```
END IF
TU=MIN(MAX(T1,T2),MAX(T3,T4))+SMALL
TL=MAX(MIN(T1,T2),MIN(T3,T4))-SMALL
RETURN
END
```

Algorithm AS 278

Distribution of Quadratic Forms of Multivariate Generalized Student Variables

By Bruno Lecoutre†, Jean-Luc Guigues and Jacques Poitevineau

Centre National de la Recherche Scientifique and Université René Descartes, Paris, France

[Received April 1989. Final revision November 1991]

Keywords: Alternate non-central F-distribution; Probability integral; ψ^2-distribution; Series approximation

Language

Fortran 77

Description and Purpose

Bayesian analysis of normal models involves quadratic forms of multivariate generalized Student t-variables. Hence we have the following definition: if $\mathbf{y}$ is distributed $t_{p,\,q}(\mathbf{a},\, I_p)$ (p-variate t-distribution with q degrees of freedom, centred on the real p-dimensional vector $\mathbf{a}$, and of unit scale matrix I_p) then the variable $\psi^2 = \mathbf{y}'\mathbf{y}/p$ has a ψ^2-distribution with p and q degrees of freedom and with eccentricity $a^2 = \mathbf{a}'\mathbf{a}$, denoted by $\psi^2_{p,\,q}(a^2)$.

This distribution has been introduced by Lecoutre (1981, 1985) and Rouanet and Lecoutre (1983), with a different scaling (the distribution of $\mathbf{y}'\mathbf{y}$). Here we adopt the definition in Schervish (1991), who denoted the $\psi^2_{p,q}$ (a^2)-distribution ANCF(q, a, g) ('alternate non-central F'), with the correspondence $p \rightarrow q$, $q \rightarrow a$ and $a^2/(q + a^2) \rightarrow g$.

Alternatively this distribution can be characterized as the marginal distribution of ψ^2 when given $U = u$, $(p/q)u\psi^2$ has the non-central χ^2 conditional distribution $\chi^2_p(a^2u/q)$ (with p degrees of freedom and with eccentricity a^2u/q) and U has the χ^2-distribution χ^2_q. In this form we can see precisely the differences from the two following distributions.

(a) The classical non-central F-distribution can be defined as the marginal distribution of F when given $U = u$; $(p/q)uF$ has the conditional distribution $\chi^2_p(a^2)$. Hence the two distributions are related only when $a^2 = 0$: symbolically

$$\psi^2_{p,\,q}(0) = F_{p,\,q} \qquad \text{(central } F\text{-distribution).}$$

†*Address for correspondence*: Groupe Mathématiques et Psychologie, Centre National de la Recherche Scientifique et Université René Descartes, 12 rue Cujas, 75005 Paris, France.

(b) Another distribution involved in Bayesian analysis (Geisser, 1967; Rouanet and Lecoutre, 1983; Schervish, 1987) can be characterized as the marginal distribution of X when given $U=u$; X has the conditional distribution $\chi_p^2(a^2u/q)$.

From the density of the ψ^2-distribution (Lecoutre, 1981), its cumulative distribution function can be written as a mixture of the usual incomplete beta integrals:

$$F(x) = \sum_{j=0}^{+\infty} c_j G_j(x)$$

with

$$c_j = \frac{1}{j!} \frac{\Gamma(j+q/2)}{\Gamma(q/2)} \left(\frac{q}{q+a^2}\right)^{q/2} \left(\frac{a^2}{q+a^2}\right)^j$$

(hence $\Sigma_{j=0}^{+\infty} c_j = 1$),

$$G_j(x) = I_z(j+p/2, j+q/2)$$

$$= \frac{\Gamma\{2j+(p+q)/2\}}{\Gamma(j+p/2)\,\Gamma(j+q/2)} \int_0^z y^{j+p/2-1}(1-y)^{j+q/2-1}\,\mathrm{d}y,$$

$$z = \frac{px}{q+a^2+px}$$

($F(x)$ is defined for positive real p and q).

Numerical Method

(a) $a^2 = 0$:

$$F(x) = G_0(x) = I_z(p/2, q/2)$$

with $z = px/(q+px)$.

(b) $p = 1$:

$$F(x) = \sigma I_\alpha(1/2, q/2) + I_\beta(1/2, q/2)$$

with

$$\sigma = \mathrm{sign}(x - a^2),$$

$$\alpha = \frac{a^2 + x - 2\sqrt{(a^2x)}}{q + a^2 + x - 2\sqrt{(a^2x)}}$$

and

$$\beta = \frac{a^2 + x + 2\sqrt{(a^2x)}}{q + a^2 + x + 2\sqrt{(a^2x)}}.$$

(c) $p = q$ and $z = 0.5$:

$$F(x) = 0.5.$$

(d) For the general case it can easily be shown that

$$G_0(x) = I_z(p/2, q/2) \qquad (0<z<1),$$
$$G_{j+1}(x) = G_j(x) - (v + wj)R_j$$

with

$$v = \frac{q-(p+q)z}{2},$$
$$w = 1 - 2z,$$
$$R_j = \frac{\Gamma\{2j + (p+q)/2\}}{\Gamma(j+p/2+1)\,\Gamma(j+q/2+1)} z^{j+p/2}(1-z)^{j+q/2}.$$

$F(x)$ can then be obtained as the result of two iterations:

$$c_0 = \left(\frac{q}{q+a^2}\right)^{q/2},$$
$$c_{j+1} = \frac{(j+q/2)a^2}{(j+1)(q+a^2)} c_j,$$
$$R_0 = \frac{4}{pq}\,\frac{\Gamma\{(p+q)/2\}}{\Gamma(p/2)\,\Gamma(q/2)} z^{p/2}(1-z)^{q/2},$$
$$R_j = \frac{\{2j+(p+q)/2-2\}\{2j+(p+q)/2-1\}}{(j+p/2)(j+q/2)} z(1-z)R_{j-1}.$$

For $z<0.5$, the sequence of integrals $G_j(x)$ is computed. This sequence is monotone decreasing when $q \geqslant p$ and is decreasing from $j > -v/w$ when $q<p$. For $z>0.5$, p and q are permuted (except in the coefficients c_j) and z is replaced by $1-z$. Then the same algorithm gives $1-F(x)$. For $z=0.5$ the sequence is decreasing if $p \leqslant q$. Therefore if $p>q$ we proceed as for $z>0.5$.

When we stop at N terms, and assume that $N> -v/w$ if $p>q$, the truncation error is bounded by

$$\left(1 - \sum_{j=0}^{N} c_j\right) G_N(x).$$

However, the absolute error on each term $c_j G_j(x)$ is

$$\Delta\{c_j G_j(x)\} = c_j \Delta G_j(x) + G_j(x)\,\Delta c_j.$$

We assume that the absolute errors on the $G_j(x)$ are all of the same order $\Delta G_j(x) = \Delta G$ and we note ϵ the machine precision. Hence the absolute error about $\Sigma_{j=0}^{N} c_j G_j(x)$ is approximately

$$\Delta G \sum_{j=0}^{N} c_j + \epsilon \sum_{j=0}^{N} j c_j G_j(x).$$

Consequently we adopt the following rule where δ is a given maximum absolute error about $F(x)$ and assuming $\Delta G \leqslant \delta/2$.

(i) Compute the beta integral $G_0(x) = I_z(p/2, q/2)$ with a better precision than that required for $F(x)$.

(ii) Stop the process when $j = N$ (and $j > -v/w$ when $q < p$) such that

$$\left(1 - \sum_{j=0}^{N} c_j\right) G_N(x) + \epsilon \sum_{j=0}^{N} j c_j G_j(x) < \delta/2 \tag{1a}$$

or when

$$\epsilon \sum_{j=0}^{N} j c_j G_j(x) \geqslant \delta/2, \tag{1b}$$

in which case the required accuracy cannot be reached, or when the sum of the c_js (which are decreasing for $j > a^2/2$) cannot be improved (loss of accuracy in the truncation error).

Structure

REAL FUNCTION PSI2(X, P, Q, A2, DELTA, MAXITR, IFAULT)

Formal parameters

X	Real	input:	the value of the variable x ($X \geqslant 0$)
P	Real	input:	the first number of degrees of freedom ($P > 0$)
Q	Real	input:	the second number of degrees of freedom ($Q > 0$)
A2	Real	input:	the eccentricity parameter (A2 $\geqslant 0$)
DELTA	Real	input:	the maximum absolute error required on PSI2 (must be less than 1 and greater than PREC defined in the PARAMETER statement); this stopping criterion is only used in the general case
MAXITR	Integer	input:	the maximum number of iterations (2000 is sufficient for most cases)
IFAULT	Integer	output:	a fault indicator: = 0 for no fault; = 1 for an invalid input argument (PSI2 returns −1.0); = 2 if the maximum number of iterations is reached (PSI2 returns the value at the last iteration);

= 3 if the required accuracy cannot be reached (PSI2 returns the value at the last iteration);
= 4 for an error in the BETAIX function (PSI2 returns -1.0) (usually it means that the argument $B(p/2, q/2)$, needed by BETAIN, cannot be calculated);
= 5 for an error in the ALNGAM function (PSI2 returns -1.0)

Auxiliary Algorithms

The program calls the following auxiliary functions.

(a) BETAIX is merely an intermediate function for BETAIN, used here to save code in PSI2: it calculates parameter $B(p, q)$ and then calls BETAIN.
(b) BETAIN (algorithm AS 63), due to Majumder and Bhattacharjee (1973), calculates the incomplete beta integral.
(c) ALNGAM (algorithm AS 245), due to Macleod (1989), calculates the natural logarithm of the complete gamma function $\Gamma(y)$ $(y > 0)$.

Constants

The constants PREC, XLOW and EXPL are machine dependent. PREC is the machine precision: the smallest real number such that 1.0 + PREC > 1.0. XLOW is the smallest real number representable in the machine. EXPL is the smallest admissible argument for the exponential function, i.e. ln XLOW.

Precision

The program can be converted to double precision by

(a) changing REAL FUNCTION to DOUBLE PRECISION FUNCTION,
(b) changing REAL to DOUBLE PRECISION,
(c) changing the constants in the PARAMETER statements to double-precision versions and
(d) changing the auxiliary functions ALNGAM, BETAIX and BETAIN to double-precision versions.

Time

Some execution times are given in Table 1. The tests were carried out on a mainframe system (IBM 3090 computer at the Centre Inter Régional de Calcul Electronique (CIRCE), Orsay) and on a personal computer (at 8 MHz, with a mathematics coprocessor and with a Lahey F77L 3.01 Fortran compiler). The value of

TABLE 1
Execution times

p	q	a^2	*No. of iterations*	*Times (cs) for the following machines:* *IBM 3090*	*Personal computer*
10	10	10	14	0.0263	6
10	10	100	118	0.1353	28
10	100	10	13	0.0227	5
10	100	100	85	0.1026	22
100	10	10	9	0.0196	5
100	10	100	59	0.0713	16
100	100	10	12	0.0244	5
100	100	100	80	0.0961	22
10	1000	10	13	0.0221	5
10	10	1000	1164	1.3056	280
10	10	2000	2360	2.8376	561

the variable was taken as the mean plus one standard deviation and the accuracy parameter DELTA was set to 10^{-3}. The units are hundredths of a second. The number of iterations performed is also reported. As can be seen, the execution time is greatly influenced by the eccentricity parameter a^2.

Accuracy

In any case the accuracy cannot be better than that of the auxiliary functions.

In the special cases mentioned earlier, the accuracy is that of the auxiliary function for the incomplete beta integral. In the general case, the absolute error will be controlled by argument DELTA. Loss of accuracy can occur and is flagged by the fault indicator.

Additional Comments

To prevent loss of accuracy, it is wise to take DELTA > MAXITR × PREC (where PREC is the machine precision) as a rule of thumb. Consequently the double-precision version should be favoured if one needs an absolute error lower than 10^{-3} (at least on 32-bit machines).

Acknowledgements

The authors wish to thank the referee and the Algorithms Editor for their remarks which have improved both the paper and the program.

All the development and tests of the programs were carried out at the CIRCE, Orsay.

References

Geisser, S. (1967) Estimation associated with linear discriminants. *Ann. Math. Statist.*, **38**, 807–817.

Lecoutre, B. (1981) Extensions de l'analyse de la variance: l'analyse bayésienne des comparaisons. *Math. Sci. Hum.*, **75**, 49–69.

—— (1985) Reconsideration of the *F*-test of the analysis of variance: the semi-Bayesian significance test. *Communs Statist. Theory Meth.*, **14**, 2437–2446.

Macleod, A. J. (1989) Algorithm AS 245: a robust and reliable algorithm for the logarithm of the gamma function. *Appl. Statist.*, **38**, 397–402.
Majumder, K. L. and Bhattacharjee, G. P. (1973) Algorithm AS 63: The incomplete beta integral. *Appl. Statist.*, **22**, 409–411.
Rouanet, H. and Lecoutre, B. (1983) Specific inference in ANOVA: from significance tests to Bayesian procedures. *Br. J. Math. Statist. Psych.*, **36**, 252–268.
Schervish, M. J. (1987) A review of multivariate analysis. *Statist. Sci.*, **2**, 396–443.
——— (1991) Bayesian analysis of linear models. *4th Valencia Int. Meet. Bayesian Statistics, Peñiscola.*

```
      REAL FUNCTION PSI2(X, P, Q, A2, DELTA, MAXITR, IFAULT)
C
C        ALGORITHM AS 278.1 APPL.STATIST. (1992) VOL.41, NO.3
C
C        Calculates the probability that a random variable distributed
C        according to the psi-square distribution with P and Q degrees
C        of freedom and A2 eccentricity parameter, is less than or
C        equal to X
C
      REAL X, P, Q, A2, DELTA
      INTEGER MAXITR, IFAULT
C
      REAL ALNGAM, BETAIX
      EXTERNAL ALNGAM, BETAIX
C
      REAL AQAL, BL, CL, DAX, DJ, DJ1, ERP, ERR, EXPL, G,
     *     GCJ, GCL, HALF, ONE, PQ2, PREC, P2, QQAL, Q2, R,
     *     RL, SUM, SUMC, SUMCP, TWO, V, W, XLOW, XP, Y, YYL,
     *     Z, ZERO, ZM, ZN
      INTEGER IOK, J, JJ, JJJ, JOK, KOK
      LOGICAL  LALF
C
      PARAMETER (ZERO=0.0E0, HALF=0.5E0, ONE=1.0E0, TWO=2.0E0)
C
C        machine-dependent constants
C
      PARAMETER (PREC=1.0E-6, XLOW=1.2E-38, EXPL=-87.315E0)
      PARAMETER (XP=XLOW/PREC)
C
      IFAULT = 0
C
C        test for valid input arguments
C
      IF (X .LT. ZERO .OR. A2 .LT. ZERO
     *   .OR. P .LE. ZERO .OR. Q .LE. ZERO
     *   .OR. DELTA .GE. ONE .OR. DELTA .LE. PREC) THEN
         IFAULT = 1
         PSI2 = -ONE
         RETURN
      ENDIF
C
C        define useful parameters
C
      P2  = P * HALF
      Q2  = Q * HALF
      PQ2 = P2 + Q2
C
C        case A2 = 0
C
      IF (A2 .LT. PREC) THEN
         PSI2 = BETAIX( P*X/(P*X+Q), P2, Q2, IOK )
         IF (IOK .NE. 0) THEN
            IFAULT = 4
            PSI2 = -ONE
```

```
            ENDIF
            RETURN
         ENDIF
C
C           case P = 1
C
         IF (ABS(P-ONE) .LT. PREC) THEN
            DAX  = TWO * SQRT( A2*X )
            PSI2 = HALF * ( SIGN( ONE, X-A2 ) *
     *             BETAIX( (A2+X-DAX)/(A2+X-DAX+Q),HALF,Q2,IOK ) +
     *             BETAIX( (A2+X+DAX)/(A2+X+DAX+Q),HALF,Q2,JOK ) )
            IF (IOK .NE. 0 .OR. JOK .NE. 0) THEN
               IFAULT = 4
               PSI2 = -ONE
            ENDIF
            RETURN
         ENDIF
C
         Y = P*X / (A2+Q+P*X)
C
C           case P = Q and Y = 0.5
C
         IF (ABS(Y-HALF) .LT. PREC .AND. ABS(P-Q) .LT. PREC) THEN
            PSI2 = HALF
            RETURN
         ENDIF
C
C           calculate 1-F(X) or F(X) (LALF is the indicator)
C
         IF (Y .GT. HALF .OR. (Y .EQ. HALF .AND. P .GT. Q) ) THEN
            LALF = .TRUE.
            Z    = ONE - Y
            ZM   = Q2
            ZN   = P2
          ELSE
            LALF = .FALSE.
            Z    = Y
            ZM   = P2
            ZN   = Q2
         ENDIF
C
C           Y near 0 or 1 ?
C
         IF (Z .LT. PREC) THEN
            IF( LALF )THEN
               PSI2 = ONE
              ELSE
               PSI2 = ZERO
            ENDIF
            RETURN
         ENDIF
C
C           General case (iterations)
C           G's are decreasing for J >= JJ = -V/W + 1
C           CL's  are decreasing for J >= JJJ = A2/2 + 1
C           GCJ, SUMC, SUMCP are only used for stopping rule
C           logs are used to avoid underflows
C
         ERR  = DELTA * HALF
         ERP  = ERR / PREC
         YYL  = LOG( Y*(ONE-Y) )
         QQAL = Q2 * LOG( Q/(Q+A2) )
         AQAL = LOG( A2/(Q+A2) )
         V    = PQ2 * Z - ZN
         W    = Z + Z - ONE
C
C           initialize
C
```

```
      BL = ZERO
      CL = QQAL
      G  = BETAIX( Z, ZM, ZN, IOK )
      IF (IOK .NE. 0) THEN
         IFAULT = 4
         PSI2 = -ONE
         RETURN
      ENDIF
      RL = P2 * LOG(Y)  +  Q2 * LOG(ONE-Y)  +  ALNGAM(PQ2,IOK)
     *     - ALNGAM(P2,JOK) - ALNGAM(Q2,KOK) - LOG(P2) - LOG(Q2)
      IF (IOK+JOK+KOK .NE. 0) THEN
         IFAULT = 5
         PSI2 = -ONE
         RETURN
      ENDIF
      IF (RL .GE. EXPL) THEN
         R = EXP( RL )
        ELSE
         R = ZERO
      ENDIF
      SUM   = ZERO
      GCJ   = ZERO
      SUMC  = ZERO
      SUMCP = SUMC
      DJ    = ZERO
C
C        define minimum numbers of iterations (JJ and JJJ)
C
      IF (ZM .GT. ZN .AND. W .NE. ZERO) THEN
         JJ = INT( -V/W ) + 1
        ELSE
         JJ = 0
      ENDIF
      JJJ = INT( A2*HALF ) + 1
C
C        iteration loop
C
      DO 10 J = 0, MAXITR
C
         IF (G .GT. ZERO) THEN
            GCL = LOG(G) + CL
            IF (GCL .GE. EXPL) THEN
               GCL = EXP( GCL )
               SUM = SUM + GCL
               GCJ = GCJ + GCL*DJ
C                 check loss of accuracy
               IF (GCJ .GE. ERP) THEN
                  IFAULT = 3
                  GOTO 20
               ENDIF
            ENDIF
         ENDIF
C
         IF (CL .GE. EXPL) SUMC = SUMC + EXP(CL)
C
C           check accuracy (stopping rule)
C
         IF (J .GE. JJ) THEN
C              XP is used to prevent possible underflow
            IF (GCJ .GE. XP) THEN
               IF (PREC*GCJ+G*(ONE-SUMC) .LT. ERR) GOTO 20
              ELSE
               IF (         G*(ONE-SUMC) .LT. ERR) GOTO 20
            ENDIF
            IF (J .GE. JJJ .AND. SUMC .EQ. SUMCP .AND.
     *         ABS(ONE-SUMC) .GE. PREC) THEN
               IFAULT = 3
               GOTO 20
```

```
            ENDIF
         ENDIF
C
C           prepare next iteration
C
         SUMCP = SUMC
         DJ1   = DJ + ONE
         BL    = BL + LOG( (Q2+DJ)/DJ1 )
         CL    = BL + QQAL + DJ1*AQAL
         G     = G  + (V+W*DJ)*R
         RL    = RL + YYL +
     *              LOG( ((DJ+DJ+PQ2)/(DJ1+P2)) * ((DJ+DJ1+PQ2)/(DJ1+Q2)) )
         IF (RL .GE. EXPL) THEN
            R = EXP( RL )
           ELSE
            R = ZERO
         ENDIF
         DJ = DJ1

   10 CONTINUE
C
C        we get here if maximum number of iterations is reached
C
      IFAULT = 2
C
C           the end
C
   20 CONTINUE
      IF (LALF) THEN
         PSI2 = ONE - SUM
        ELSE
         PSI2 = SUM
      ENDIF
      END
C
      REAL FUNCTION BETAIX(X, P, Q, IFAULT)
C
C        ALGORITHM AS 278.2 APPL.STATIST. (1992) VOL.41, NO.3
C
      REAL X, P, Q
      INTEGER IFAULT
C
      REAL ALNGAM, BETAIN
      EXTERNAL ALNGAM, BETAIN
C
      REAL       BETA, EXPL, ZERO
      INTEGER    IOK, JOK, KOK
C
      PARAMETER (ZERO=0.0E0)
C
C        machine-dependent constant
C
      PARAMETER (EXPL=-87.315E0)
C
      BETAIX = ZERO
C
C        calculate beta parameter for betain
C
      BETA = ALNGAM(P,JOK) + ALNGAM(Q,KOK) - ALNGAM(P+Q,IOK)
      IF (IOK+JOK+KOK .NE. 0) THEN
         IFAULT = 5
         RETURN
      ENDIF
      IF (BETA .GE. EXPL) THEN
         BETA =  EXP(BETA)
        ELSE
         IFAULT = 4
         RETURN
```

```
ENDIF
BETAIX = BETAIN(X, P, Q, BETA, IFAULT)
END
```

Remark AS R90

Least Squares Initial Values for the L_1-norm Fitting of a Straight Line—a Remark on Algorithm AS 238: A Simple Recursive Procedure for the L_1 Norm Fitting of a Straight Line

By R. W. Farebrother†

University of Manchester, UK

[Received October 1989. Final revision August 1991]

Keywords: Least absolute values; Least squares initial values; Linear regression; Minimum absolute deviations; Sorting procedures; Weighted median

Introduction

In a series of recent papers Gentle, Narula and Sposito (Gentle *et al.*, 1987, 1988; Narula *et al.*, 1991) have compared the computational efficiency of several algorithms for evaluating the L_1-norm estimates of α^* and β^* in the simple linear regression model

$$y_i = \alpha^* + \beta^* x_i + \epsilon_i, \qquad i = 1, 2, \ldots, n. \tag{1}$$

In particular they have drawn attention to the procedure proposed by Josvanger and Sposito (1983) as its performance was found to compare favourably with that of Armstrong and Kung's (1978) implementation of Barrodale and Roberts's (1973) procedure.

Observing that Josvanger and Sposito's procedure is essentially a variant of Karst's (1958) procedure prompts me to propose the following three variants of two procedures (karst and simbarrob) published by Farebrother (1988):

(a) karstam, a variant of karst whose initial basis is supplied by the additional parameter jin;
(b) josito, a variant of karstam which finds the weighted median by a sample separation procedure;
(c) simbarrob2, a variant of simbarrob whose initial basis is supplied by the additional parameters jin1 and jin2.

Structure

The three additional parameters

†*Address for correspondence*: Department of Econometrics and Social Statistics, University of Manchester, Manchester, M13 9PL, UK.