

nag_hypergeom_dist (g01blc)

1. Purpose

nag_hypergeom_dist (g01blc) returns the lower tail, upper tail and point probabilities associated with a hypergeometric distribution.

2. Specification

```
#include <nag.h>
#include <nagg01.h>

void nag_hypergeom_dist(Integer n, Integer l, Integer m, Integer k,
                        double *plek, double *pgtk, double *peqk, NagError *fail)
```

3. Description

Let X denote a random variable having a hypergeometric distribution with parameters n , l and m ($n \geq l \geq 0$, $n \geq m \geq 0$). Then

$$\text{Prob}\{X = k\} = \frac{\binom{m}{k} \binom{n-m}{l-k}}{\binom{n}{l}},$$

where $\max(0, l - (n - m)) \leq k \leq \min(l, m)$, $0 \leq l \leq n$ and $0 \leq m \leq n$.

The hypergeometric distribution may arise if in a population of size n a number m are marked. From this population a sample of size l is drawn and of these k are observed to be marked.

The mean of the distribution = $\frac{lm}{n}$, and the variance = $\frac{lm(n-l)(n-m)}{n^2(n-1)}$.

This routine computes for given n , l , m and k the probabilities:

```
plek = Prob{ $X \leq k$ }
pgtk = Prob{ $X > k$ }
peqk = Prob{ $X = k$ }.
```

The method is similar to the method for the Poisson distribution described in Knüsel (1986).

4. Parameters

n

Input: the parameter n of the hypergeometric distribution.
Constraint: **n** ≥ 0 .

l

Input: the parameter l of the hypergeometric distribution.
Constraint: $0 \leq \mathbf{l} \leq \mathbf{n}$.

m

Input: the parameter m of the hypergeometric distribution.
Constraint: $0 \leq \mathbf{m} \leq \mathbf{n}$.

k

Input: the integer k which defines the required probabilities.
Constraint: $\max(0, \mathbf{l} - (\mathbf{n} - \mathbf{m})) \leq \mathbf{k} \leq \min(\mathbf{l}, \mathbf{m})$

plek

Output: the lower tail probability, Prob{ $X \leq k$ }.

pgtk

Output: the upper tail probability, Prob{ $X > k$ }.

peqk

Output: the point probability, $\text{Prob}\{X = k\}$.

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: **n** = *<value>*.

On entry, **l** must not be less than 0: **l** = *<value>*.

On entry, **k** must not be less than 0: **k** = *<value>*.

On entry, **m** must not be less than 0: **m** = *<value>*.

NE_2_INT_ARG_GT

On entry, **l** = *<value>* while **n** = *<value>*. These parameters must satisfy $l \leq n$.

On entry, **m** = *<value>* while **n** = *<value>*. These parameters must satisfy $m \leq n$.

On entry, **k** = *<value>* while **l** = *<value>*. These parameters must satisfy $k \leq l$.

On entry, **k** = *<value>* while **m** = *<value>*. These parameters must satisfy $k \leq m$.

NE_4_INT_ARG_CONS

On entry, **k** = *<value>*, **l** = *<value>*, **m** = *<value>*, **n** = *<value>*. These parameters must satisfy $k \geq l + m - n$.

NE_ARG_TOO_LARGE

On entry, **n** is too large to be represented exactly as a double precision number.

NE_VARIANCE_TOO_LARGE

On entry, the variance = $\frac{lm(n-l)(n-m)}{n^2(n-1)}$ exceeds 10^6 .

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

The time taken by the routine depends on the variance (see Section 3) and on k . For given variance, the time is greatest when $k \approx lm/n$ (= the mean), and is then approximately proportional to the square-root of the variance.

6.1. Accuracy

Results are correct to a relative accuracy of at least 10^{-6} on machines with a precision of 9 or more decimal digits, and to a relative accuracy of at least 10^{-3} on machines of lower precision (provided that the results do not underflow to zero).

6.2. References

Knüsel L (1986) Computation of the Chi-square and Poisson Distribution. *SIAM J. Sci. Statist. Comput.* **7** 1022–1036.

7. See Also

nag_binomial_dist (g01bjc)

nag_poisson_dist (g01bkc)

8. Example

This example program reads values of n , l , m and k from a data file until end-of-file is reached, and prints the corresponding probabilities.

8.1. Program Text

```

/* nag_hypergeom_dist(g01blc) Example Program.
 *
 * Copyright 1996 Numerical Algorithms Group.
 *
 * Mark 4, 1996.
 *
 */

#include <nag.h>
#include <nag_stdlib.h>
#include <stdio.h>
#include <nagg01.h>

main()
{
    double plek, peqk, pgtk;

    Integer k, l, m, n;

    Vprintf("g01blc Example Program Results\n");

    /*      Skip heading in data file */
    Vscanf("%*[^\\n] ");

    Vprintf("\n      n      l      m      k      plek      pgtk      peqk\n\n");

    while((scanf("%ld %ld %ld %ld%*[^\\n]", &n, &l, &m, &k)) != EOF)
    {
        g01blc(n, l, m, k, &plek, &pgtk, &peqk, NAGERR_DEFAULT);
        Vprintf(" %4ld%4ld%4ld%4ld%10.5f%10.5f%10.5f\n",
            n, l, m, k, plek, pgtk, peqk);
    }
    exit(EXIT_SUCCESS);
}

```

8.2. Program Data

```

g01blc Example Program Data
10  2  5  1      : n, l, m, k
40 10  3  2
155 35 122 22
1000 444 500 220

```

8.3. Program Results

g01blc Example Program Results

n	l	m	k	plek	pgtk	peqk
10	2	5	1	0.77778	0.22222	0.55556
40	10	3	2	0.98785	0.01215	0.13664
155	35	122	22	0.01101	0.98899	0.00779
1000	444	500	220	0.42429	0.57571	0.04913
