Utility: nagdmc_dsu

Purpose

nagdmc_dsu computes means and, optionally, standard deviations of data values using a single pass through the data.

Declaration

```
#include <nagdmc.h>

void nagdmc_dsu(long rec1, long nvar, long nrec, long dblk, double data[],
                void (*dfun)(long, long, double [], char *, int *), char *comm,
                long chunksize, long iwts, double xbar[], double s[],
                double *wsum, long *nzw, int *info);
```

Parameters

1: **rec1** – `long` *Input*

On entry: the index in the data of the first data record used in the analysis.

Constraint: **rec1** ≥ 0.

2: **nvar** – `long` *Input*

On entry: the number of variables in the data.

Constraint: **nvar** ≥ 1.

3: **nrec** – `long` *Input*

On entry: the number of consecutive records, beginning at **rec1**, used in the analysis.

Constraint: **nrec** > 1.

4: **dblk** – `long` *Input*

On entry: the total number of records in the data block.

Constraint: **dblk** $\geq$ **rec1** + **nrec**.

5: **data**[**dblk** $*$ **nvar**] – `double` *Input*

On entry: the data values for the jth variable (for $j = 0, 1, \ldots,$ **nvar**-1) are stored in **data**[$i*$**nvar**$+j$], for $i = 0, 1, \ldots,$ **dblk** $- 1$. When the data function is used, **data** is not referenced.

6: **dfun** – function supplied by user *External Procedure*

On entry: the pointer to a data function supplied by the user.

Constraint: if **dfun** is a valid pointer, **data** must be 0.

The specification of **dfun** is:

```
void dfun(long irec, long chunksize, double x[], char *comm, int *ierr)
```

1: **irec** – `long` *Input*

On entry: the index in the data of the first record returned.

2: **chunksize** – `long` *Input*

On entry: the number of consecutive records returned.

3: **x**[**chunksize**$*$**nvar**] – `double` *Output*

On exit: data values for the jth variable (for $j = 0, 1, \ldots,$ **nvar** $- 1$) must be returned in **x**[$i *$ **nvar** $+ j$], for $i = 0, 1, \ldots,$ **chunksize** $- 1$.

4: **comm** – `char *` *Input*

On entry: a communication parameter allowing additional information to be passed to **dfun**. This parameter is passed 'as is' through the calling function.

5: **ierr** – `int *` *Output*

On exit: if the value pointed to by **ierr** on return is greater than 100, the NAG DMC function will terminate immediately and **info** will point to this value.

7: **comm** – `char *` *Input*

On entry: a communication parameter allowing additional information to be passed to **dfun**. This parameter is passed 'as is' through the calling function.

8: **chunksize** – `long` *Input*

On entry: if the data function is used, the function inputs no more than **chunksize** data records at a time; otherwise **chunksize** is not referenced.

Constraint: if **dfun** $\neq 0$, **chunksize** ≥ 1.

9: **iwts** – `long` *Input*

On entry: if **iwts** $= -1$, no weights are used; otherwise **iwts** is the index in **data** in which the weights are stored.

Constraints: $-1 \leq$ **iwts** $<$ **nvar**; **iwts** $\neq$ **yvar**; and if **nxvar** > 0, **iwts** $\neq$ **xvar**$[i]$, for $i = 0, 1, \ldots,$ **nxvar** $- 1$.

10: **xbar[nvar]** – `double` *Output*

On exit: the array of variable means; if **iwts** ≥ 0, the value **xbar[iwts]** will not be set.

11: **s[nvar]** – `double` *Output*

On exit: if required, the standard deviations of variables; otherwise **s** must be set to 0. Note that if **iwts**≥ 0, the value **s[iwts]** will not be set.

12: **wsum** – `double *` *Output*

On exit: if **iwts** ≥ 0, **wsum** gives the sum of the weights; otherwise its value equals **nrec**.

13: **nzw** – `long *` *Output*

On exit: the number of data records with positive weights (equals **nrec** if **iwts** $= -1$).

14: **info** – `int *` *Output*

On exit: **info** gives information on the success of the function call:

0: the function successfully completed its task.

i; $i = 1, 2, 3, 4, 6, 8, 9$: the specification of the ith formal parameter was incorrect.

53: there were not at least two data records with positive weight values.

99: the function failed to allocate enough memory.

> 100: an error occurred in a function specified by the user.

Notation

nvar	the number of variables, m.
nrec	the number of data records, n.
iwts	if **iwts** ≥ 0, **iwts** is the index in the data that defines the weights, w_i, for $i = 1, 2, \ldots, n$.
xbar	the sample means, $\bar{x}_j$, for $j = 1, 2, \ldots, m$.
s	the standard deviations, s_j, for $j = 1, 2, \ldots, m$.

Description

For n data records on m variables a one-pass update algorithm (West 1979) is used to compute the means and standard deviations of variables.

Let x_i be the ith data record, for $i = 1, 2, \ldots, n$, which takes a value x_{ij} for the jth variable, for $j = 1, 2, \ldots, m$. The mean value of the jth variable is given by,

$$\bar{x}_j = \frac{\sum_{i=1}^{n} w_i x_{ij}}{\sum_{i=1}^{n} w_i}, \quad j = 1, 2, \ldots, m.$$

The standard deviation of the jth variable is given by,

$$s_j = \frac{\displaystyle\sum_{i=1}^{n} w_i (x_{ij} - \bar{x}_j)(x_{ij} - \bar{x}_j)}{\displaystyle\sum_{i=1}^{n} w_i - 1}, \quad j = 1, 2, \ldots, m.$$

References and Further Reading

West D H D (1979) Updating mean and variance estimates: an improved method *Comm. ACM* **22** (9) 532–535.

See Also

pca_ex.c   an example calling program.