nag_ref_vec_binomial (g05edc)

1. Purpose

nag_ref_vec_binomial (g05edc) sets up the reference vector **r** for a binomial distribution of the number of successes in n trials, each with probability of success p.

2. Specification

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

void nag_ref_vec_binomial(Integer n, double p, double **r, NagError *fail)
```

3. Description

nag_ref_vec_binomial sets up a reference vector for use in nag_return_discrete (g05eyc). Together these routines produce random numbers from the binomial distribution defined by:

$$\begin{array}{lcll} P(I=i) & = & \frac{n!}{i!(n-i)!}p^i(1-p)^{n-i} & \text{if } i=0,\ldots,n, \\ P(I=i) & = & 0 & \text{otherwise} \end{array}.$$

The reference array is found by a recurrence relation if $np(1-p) < 50$; otherwise Stirling's approximation is used.

4. Parameters

n

Input: the number of trials, n, of the distribution.
Constraint: $\mathbf{n} \geq 0$.

p

Input: the probability of success, p, of the distribution.
Constraint: $0.0 \leq \mathbf{p} \leq 1.0$.

r

Output: reference vector for which memory will be allocated internally. If no memory is allocated to **r** (e.g. when an input error is detected) then **r** will be NULL on return, otherwise the user should use the NAG macro **NAG_FREE** to free the storage allocated by **r** when it is no longer of use.

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 0: $\mathbf{n} = \langle value \rangle$.

NE_REAL_ARG_LT

On entry, **p** must not be less than 0.0: $\mathbf{p} = \langle value \rangle$.

NE_REAL_ARG_GT

On entry, **p** must not be greater than 1.0: $\mathbf{p} = \langle value \rangle$.

NE_ALLOC_FAIL

Memory allocation failed.

6. Further Comments

6.1. Accuracy

Not applicable.

6.2. References

Kendall M G and Stuart A (1969) *The Advanced Theory of Statistics (Vol 1)* (3rd Edn) Griffin.
Knuth D E (1981) *The Art of Computer Programming (Vol 2)* (2nd Edn) Addison-Wesley.

7. See Also

nag_random_init_repeatable (g05cbc)
nag_random_init_nonrepeatable (g05ccc)
nag_random_normal (g05ddc)
nag_ref_vec_poisson (g05ecc)
nag_return_discrete (g05eyc)

8. Example

The example program sets up a reference vector for a binomial distribution with $n = 100$ and $p = 0.5$; it then prints the first five pseudo-random numbers generated by nag_return_discrete (g05eyc), after initialisation by nag_random_init_repeatable (g05cbc).

8.1. Program Text

```
/* nag_ref_vec_binomial(g05edc) Example Program
 *
 * Copyright 1991 Numerical Algorithms Group.
 *
 * Mark 2, 1991.
 *
 * Mark 3 revised, 1994.
 */

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

#define N 100
#define P 0.5

main()
{
  Integer i, x;
  double *r;

  Vprintf("g05edc Example Program Results\n");
  g05cbc((Integer)0);
  g05edc((Integer)N, (double)P, &r, NAGERR_DEFAULT);
  for (i=1; i<=5; i++)
    {
      x = g05eyc(r);
      Vprintf("%5ld\n", x);
    }
  NAG_FREE(r);
  exit(EXIT_SUCCESS);
}
```

8.2. Program Data

None.

8.3. Program Results

```
g05edc Example Program Results
   54
   46
   48
   46
   56
```