nag_ran_sample_vec (g05ejc)

1. Purpose

nag_ran_sample_vec (g05ejc) selects a pseudo-random sample without replacement from an integer vector.

2. Specification

```
#include <nag.h>
#include <nagg05.h>

void nag_ran_sample_vec(Integer ia[], Integer n, Integer iz[],
                        Integer m, NagError *fail)
```

3. Description

The function peforms a single pseudo-random selection of m elements from vector **ia** of length n and then places them in vector **iz**. Their order in **ia** will be preserved in **iz**. Each of the $\binom{n}{m}$ possible combinations of elements of **ia** may be regarded as being equiprobable.

4. Parameters

ia[n]
Input: the population to be sampled.

n
Input: the number of elements in the vector to be sampled.
Constraint: $\mathbf{n} \geq 1$.

iz[m]
Output: the selected sample.

m
Input: the sample size.
Constraint: $1 \leq \mathbf{m} \leq \mathbf{n}$.

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: $\mathbf{n} = \langle value \rangle$.
On entry, **m** must not be less than 1: $\mathbf{m} = \langle value \rangle$.

NE_2_INT_ARG_GT
On entry, $\mathbf{m} = \langle value \rangle$ while $\mathbf{n} = \langle value \rangle$.
These parameters must satisfy $\mathbf{m} \leq \mathbf{n}$.

6. Further Comments

If n is greater than 60 it is theoretically impossible to generate all $\binom{n}{m}$ combinations unless m is near 1 or near n. This is because the number of possible combinations exceeds the cycle length of the internal random number generator.

The time taken by the function is of order n.

In order to sample other kinds of objects (i.e., vectors, or matrices of higher dimensions), the following technique may be used:

(a) Set **ia**$[i-1] = i$, for $i = 1, 2, \ldots, n$ (where n is the number of objects)

(b) Use nag_ran_sample_vec to take a sample from **ia** and put it into **iz**

(c) Use the contents of **iz** as a set of indices to access the relevant object.

In order to divide a population into several groups, nag_ran_permut_vec (g05ehc) is more efficient.

6.1. Accuracy

Not applicable.

6.2. References

Kendall M G and Stuart A (1969) *The Advanced Theory of Statistics (Vol 2).* (3rd Edn) Griffin, London.

Knuth D E (1981) *The Art of Computer Programming (Vol 2).* (2nd Edn) Addison-Wesley.

7. See Also

nag_ran_permut_vec (g05ehc)

8. Example

From a vector containing 0 and the first 7 positive integers in ascending order, random samples of size 1,2...,8 are selected and printed.

8.1. Program Text

```
/* nag_ran_sample_vec(g05ejc) Example Program
 *
 * Copyright 1994 Numerical Algorithms Group.
 *
 * Mark 3, 1994.
 */

#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagg05.h>

#define NMAX 8

main()
{
  Integer i, n, m, k;
  Integer ia[NMAX], ib[NMAX];
  Integer seed = 0;

  Vprintf("g05ejc Example Program Results\n");
  g05cbc(seed);
  n = NMAX;

  for (i = 0; i < n; ++i)
    ia[i] = i;

  Vprintf ("\nSamples from the first %ld integers \n\n", n);
  Vprintf("Sample size            Values \n");

  for (m = 1; m <= n; ++m)
    {
      g05ejc(ia, n, ib, m, NAGERR_DEFAULT);
      Vprintf("     %ld                    ", m);
      for (k = 0; k < m; ++k)
        Vprintf("%ld  ",ib[k]);
      Vprintf("\n");
    }
  exit(EXIT_SUCCESS);
}
```

8.2. Program Data

None.

8.3. Program Results

```
g05ejc Example Program Results

Samples from the first 8 integers

Sample size            Values
     1                   5
     2                   0  6
     3                   0  2  3
     4                   0  1  5  7
     5                   0  2  3  5  6
     6                   0  1  2  3  4  5
     7                   0  1  2  3  5  6  7
     8                   0  1  2  3  4  5  6  7
```