Data Imputation: nagdmc_impute_dist

Purpose

nagdmc_impute_dist imputes data values based on distances to donor data records, for different distance metrics.

nagdmc_impute_dist returns an array containing the indexes of imputed values and donor records (see 'Explanatory Code'). The memory used by this array should be returned to the operating system by the user as indicated in the Essential Introduction.

Declaration

```
#include <nagdmc.h>
long *nagdmc_impute_dist(long rec1, long nvar, long nrec, long dblk, double data[],
                         long ncat[], long maxcat, long cat[], double wts[],
                         double mval, int stype[], long randsel, long ndonors,
                         long matchall, int reuse, double minwt, double R[],
                         int dtype[], double udist[], long *nrepl, long *nfail,
                         long *noimp, 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/Output*

On entry: 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** $- 1$.

On exit: missing values in **data** are replaced by their estimates.

6: **ncat**[**nvar**] – `long` *Input*

On entry: **ncat**[i] contains the number of categories on the ith variable, for $i = 0, 1, \ldots,$ **nvar** $- 1$. If the ith variable is continuous, **ncat**[i] must be set equal to zero. If all variables in the analysis are continuous, **ncat** must be 0.

Constraints: if not 0, **ncat**[i] ≥ 0, for $i = 0, 1, \ldots,$ **nvar** $- 1$.

7: **maxcat** – `long` *Input*

On entry: the maximum number of categories on any categorical variable. If all variables are continuous, **maxcat** must be set equal to zero.

Constraints: if **ncat** is 0, **maxcat** $= 0$; otherwise **maxcat** $\geq$ **ncat**[i], for $i = 0, 1, \ldots,$ **nvar** $- 1$.

8: **cat**[**nvar*maxcat**] – `long` *Input*

On entry: the categories for the categorical variables. The categories for the ith variable are stored in **cat**[$i *$ **maxcat** $+ j$], for $j = 0, 1, \ldots,$ **ncat**[i] $- 1$. If all variables in the analysis are continuous, **cat** must be 0.

Constraint: if **ncat** is 0, **cat** must be 0.

9: **wts[nvar]** – `double` *Input*

On entry: if **wts** is 0, the weight on each variable in the analysis is set to 1.0; otherwise **wts**$[i]$ is the weight on the ith variable used to calculate the distance between donor records and records containing missing values, for $i = 0, 1, \ldots,$ **nvar** $- 1$.

Constraints: if **wts** $\neq 0$, **wts**$[i] \geq 0.0$, for $i = 0, 1, \ldots,$ **nvar** $- 1$; and the sum of elements in **wts** must be greater than 0.0.

10: **mval** – `double` *Input*

On entry: all values in **data** equal within machine precision to **mval** are considered missing from the analysis.

Suggested value: a value outside the interval $[a, b]$, where a and b are the minimum and maximum value in your data, respectively.

11: **stype[3]** – `long` *Input*

On entry: controls the search strategy:

stype$[0] = -1$, only donors with the same value of categorical variable **stype**$[1]$ are searched, and **stype**$[2]$ is not referenced;

stype$[0] = 0$, all donors are searched, and **stype**$[1]$ and **stype**$[2]$ are not referenced;

stype$[0] = 1$, and only donors between record numbers **stype**$[1]$ and **stype**$[2]$ are searched.

Constraints: **stype**$[0] \in \{-1, 0, 1\}$; if **stype**$[0] = -1$, $0 \leq$ **stype**$[1] <$ **nvar**, and **ncat**[**stype**$[1]] > 0$; if **stype**$[0] = 1$, **rec1** $\leq$ **stype**$[1] <$ **stype**$[2] <$ **nrec**.

12: **randsel** – `int` *Input*

On entry: indicates what random selection from donors is to be performed:

if **randsel** $= -1$, the seed for random selection is taken from the system clock;

if **randsel** $= 0$, random selection is not used and the first donor is used;

if **randsel** > 0, **randsel** is used as the repeatable seed to start the random selection.

Constraint: **randsel** ≥ -1.

13: **ndonors** – `long` *Input*

On entry: if **randsel** $\neq 0$, the donor record will be selected at random from up to the first **ndonors** best donors encountered; otherwise **ndonors** is not referenced.

Constraint: **ndonors** ≥ 1.

14: **matchall** – `long` *Input*

On entry: if **matchall** $= -1$, all missing values in a record are imputed; otherwise only values on the **matchall** variable are imputed.

Constraint: $-1 \leq$ **matchall** $<$ **nvar**.

15: **reuse** – `int` *Input*

On entry: if **reuse** $= 1$, the same donor record may be re-used to impute values for different records without penalty; otherwise **reuse** must equal zero and subsequent distances to donor records are penalised by adding **nvar** or, if **wts** $\neq 0$, the sum of elements in **wts**.

Constraint: **reuse** $\in \{0, 1\}$.

16: **minwt** – `double` *Input*

On entry: values equal to **mval** in a target record are imputed if the sum of weights on variables with values not equal to **mval** in the donor record is greater than or equal to **minwt**; othwerwise data values are not imputed and the value of **nfail** is increased by one.

Constraint: **minwt** ≥ 0.0.

17: **R[nvar]** – `double` *Input*

On entry: **R**$[i]$ contains the scaling factor for continuous variable i, for $i = 0, 1, \ldots,$ **nvar** $- 1$. For Euclidean and Manhattan distances **R**$[i]$ is a multiplicative standardisation. For threshold distances, it is the threshold value and for regression distances it is the regression coefficient. If **R** is 0, the scaling factor for continuous variable i is taken to be 1.0. If all variables are cateogrical, **R** is not referenced and should be 0.

Constraints: if **R** is not 0, **R**$[i] > 0.0$, for a continuous variable i, for $i = 0, 1, \ldots,$ **nvar** $- 1$.

18: **dtype**[**nvar**] – `char` *Input*

On entry: **dtype**[i] signifies the distance measure used for the ith variable, for $i = 0, 1, \ldots,$ **nvar** $- 1$. The available measures depend on the type of variable: categorical or continuous. The parameter **ncat** is used to determine categorical and continuous variables.

For continuous variables the choices for **dtype**[i] are:

dtype[i] $= 0$, for Euclidean distance;
dtype[i] $= 1$, for Manhattan distance;
dtype[i] $= 2$, for regression distance;
dtype[i] $= 3$, for threshold distance.

For categorical variables the choices for **dtype**[i] are:

dtype[i] $= 0$, for simple matching;
dtype[i] $= 1$, for rank difference distances;
dtype[i] $= 2$, for distances based on user-supplied metric.

Constraints: if **ncat** $= 0$ or **ncat**[i] $= 0$, **dtype**[i] $\in \{0, 1, 2, 3\}$, for $i = 0, 1, \ldots,$ **nvar** $- 1$; if **ncat**[i] > 0, **dtype**[i] $\in \{0, 1, 2\}$, for $i = 0, 1, \ldots,$ **nvar** $- 1$.

19: **udist**[**nvar**$*$**maxcat**$*$**maxcat**] – `double` *Input*

On entry: the distance tables for user-supplied distances. The **nvar** **ncat**[i] by **ncat**[i] tables are stored in **nvar** **maxcat** $*$ **maxcat** blocks. If the user-supplied distance option for categorical variables is not selected for any variable in the analysis, **udist** is not referenced.

20: **nrepl** – `long *` *Output*

On exit: the number of **mval** values replaced by the function. The value of this parameter is determines the length of the array returned by the function, see the 'Explanatory Code'.

21: **nfail** – `long *` *Output*

On exit: the number of data records requiring imputation but for which the sum of variable weights for non-missing values in the donor record is less than **minwt**.

22: **noimp** – `long *` *Output*

On exit: the number of data records requiring imputation for which no donor records are available.

23: **info** – `int *` *Output*

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

0: the function successfully completed its task.

i; $i = 1, 2, 3, 4, 6, 7, 8, 9, 11, 12, \ldots, 18$: the specification of the ith formal parameter was incorrect.

20: no missing values were found in the data; check your definition of **mval**.

50: if **stype**[0] $= -1$, a category for the **stype**[1] variable is incorrect.

55: a value for a categorical variable is incorrectly specified. Check values in the **ncat**, **cat** and **data** arrays.

99: the function failed to allocate enough memory.

Notation

nrec the number of data records, n.
data the data set X.
nxvar determines the number of variables, m.
mval the value of missing data values, z.
wts the weights w_j, for $j = 1, 2, \ldots, m$.
ndonors if random selection is used, the value t.
minwt the minimum value for sum of weights on variables, v.
R the scaling factors R_j, for $j = 1, 2, \ldots, m$.

Description

Let X be a set of n data records x_i, for $i = 1, 2, \ldots, n$. The jth variable of the ith record x_i takes either the value x_{ij} or a dummy value, z, representing values in X missing at random, for $j = 1, 2, \ldots, m$.

Given a target record x_k in X containing one or more variables with value z, a distance s is computed to each of the data records x_i, for $i = 1, 2, \ldots, n$ and $i \neq k$. A low value of s indicates that the record x_i is a close match to the target record x_k. Furthermore, let d_j be the difference between the value of the jth variable for the ith data record and the target record: $x_{ij} - x_{kj}$.

For target record k and the ith data record, each of the $p \leq m$ continuous variables in X contributes to s by evaluating:

$$y = w_j f(R_j, d_j),$$

where:

(a) $w_j \in \mathbb{R}$ is the weight given to the jth variable;
(b) $R_j \in \mathbb{R}$ is an optional scaling factor;
(c) $f(\cdot) \in \mathbb{R} \rightarrow \mathbb{R}$ is continuous distance function that takes one of the forms:
    (i) Euclidean distance, $f(R_j, d_j) = (R_j d_j)^2$;
    (ii) Manhattan distance, $f(R_j, d_j) = R_j |d_j|$;
    (iii) regression distance, $f(R_j, d_j) = R_j d_j$;
    (iv) threshold distance, if $d_j > R_j$, $f(R_j, d_j) = 1$; otherwise $f(R_j, d_j) = 0$.

An appropriate scaling factor R_j for continuous variables is the reciprocal of the range of the jth variable. Distances defined on continuous variables contribute to the value of s by adding:

(a) the square root of the sum of squared values of y over all continuous variables that use the Euclidean distance;
(b) the sum of absolute values of y over all continuous variables that use the Manhattan distance;
(c) the absolute value of the sum of y over all continuous variables that use the regression distance;
(d) the sum of y over all continuous variables that use the threshold distance.

For target record k and the ith data record, each of the $q = m - p$ categorical variables in X adds to s a value:

$$\sum_{j=1}^{q} w_j g(d_j),$$

where the categorical distance function $g(\cdot)$ takes one of the forms:

(a) simple matching, if $d_j = 0$, $g(d_j) = 1$; otherwise $g(d_j) = 0$;
(b) scaled rank difference for a variable with c_j categories, $g(d_j) = |d_j|/(c_j - 1)$;
(c) user-supplied distance from tables for the jth variable between category values x_{ij} and x_{kj}.

In the case of a user-supplied distance for a variable with r categories, an r by r matrix (usually symmetric) containing the distances for each possible pair of values has to be supplied.

The donor record is the data record used to replace missing values in the target record. Donor records can be selected at random or by using the data record with the lowest distance score. If random selection is not used, missing values on the jth variable in the target record are replaced by the jth values in the data record with the lowest score, s; otherwise a donor record is chosen at random from the t data records with the lowest scores.

Optionally, penalised distances to donor records can be used, in which case the value:

$$h * \sum_{j=1}^{m} w_j$$

is added to distances from a target record to a record which has been previously chosen h times as a donor record.

The t closest records to x_k in X can be searched in one of three ways:

(a) search all records x_i in X, $i \neq k$, for $i = 1, 2, \ldots, n$;
(b) search a subset of X defined by record numbers in the range $[a, b]$;
(c) search all records with a particular value u of a categorical variable l.

The variable weights can be used to select and give importance to variables when matching records with missing values to donor records. If the weight of the jth variable, w_j, is set to zero, no matching for imputation is carried out for that variable.

Missing values z will not be imputed if the sum of weights on variables without missing values for the donor record is below a minimum value, v.

References and Further Reading

None.

Explanatory Code

The following C function prints the index in the data and the imputed value of each of the **nrepl** replaced values after a successful call to **nagdmc_impute_dist** returning `indexes`.

```
#include <stdio.h>

void imputed_values(long nvar, double data[], long nrepl, long indexes[]) {
    long i, j, k, l;

#define MISSING_ROW(I) indexes[I]
#define MISSING_COL(I) indexes[I+nrepl]
#define DONOR_REC(I) indexes[I+2*nrepl]
#define DATA(I,J) data[(I)*nvar+J]

    printf("\n\tRow \tCol \tValue \tDonor Record\n");
    for (i=0; i<nrepl; ++i) {
        j = MISSING_ROW(i);
        k = MISSING_COL(i);
        l = DONOR_REC(i);
        printf("\t%-4li\t%-4li\t%-6.3f\t%-li\n",j,k,DATA(j,k),l);
    }
}
```

See Also

nagdmc_free_impute returns memory allocated by **nagdmc_impute_dist** to the operating system.
impute_dist_ex.c the example calling program.