

NAG Library Routine Document

G01TCF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

1 Purpose

G01TCF returns a number of deviates associated with the given probabilities of the χ^2 -distribution with real degrees of freedom.

2 Specification

```
SUBROUTINE G01TCF (LTAIL, TAIL, LP, P, LDF, DF, X, IVALID, IFAIL)
  INTEGER          LTAIL, LP, LDF, IVALID(*), IFAIL
  REAL (KIND=nag_wp) P(LP), DF(LDF), X(*)
  CHARACTER(1)    TAIL(LTAIL)
```

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

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

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 routine are designed to allow maximum flexibility in the supply of vector parameters 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 Parameters

1: LTAIL – INTEGER *Input*
On entry: the length of the array TAIL.
Constraint: LTAIL > 0.

- 2: TAIL(LTAIL) – CHARACTER(1) array *Input*
On entry: indicates which tail the supplied probabilities represent. For $j = ((i - 1) \bmod \text{LTAIL}) + 1$, for $i = 1, 2, \dots, \max(\text{LTAIL}, \text{LP}, \text{LDF})$:
TAIL(j) = 'L'
The lower tail probability, i.e., $p_i = P(X_i \leq x_{p_i} : \nu_i)$.
TAIL(j) = 'U'
The upper tail probability, i.e., $p_i = P(X_i \geq x_{p_i} : \nu_i)$.
Constraint: TAIL(j) = 'L' or 'U', for $j = 1, 2, \dots, \text{LTAIL}$.
- 3: LP – INTEGER *Input*
On entry: the length of the array P.
Constraint: LP > 0.
- 4: P(LP) – REAL (KIND=nag_wp) array *Input*
On entry: p_i , the probability of the required χ^2 -distribution as defined by TAIL with $p_i = P(j)$, $j = ((i - 1) \bmod \text{LP}) + 1$.
Constraints:
if TAIL(k) = 'L', $0.0 \leq P(j) < 1.0$;
otherwise $0.0 < P(j) \leq 1.0$.
Where $k = (i - 1) \bmod \text{LTAIL} + 1$ and $j = (i - 1) \bmod \text{LP} + 1$.
- 5: LDF – INTEGER *Input*
On entry: the length of the array DF.
Constraint: LDF > 0.
- 6: DF(LDF) – REAL (KIND=nag_wp) array *Input*
On entry: ν_i , the degrees of freedom of the χ^2 -distribution with $\nu_i = \text{DF}(j)$, $j = ((i - 1) \bmod \text{LDF}) + 1$.
Constraint: DF(j) > 0.0, for $j = 1, 2, \dots, \text{LDF}$.
- 7: X(*) – REAL (KIND=nag_wp) array *Output*
Note: the dimension of the array X must be at least $\max(\text{LTAIL}, \text{LP}, \text{LDF})$.
On exit: x_{p_i} , the deviates for the χ^2 -distribution.
- 8: IVALID(*) – INTEGER array *Output*
Note: the dimension of the array IVALID must be at least $\max(\text{LTAIL}, \text{LP}, \text{LDF})$.
On exit: IVALID(i) indicates any errors with the input arguments, with
IVALID(i) = 0
No error.
IVALID(i) = 1
On entry, invalid value supplied in TAIL when calculating x_{p_i} .
IVALID(i) = 2
On entry, invalid value for p_i .
IVALID(i) = 3
On entry, $\nu_i \leq 0.0$.

INVALID(i) = 4

p_i is too close to 0.0 or 1.0 for the result to be calculated.

INVALID(i) = 5

The solution has failed to converge. The result should be a reasonable approximation.

9: IFAIL – INTEGER

Input/Output

On entry: IFAIL must be set to 0, –1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value –1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value –1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL = 0 or –1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = 1

On entry, at least one value of TAIL, P or DF was invalid, or the solution failed to converge. Check INVALID for more information.

IFAIL = 2

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

IFAIL = 3

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

IFAIL = 4

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

IFAIL = –999

Dynamic memory allocation failed.

7 Accuracy

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

8 Further Comments

For higher accuracy the relationship described in Section 3 may be used and a direct call to G01TFF made.

9 Example

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

9.1 Program Text

```

Program g01tcfe
!   G01TCF Example Program Text

!   Mark 24 Release. NAG Copyright 2012.

!   .. Use Statements ..
Use nag_library, Only: g01tcf, nag_wp
!   .. Implicit None Statement ..
Implicit None
!   .. Parameters ..
Integer, Parameter          :: nin = 5, nout = 6
!   .. Local Scalars ..
Integer                     :: i, ifail, ldf, lout, lp, ltail
!   .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: df(:), p(:), x(:)
Integer, Allocatable        :: ivalid(:)
Character (1), Allocatable  :: tail(:)
!   .. Intrinsic Procedures ..
Intrinsic                   :: max, mod, repeat
!   .. Executable Statements ..
Write (nout,*) 'G01TCF Example Program Results'
Write (nout,*)

!   Skip heading in data file
Read (nin,*)

!   Read in the input vectors
Read (nin,*) ltail
Allocate (tail(ltail))
Read (nin,*) tail(1:ltail)

Read (nin,*) lp
Allocate (p(lp))
Read (nin,*) p(1:lp)

Read (nin,*) ldf
Allocate (df(ldf))
Read (nin,*) df(1:ldf)

!   Allocate memory for output
lout = max(ltail,lp,ldf)
Allocate (x(lout),ivalid(lout))

!   Calculate deviates (inverse CDF)
ifail = -1
Call g01tcf(ltail,tail,lp,p,ldf,df,x,ivalid,ifail)

If (ifail==0 .Or. ifail==1) Then
!   Display titles
Write (nout,*) '   TAIL      P          DF          X          IVALID'
Write (nout,*) repeat('-',48)

!   Display results
Do i = 1, lout
    Write (nout,99999) tail(mod(i-1,ltail)+1), p(mod(i-1,lp)+1), &
        df(mod(i-1,ldf)+1), x(i), ivalid(i)
End Do
End If

99999 Format (5X,A1,4X,F6.3,4X,F6.2,3X,F7.3,4X,I3)
End Program g01tcfe

```

9.2 Program Data

G01TCF Example Program Data

```

1                :: LTAIL
'L'             :: TAIL
3                :: LP
0.01 0.428 0.869 :: P
3                :: LDF
20.0 7.5 45.0   :: DF

```

9.3 Program Results

G01TCF Example Program Results

TAIL	P	DF	X	INVALID
L	0.010	20.00	8.260	0
L	0.428	7.50	6.201	0
L	0.869	45.00	55.738	0