nag_mv_z_scores (g03zac)

1. Purpose

nag_mv_z_scores (g03zac) produces standardized values (z-scores) for a data matrix.

2. Specification

```
#include <nag.h>
#include <nagg03.h>

void nag_mv_z_scores(Integer n, Integer m, double x[], Integer tdx,
         Integer nvar, Integer isx[], double s[], double e[],
         double z[], Integer tdz, NagError *fail)
```

3. Description

For a data matrix, X, consisting of n observations on p variables, with elements x_{ij}, nag_mv_z_scores (g03zac) computes a matrix, Z, with elements z_{ij} such that:

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}, \quad i = 1, 2, \ldots, n; \;\; j = 1, 2, \ldots, p,$$

where μ_j is a location shift and σ_j is a scaling factor. Typically, μ_j will be the mean and σ_j will be the standard deviation of the jth variable and therefore the elements in column j of Z will have zero mean and unit variance.

4. Parameters

n
Input: the number of observations in the data matrix, n.
Constraint: $\mathbf{n} \geq 1$.

m
Input: the number of variables in the data array **x**.
Constraint: $\mathbf{m} \geq$ **nvar**.

x[n][tdx]
Output: $\mathbf{x}[i-1][j-1]$ must contain the ith sample point for the jth variable x_{ij}, for $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots,$**m**.

tdx
Input: the last dimension of the array **x** as declared in the calling program.
Constraint: $\mathbf{tdx} \geq \mathbf{m}$.

nvar
Input: the number of variables to be standardised, p.
Constraint: $\mathbf{nvar} \geq 1$.

isx[m]
Output: $\mathbf{isx}[j-1]$ indicates whether or not the observations on the jth variable are included in the matrix of standardized values.
If $\mathbf{isx}[j-1] \neq 0$, then the observations from the jth variable are included.
If $\mathbf{isx}[j-1] = 0$, then the observations from the jth variable are not included.
Constraint: $\mathbf{isx}[j-1] \neq 0$ for **nvar** values of j.

s[m]
Input: if $\mathbf{isx}[j-1] \neq 0$, then $\mathbf{s}[j-1]$ must contain the scaling (standard deviation), σ_j, for the jth variable.
If $\mathbf{isx}[j-1] = 0$, then $\mathbf{s}[j-1]$ is not referenced.
Constraint: if $\mathbf{isx}[j-1] \neq 0$, then $\mathbf{s}[j-1] > 0.0$ for $j = 1, 2, \ldots,$**m**.

e[m]
Input: if **isx**$[j-1] \neq 0$, then **e**$[j-1]$ must contain the location shift (mean), μ_j, for the jth variable.
If **isx**$[j-1] = 0$, then **e**$[j-1]$ is not referenced.

z[n][tdz]
Output: the matrix of standardized values (z-scores), Z.

tdz
Input: the last dimension of the array **z** as declared in the calling program.
Constraint: **tdz** $\geq$ **nvar**.

fail
The NAG error parameter, see the Essential Introduction to the NAG C Library.

5. Error Indications and Warnings

NE_INT_ARG_LT
On entry, **n** must not be less than 1: **n** = ⟨*value*⟩.
On entry, **nvar** must not be less than 1: **nvar** = ⟨*value*⟩.

NE_2_INT_ARG_LT
On entry, **m** = ⟨*value*⟩ while **nvar** = ⟨*value*⟩.
These parameters must satisfy **m** $\geq$ **nvar**.
On entry, **tdx** = ⟨*value*⟩ while **m** = ⟨*value*⟩.
These parameters must satisfy **tdx** $\geq$ **m**.
On entry, **tdz** = ⟨*value*⟩ while **nvar** = ⟨*value*⟩.
These parameters must satisfy **tdz** $\geq$ **nvar**.

NE_VAR_INCL_INDICATED
The number of variables, **nvar** in the analysis = ⟨*value*⟩, while number of variables included in the analysis via array **isx** = ⟨*value*⟩.
Constraint: these two numbers must be the same.

NE_INTARR_REALARR
On entry, **isx**[⟨*value*⟩] = ⟨*value*⟩, **s**[⟨*value*⟩] = ⟨*value*⟩.
Constraint: if **isx**$[j-1] = 0$ then **s**$[j-1] > 0.0$, j=1,2,...,m.

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

Means and standard deviations may be obtained using nag_summary_stats_1var (g01aac) or nag_corr_cov (g02bxc).

6.1. Accuracy

Standard accuracy is achieved.

6.2. References

None.

7. See Also

nag_summary_stats_1var (g01aac)
nag_corr_cov (g02bxc)

8. Example

A 4 by 3 data matrix is input along with location and scaling values. The first and third columns are scaled and the results printed.

8.1. Program Text

```
/* nag_mv_z_scores (g03zac) Example Program.
 *
 * Copyright 1998 Numerical Algorithms Group.
 *
 * Mark 5, 1998.
 *
 */
#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagg03.h>

#define NMAX 4
#define MMAX 3

main()
{
  double e[MMAX], s[MMAX], x[NMAX][MMAX], z[NMAX][MMAX];

  Integer nvar;
  Integer   isx[MMAX];
  Integer i, j, m, n;
  Integer tdx=MMAX, tdz=MMAX;

  Vprintf("g03zac Example Program Results\n\n");

  /*  Skip headings in data file */
  Vscanf("%*[^\n]");
  Vscanf("%ld",&n);
  Vscanf("%ld",&m);
  Vscanf("%ld",&nvar);

  if (m <= MMAX && n <= NMAX)
    {
      for (i = 0; i < n; ++i)
        {
          for (j = 0; j < m; ++j)
            Vscanf("%lf",&x[i][j]);
        }
      for (j = 0; j < m; ++j)
        Vscanf("%ld",&isx[j]);

      for (j = 0; j < m; ++j)
        Vscanf("%lf",&e[j]);

      for (j = 0; j < m; ++j)
        Vscanf("%lf",&s[j]);

      g03zac(n, m, (double *)x, tdx, nvar, isx, s, e, (double *)z, tdz, NAGERR_DEFAULT);

      Vprintf("\nStandardized Values\n\n");
      for (i = 0; i < n; ++i)
        {
          for (j = 0; j < nvar; ++j)
            Vprintf("%8.3f",z[i][j]);
          Vprintf("\n");
        }
      exit(EXIT_SUCCESS);
    }
  else
    {
      Vprintf("Incorrect input value of n or m.\n");
      exit(EXIT_FAILURE);
    }

}
```

8.2. Program Data

```
g03zac Example Program Data
4 3 2
15.0 0.0 1500.0
12.0 1.0 1000.0
18.0 2.0 1200.0
14.0 3.0  500.0
 1      0    1
14.75 0.0 1050.0
 2.50 0.0  420.3
```

8.3. Program Results

```
g03zac Example Program Results


Standardized Values

   0.100   1.071
  -1.100  -0.119
   1.300   0.357
  -0.300  -1.309
```