NAG Library Function Document

nag_deviates_chi_sq_vector (g01tcc)

1 Purpose

nag_deviates_chi_sq_vector (g01tcc) returns a number of deviates associated with the given probabilities of the χ^2-distribution with real degrees of freedom.

2 Specification

```
#include <nag.h>
#include <nagg01.h>
void nag_deviates_chi_sq_vector (Integer ltail,
     const Nag_TailProbability tail[], Integer lp, const double p[],
     Integer ldf, const double df[], double x[], Integer ivalid[],
     NagError *fail)
```

3 Description

The deviate, x_{p_i}, associated with the lower tail probability p_i of the χ^2-distribution with ν_i degrees of freedom is defined as the solution to

$$P\left(X_i \leq x_{p_i} : \nu_i\right) = p_i = \frac{1}{2^{\nu_i/2}\Gamma(\nu_i/2)}\int_0^{x_{p_i}} e^{-X_i/2} X_i^{\nu_i/2-1}\, dX_i, \quad 0 \leq x_{p_i} < \infty;\ \nu_i > 0.$$

The required x_{p_i} is found by using the relationship between a χ^2-distribution and a gamma distribution, i.e., a χ^2-distribution with ν_i degrees of freedom is equal to a gamma distribution with scale parameter 2 and shape parameter $\nu_i/2$.

For very large values of ν_i, greater than 10^5, Wilson and Hilferty's Normal approximation to the χ^2 is used; see Kendall and Stuart (1969).

The input arrays to this function are designed to allow maximum flexibility in the supply of vector arguments by re-using elements of any arrays that are shorter than the total number of evaluations required. See Section 2.6 in the g01 Chapter Introduction for further information.

4 References

Best D J and Roberts D E (1975) Algorithm AS 91. The percentage points of the χ^2 distribution *Appl. Statist.* **24** 385–388

Hastings N A J and Peacock J B (1975) *Statistical Distributions* Butterworth

Kendall M G and Stuart A (1969) *The Advanced Theory of Statistics (Volume 1)* (3rd Edition) Griffin

5 Arguments

1: **ltail** – Integer *Input*

On entry: the length of the array **tail**.

Constraint: **ltail** > 0.

2: **tail**[**ltail**] – const Nag_TailProbability *Input*

On entry: indicates which tail the supplied probabilities represent. For $j = (i-1) \bmod \mathbf{ltail}$, for $i = 1, 2, \ldots, \max(\mathbf{ltail}, \mathbf{lp}, \mathbf{ldf})$:

tail$[j]$ = Nag_LowerTail
The lower tail probability, i.e., $p_i = P\left(X_i \leq x_{p_i} : \nu_i\right)$.

tail$[j]$ = Nag_UpperTail
The upper tail probability, i.e., $p_i = P\left(X_i \geq x_{p_i} : \nu_i\right)$.

Constraint: **tail**$[j-1]$ = Nag_LowerTail or Nag_UpperTail, for $j = 1, 2, \ldots,$ **ltail**.

3: **lp** – Integer *Input*

On entry: the length of the array **p**.

Constraint: **lp** > 0.

4: **p**[**lp**] – const double *Input*

On entry: p_i, the probability of the required χ^2-distribution as defined by **tail** with $p_i =$ **p**$[j]$, $j = (i-1) \bmod$ **lp**.

Constraints:

if **tail**$[k]$ = Nag_LowerTail, $0.0 \leq$ **p**$[j] < 1.0$;
otherwise $0.0 <$ **p**$[j] \leq 1.0$.

Where $k = (i-1) \bmod$ **ltail** and $j = (i-1) \bmod$ **lp**.

5: **ldf** – Integer *Input*

On entry: the length of the array **df**.

Constraint: **ldf** > 0.

6: **df**[**ldf**] – const double *Input*

On entry: ν_i, the degrees of freedom of the χ^2-distribution with $\nu_i =$ **df**$[j]$, $j = (i-1) \bmod$ **ldf**.

Constraint: **df**$[j-1] > 0.0$, for $j = 1, 2, \ldots,$ **ldf**.

7: **x**[dim] – double *Output*

Note: the dimension, *dim*, of the array **x** must be at least $\max(\mathbf{ltail}, \mathbf{lp}, \mathbf{ldf})$.

On exit: x_{p_i}, the deviates for the χ^2-distribution.

8: **ivalid**[dim] – Integer *Output*

Note: the dimension, *dim*, of the array **ivalid** must be at least $\max(\mathbf{ltail}, \mathbf{lp}, \mathbf{ldf})$.

On exit: **ivalid**$[i-1]$ indicates any errors with the input arguments, with

ivalid$[i-1] = 0$
No error.

ivalid$[i-1] = 1$
On entry, invalid value supplied in **tail** when calculating x_{p_i}.

ivalid$[i-1] = 2$
On entry, invalid value for p_i.

ivalid$[i-1] = 3$
On entry, $\nu_i \leq 0.0$.

ivalid$[i-1] = 4$
p_i is too close to 0.0 or 1.0 for the result to be calculated.

ivalid$[i-1] = 5$
The solution has failed to converge. The result should be a reasonable approximation.

9: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.

NE_ARRAY_SIZE

On entry, array size $= \langle value \rangle$.
Constraint: **ldf** > 0.

On entry, array size $= \langle value \rangle$.
Constraint: **lp** > 0.

On entry, array size $= \langle value \rangle$.
Constraint: **ltail** > 0.

NE_BAD_PARAM

On entry, argument $\langle value \rangle$ had an illegal value.

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 contact NAG for assistance.

NW_IVALID

On entry, at least one value of **tail**, **p** or **df** was invalid, or the solution failed to converge. Check **ivalid** for more information.

7 Accuracy

The results should be accurate to five significant digits for most argument values. Some accuracy is lost for p_i close to 0.0 or 1.0.

8 Parallelism and Performance

Not applicable.

9 Further Comments

For higher accuracy the relationship described in Section 3 may be used and a direct call to nag_deviates_gamma_vector (g01tfc) made.

10 Example

This example reads lower tail probabilities for several χ^2-distributions, and calculates and prints the corresponding deviates.

10.1 Program Text

```
/* nag_deviates_chi_sq_vector (g01tcc) Example Program.
 *
 * Copyright 2011, Numerical Algorithms Group.
 *
 * Mark 23, 2011.
 */
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg01.h>

int main(void)
{
  /* Integer scalar and array declarations */
  Integer ltail, lp, ldf, i, lout;
  Integer *ivalid = 0;
  Integer exit_status = 0;

  /* NAG structures */
  NagError fail;
  Nag_TailProbability *tail = 0;

  /* Double scalar and array declarations */
  double *p = 0, *df = 0, *x = 0;

  /* Character scalar and array declarations */
  char ctail[40];

  /* Initialise the error structure to print out any error messages */
  INIT_FAIL(fail);

  printf("nag_deviates_chi_sq_vector (g01tcc) Example Program Results\n\n");

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

  /* Read in the input vectors */
  scanf("%ld%*[^\n] ", &ltail);
  if (!(tail = NAG_ALLOC(ltail, Nag_TailProbability))) {
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
  }
  for (i = 0; i < ltail; i++) {
    scanf("%39s", ctail);
    tail[i] = (Nag_TailProbability) nag_enum_name_to_value(ctail);
  }
  scanf("%*[^\n] ");
  scanf("%ld%*[^\n] ", &lp);
  if (!(p = NAG_ALLOC(lp, double))) {
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
  }
  for (i = 0; i < lp; i++)
    scanf("%lf", &p[i]);
  scanf("%*[^\n] ");
  scanf("%ld%*[^\n] ", &ldf);
  if (!(df = NAG_ALLOC(ldf, double))) {
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
  }
  for (i = 0; i < ldf; i++)
    scanf("%lf", &df[i]);
  scanf("%*[^\n] ");


  /* Allocate memory for output */
```

```
  lout = MAX(ltail,MAX(lp,ldf));
  if (!(x = NAG_ALLOC(lout, double)) ||
      !(ivalid = NAG_ALLOC(lout, Integer))) {
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
  }

  /* Calculate probability */
  nag_deviates_chi_sq_vector(ltail, tail, lp, p, ldf, df, x, ivalid, &fail);
  if (fail.code != NE_NOERROR) {
    printf("Error from nag_deviates_chi_sq_vector (g01tcc).\n%s\n",
            fail.message);
    exit_status = 1;
    if (fail.code != NW_IVALID) goto END;
  }

  /* Display title */
  printf("        tail          p         df        x     ivalid\n");
  printf(" ------------------------------------------------------\n");

  /* Display results */
  for (i = 0; i < lout; i++)
    printf(" %15s   %6.3f    %6.1f   %7.4f    %3ld\n",
           nag_enum_value_to_name(tail[i%ltail]), p[i%lp], df[i%ldf],
           x[i], ivalid[i]);

 END:
  NAG_FREE(tail);
  NAG_FREE(p);
  NAG_FREE(df);
  NAG_FREE(x);
  NAG_FREE(ivalid);

  return(exit_status);
}
```

10.2 Program Data

```
nag_deviates_chi_sq_vector (g01tcc) Example Program Data
1                                                                   :: ltail
Nag_LowerTail                                                       :: tail
3                                                                   :: lp
0.010 0.428 0.869                                                   :: p
3                                                                   :: ldf
20.0 7.5 45.0                                                       :: df
```

10.3 Program Results

```
nag_deviates_chi_sq_vector (g01tcc) Example Program Results

        tail          p         df        x     ivalid
 ------------------------------------------------------
   Nag_LowerTail    0.010      20.0    8.2604      0
   Nag_LowerTail    0.428       7.5    6.2006      0
   Nag_LowerTail    0.869      45.0   55.7381      0
```