (Nag_SumSquare mean, Integer m, double wt,
    const double x[], Integer incx, double *sw, double xbar[], double c[],
    NagError *fail)
```

#### 3 Description

nag\_sum\_sqs\_update (g02btc) is an adaptation of West's WV2 algorithm; see West (1979). This routine updates the weighted means of variables and weighted sums of squares and cross-products or weighted sums of squares and cross-products of deviations about the mean for observations on  $m$  variables  $X_j$ , for  $j = 1, 2, \dots, m$ . 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

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

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  $i$ .

#### 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: **mean** – Nag\_SumSquare *Input*

*On entry:* indicates whether nag\_sum\_sqs\_update (g02btc) is to calculate sums of squares and cross-products, or sums of squares and cross-products of deviations about the mean.

If **mean** = **Nag\_AboutMean**, the sums of squares and cross-products of deviations about the mean are calculated.

If **mean** = **Nag\_AboutZero**, the sums of squares and cross-products are calculated.

*Constraint:* **mean** = **Nag\_AboutMean** or **Nag\_AboutZero**.

- 2: **m** – Integer *Input*  
*On entry:* the number,  $m$ , of variables.  
*Constraint:*  $m \geq 1$ .
- 3: **wt** – double *Input*  
*On entry:* the weight to use for the current observation,  $w_i$ .  
 For unweighted means and cross-products set **wt** = 1.0. The use of a suitable negative value of **wt**, e.g.,  $-w_i$  will have the effect of deleting the observation.
- 4: **x[dim]** – const double *Input*  
**Note:** the dimension,  $dim$ , of the array **x** must be at least  $m \times incx$ .  
*On entry:* **x**[( $j - 1$ )**incx**] must contain the value of the  $j$ th variable for the current observation,  $j = 1, 2, \dots, m$ .
- 5: **incx** – Integer *Input*  
*On entry:* the increment of **x**.  
*Constraint:* **incx** > 0.
- 6: **sw** – double \* *Input/Output*  
*On entry:* the sum of weights for the previous observations,  $W_{i-1}$ .  
 If **sw** = 0.0, the update procedure is initialised.  
 If **sw** + **wt** = 0.0, then all elements of **xbar** and **c** are set to zero.  
*Constraint:* **sw**  $\geq$  0.0 and **sw** + **wt**  $\geq$  0.0.  
*On exit:* **sw** contains the updated sum of weights,  $W_i$ .
- 7: **xbar[m]** – double *Input/Output*  
*On entry:* **xbar**[ $j - 1$ ] must contain the weighted mean of the  $j$ th variable for the previous ( $i - 1$ ) observations,  $\bar{x}_j(i - 1)$ , for  $j = 1, 2, \dots, m$ .  
*On exit:* **xbar**[ $j - 1$ ] contains the weighted mean of the  $j$ th variable,  $\bar{x}_j(i)$ , for  $j = 1, 2, \dots, m$ .
- 8: **c[dim]** – double *Input/Output*  
**Note:** the dimension,  $dim$ , of the array **c** must be at least  $(m \times m + m)/2$ .  
*On entry:* if **sw**  $\neq$  0.0, **c** must contain the upper triangular part of the matrix of weighted sums of squares and cross-products or weighted sums of squares and cross-products of deviations about the mean. It is stored packed form by column, i.e., the cross-product between the  $j$ th and  $k$ th variable,  $k \geq j$ , is stored in **c**[ $k \times (k - 1)/2 + j - 1$ ].  
*On exit:* the update sums of squares and cross-products stored as on input.
- 9: **fail** – NagError \* *Input/Output*  
 The NAG error parameter (see the Essential Introduction).

## 6 Error Indicators and Warnings

### NE\_INT

On entry, **incx** =  $\langle value \rangle$ .

Constraint: **incx**  $\geq$  1.

On entry, **m** =  $\langle value \rangle$ .

Constraint: **m**  $\geq$  1.

**NE\_REAL**

On entry, **sw** =  $\langle value \rangle$ .  
Constraint: **sw**  $\geq$  0.0.

**NE\_SUM\_WEIGHT**

On entry,  $(\mathbf{sw} + \mathbf{wt}) < 0.0$ :  $(\mathbf{sw} + \mathbf{wt}) = \langle value \rangle$ .

**NE\_BAD\_PARAM**

On entry, parameter  $\langle value \rangle$  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 method see Chan *et al.* (1982) and West (1979).

## 8 Further Comments

nag\_sum\_sqs\_update (g02btc) may be used to update the results returned by nag\_sum\_sqs (g02buc).

nag\_cov\_to\_corr (g02bwc) may be used to calculate the correlation matrix from the matrix of sums of squares and cross-products of deviations about the mean.

## 9 Example

None.

---