```
      SUBROUTINE TRANS(A, LB, LC, LD, N)
C
C        ALGORITHM AS 98.1  APPL. STATIST. (1976) VOL.25, NO.2
C
C        THIS SUBROUTINE CARRIES OUT THE MATRIX OPERATIONS USED
C        TO REDUCE THE SYSTEM OF EQUATIONS TO AN EQUIVALENT SET
C        OF EQUATIONS FOR WHICH THERE IS A SMALLER SET OF
C        FEASIBLE SOLUTIONS.
C
      DOUBLE PRECISION A, D
      DIMENSION A(5, 5)
      D = LD
      DO 300 LE = 1, N
  300 A(LB, LE) = A(LB, LE) + D * A(LC, LE)
      DO 301 LE = 1, N
  301 A(LE, LB) = A(LE, LB) + D * A(LE, LC)
      RETURN
      END
```

Algorithm AS 99

Fitting Johnson Curves by Moments

By I. D. HILL, R. HILL† and R. L. HOLDER

MRC Clinical Research Centre Harrow, Middx, Britain

Wolfson Research Laboratories, Queen Elizabeth Medical Centre, Birmingham, Britain

Department of Mathematical Statistics, University of Birmingham, Britain

Keywords: JOHNSON CURVES; TRANSFORMATIONS OF NORMAL CURVE; METHOD OF MOMENTS; CURVE FITTING

LANGUAGE

ISO Fortran

DESCRIPTION AND PURPOSE

Johnson (1949) described a system of frequency curves consisting of:

(1) the lognormal system (or S_L): $z = \gamma + \delta \ln (x - \xi) \quad \xi < x$,
(2) the unbounded system (or S_U): $z = \gamma + \delta \sinh^{-1} ((x - \xi)/\lambda)$,
(3) the bounded system (or S_B): $z = \gamma + \delta \ln ((x - \xi)/(\xi + \lambda - x)) \quad \xi < x < \xi + \lambda$,

where z is a standardized normal variable in each case.

For the sake of completeness we have included (4) the normal curve itself; (5) the special case of the S_B curves on the $\beta_2 = \beta_1 + 1$ boundary, which we have called S_T (T standing for "two-ordinate").

To make the first four moments of x match those of any required distribution it is necessary to determine which of the transformations is required and to evaluate the parameters γ, δ, λ and ξ.

Fitting by moments is not always a desirable procedure. However, in a number of situations it is quite adequate, without any pretence that it can be regarded as giving the "best" solution in any sense. In particular, it may be worth while to produce starting values from which to seek for a maximum likelihood solution. Also, moments can sometimes be calculated theoretically, and thus not be subject to sampling error, in which case the objections to fitting by moments do not apply. For discussion of some of the alternative methods of estimating parameters, see Ord (1972).

† Present address: Research Planning and Development Section, Birmingham Social Services Department.

This algorithm supplements Tables 34, 35 and 36 of Pearson and Hartley (1972), for S_U and S_B curves. These tables are perfectly adequate for many purposes, but interpolation or extrapolation may be hazardous when the required curve is near one of the boundaries.

Numerical Method

Defining, as is customary, $\sqrt{\beta_1}$, as μ_3/σ^3 and β_2 as μ_4/σ^4, the S_L curves lie on a line in the $\beta_1\beta_2$ plane—thus for these curves β_1 determines β_2. Using ω to denote $\exp(\delta^{-2})$, the S_L β_2 value is found by solving

$$(\omega-1)(\omega+2)^2 = \beta_1$$

for ω, and then evaluating

$$\beta_2 = \omega^4+2\omega^3+3\omega^2-3.$$

If the required β_2 is less than this value, S_B (or S_T) is appropriate; if greater, S_U is appropriate.

(1) *S_L curves*

ω having been evaluated as above,

$$\delta = (\ln\omega)^{-\frac{1}{2}}, \qquad \gamma = \tfrac{1}{2}\delta\ln\{\omega(\omega-1)/\mu_2\},$$
$$\xi = \pm\mu_1' - \exp\{(1/2\delta-\gamma)/\delta\}, \qquad \lambda = \pm 1,$$

where the $\pm$ is determined in each case to be the sign of μ_3. As Johnson (1949) points out, only three parameters are necessary for an S_L curve, but we have found it convenient to include λ as above.

(2) *S_U curves*

When $\beta_1 = 0$, the required curve is symmetrical, and

$$\omega = \{(2\beta_2-2)^{\frac{1}{2}}-1\}^{\frac{1}{2}}; \quad \delta = (\ln\omega)^{-\frac{1}{2}}; \quad \gamma = 0.$$

For an asymmetrical curve

$$\omega_1 = \{(2\beta_2-2{\cdot}8\beta_1-2)^{\frac{1}{2}}-1\}^{\frac{1}{2}}$$

is taken as a first estimate, and ω, δ and γ found by Johnson's iterative method (Elderton and Johnson, 1969, p. 127). The sign of γ is set to be the opposite of that of μ_3.

In either case ξ and λ are then found from

$$\mu_2 = \tfrac{1}{2}\lambda^2(\omega-1)\{\omega\cosh(2\gamma/\delta)+1\}; \quad \mu_1' = \xi - \lambda\omega^{\frac{1}{2}}\sinh(\gamma/\delta).$$

(3) *S_B curves*

Approaching the S_T boundary, $\delta\to 0$; approaching the S_L boundary δ tends to the same value as for an S_L curve. A first approximation to δ can be found by interpolating between these two values. The interpolation is made by assuming the shape of the function to be the same at the required β_1 value as it is between the same two δ values when $\beta_1 = 0$. This is well approximated by

$$\delta = (0{\cdot}626\beta_2-0{\cdot}408)/(3{\cdot}0-\beta_2)^{0{\cdot}479} \quad \text{if } \beta_2 \geqslant 1{\cdot}8,$$

and by

$$\delta = 0{\cdot}8(\beta_2-1) \quad \text{otherwise.}$$

For a given β_1 and first approximation to δ, a first approximation to γ is found using formulae due to Draper (1951).

Evaluation of the first six moments at the given δ and γ values, using Draper's (1952) form of Goodwin's (1949) integral, then enables a two-dimensional Newton–Raphson process to converge on the required values.

Since the first six moments are evaluated at each stage, when the required δ and γ have been found, the first two moments are available to determine λ and ξ.

(4) *Normal curves*

δ is set to the required value of $1/\sigma$, and γ to $\bar{x}/\sigma$; ξ and λ are set, arbitrarily, to 0.

(5) S_T *curves*

Since it is unnecessarily complicated to regard these as transformations of the normal curve, totally different meanings of the parameters are used. ξ and λ are set to the two values at which ordinates occur, and δ to the proportion of values at λ. γ is set, arbitrarily, to 0.

Structure

SUBROUTINE JNSN (*XBAR*, *SD*, *RB*1, *BB*2, *ITYPE*, *GAMMA*, *DELTA*, *XLAM*, *XI*, *IFAULT*)

Formal parameters

XBAR	Real	input:	the required mean
SD	Real	input:	the required standard deviation
*RB*1	Real	input:	the required value of $\sqrt{\beta_1}$, taking the same sign as the third moment about the mean
*BB*2	Real	input:	the required value of β_2; or a negative value to indicate that an S_L curve is desired (or a normal if $\beta_1 = 0$), with the given values of the other three input parameters
ITYPE	Integer	output:	the type of curve fitted: $1 = S_L$, $2 = S_U$, $3 = S_B$, $4 =$ normal, $5 = S_T$
GAMMA	Real	output:	fitted value of γ
DELTA	Real	output:	fitted value of δ
XLAM	Real	output:	fitted value of λ
XI	Real	output:	fitted value of ξ
IFAULT	Integer	output:	see *failure indications* below

Failure indications

IFAULT = 0	indicates successful completion
IFAULT = 1	indicates a required standard deviation of less than zero
IFAULT = 2	indicates $\beta_2 < \beta_1 + 1$
IFAULT = 3	S_B fitting has failed to converge, so an S_L fit or an S_T fit has been made instead. The user should check whether the substituted fit is good enough for the purpose

Auxiliary Algorithms

Subroutines *SUFIT*, to fit an S_U distribution, *SBFIT*, to fit an S_B distribution, and *MOM*, to find the first six moments of an S_B distribution are included as Algorithms AS 99.1, AS 99.2 and AS 99.3 respectively.

MOM may, if desired, be replaced by any standard quadrature routine to find the first six moments.

Precision

Single precision arithmetic is generally sufficient, even on machines that use only 32 bits for real number representation.

However, if an S_B fit is required close to either the S_L or the S_T boundary, convergence may not be achieved; double precision working may then be helpful, if the approximation of taking a distribution on the boundary is regarded as inadequate. Increasing the values of *LIMIT*, set in DATA statements in *SBFIT* and *MOM* may also be worth considering, provided that the usage of computer time is not critical.

To produce a double precision version: (i) change the word REAL to DOUBLE PRECISION in each of the four subroutines; (ii) give the real constants included in DATA statements double precision values (the real constants not in DATA statements only need to be approximate and need not be changed); (iii) change ABS to DABS in 11 places, EXP to DEXP in 5 places, SQRT to DSQRT in 18 places, ALOG to DLOG in 5 places and SIGN to DSIGN in one place.

Accuracy

The parameters found are such that the values of $\sqrt{\beta_1}$ and β_2 achieved are both within $\pm TOL$ of the required values. The value of *TOL* is set in DATA statements in *JNSN*, *SUFIT* and *SBFIT*. It may be changed if desired but should be identical in the three places. *TT* in *SBFIT* should be TOL^2.

The constants *ZZ* and *VV*, set in a DATA statement in *MOM*, determine the accuracy of convergence of the outer and inner loops of the evaluation. *VV* should be considerably smaller than *ZZ*, which in turn should be smaller than *TT* in *SBFIT*.

Time

On a PDP-11/40, in single precision, typical times to fit an S_L, an S_U, an easy S_B (midway between the S_L and S_T boundaries) and a difficult S_B (close to a boundary) are 0·028 sec, 0·049 sec, 2·1 sec and 25 sec respectively.

Additional Comment

Where one end of the distribution is bounded (S_L), or both ends are bounded (S_B), there may often be a physical reason to know the value(s) of the bound(s). Fitting should then, usually, be performed conditional on the known values of such bounds. The current algorithm does not deal with these cases.

Example

As an illustration of how Algorithms AS 99, AS 100.2 and AS 66 may be put together to estimate tail areas of a distribution whose moments are known, we present the following fragment of program, that will evaluate an approximation to the area above the value C of a χ^2 distribution having F degrees of freedom. We deliberately choose something for which the precise answers are known as a test on the accuracy of the method.

```
C        SNV IS ALGORITHM AS 100.2
C        ALNORM IS ALGORITHM AS 66
C
      CALL JNSN(F, SQRT(F + F), SQRT(8.0 / F), 12.0 / F + 3.0,
     1   IT, G, D, XL, XI, IFAULT)
      IF (IFAULT .NE. 0) GOTO 100
      A = ALNORM(SNV(C, IT, G, D, XL, XI, IFAULT), .TRUE.)
```

Taking $F = 1, 2, 3, 4$ and values of C for known percentage points, the following results were found:

	Correct result	0·50	0·10	0·01
$F = 1$		0·539	0·0952	0·0105
$F = 2$	Value of	0·512	0·0972	0·0105
$F = 3$	A	0·505	0·0984	0·0104
$F = 4$		0·502	0·0990	0·0104

Acknowledgements

We are indebted to J. Draper for permission to make use of his M.Sc. thesis, and to the referee for many useful comments.

References

Draper, J. (1951). Properties of distributions resulting from certain simple transformations of the normal distribution. M.Sc. thesis, University of London.

—— (1952). Properties of distributions resulting from certain simple transformations of the normal distribution. *Biometrika*, **39**, 290–301.

Elderton, W. P. and Johnson, N. L. (1969). *Systems of Frequency Curves*. Cambridge: University Press.

Johnson, N. L. (1949). Systems of frequency curves generated by methods of translation. *Biometrika*, **36**, 149–176.

Ord, J. K. (1972). *Families of Frequency Distributions*. London: Griffin.

Pearson, E. S. and Hartley, H. O. (1972). *Biometrika Tables for Statisticians*, Vol. 2. Cambridge: University Press.


```
      SUBROUTINE JNSN(XBAR, SD, RB1, BB2, ITYPE, GAMMA, DELTA, XLAM, XI,
     *  IFAULT)
C
C        ALGORITHM AS 99  APPL. STATIST. (1976) VOL.25, NO.2
C
C        FINDS TYPE AND PARAMETERS OF A JOHNSON CURVE
C        WITH GIVEN FIRST FOUR MOMENTS
C
      REAL XBAR, SD, RB1, BB2, GAMMA, DELTA, XLAM, XI, TOL, B1, B2, Y,
     *  X, U, W, ZERO, ONE, TWO, THREE, FOUR, HALF, QUART
      LOGICAL FAULT
      DATA TOL /0.01/
      DATA ZERO, ONE, TWO, THREE, FOUR, HALF, QUART
     *    /0.0, 1.0, 2.0, 3.0, 4.0, 0.5, 0.25/
      IFAULT = 1
      IF (SD .LT. ZERO) RETURN
      IFAULT = 0
      XI = ZERO
      XLAM = ZERO
      GAMMA = ZERO
      DELTA = ZERO
      IF (SD .GT. ZERO) GOTO 10
      ITYPE = 5
      XI = XBAR
      RETURN
   10 B1 = RB1 * RB1
      B2 = BB2
      FAULT = .FALSE.
C
C        TEST WHETHER LOGNORMAL (OR NORMAL) REQUESTED
C
      IF (B2 .GE. ZERO) GOTO 30
   20 IF (ABS(RB1) .LE. TOL) GOTO 70
      GOTO 80
C
C        TEST FOR POSITION RELATIVE TO BOUNDARY LINE
C
   30 IF (B2 .GT. B1 + TOL + ONE) GOTO 60
      IF (B2 .LT. B1 + ONE) GOTO 50
C
C        ST DISTRIBUTION
C
   40 ITYPE = 5
      Y = HALF + HALF * SQRT(ONE - FOUR / (B1 + FOUR))
      IF (RB1 .GT. ZERO) Y = ONE - Y
      X = SD / SQRT(Y * (ONE - Y))
      XI = XBAR - Y * X
      XLAM = XI + X
      DELTA = Y
      RETURN
   50 IFAULT = 2
      RETURN
```

```
   60 IF (ABS(RB1) .GT. TOL .OR. ABS(B2 - THREE) .GT. TOL) GOTO 80
C
C        NORMAL DISTRIBUTION
C
   70 ITYPE = 4
      DELTA = ONE / SD
      GAMMA = -XBAR / SD
      RETURN
C
C        TEST FOR POSITION RELATIVE TO LOGNORMAL LINE
C
   80 U = ONE / THREE
      X = HALF * B1 + ONE
      Y = RB1 * SQRT(QUART * B1 + ONE)
      W = (X + Y) ** U + (X - Y) ** U - ONE
      U = W * W * (THREE + W * (TWO + W)) - THREE
      IF (B2 .LT. ZERO .OR. FAULT) B2 = U
      X = U - B2
      IF (ABS(X) .GT. TOL) GOTO 90
C
C        LOGNORMAL (SL) DISTRIBUTION
C
      ITYPE = 1
      XLAM = SIGN(ONE, RB1)
      U = XLAM * XBAR
      X = ONE / SQRT(ALOG(W))
      DELTA = X
      Y = HALF * X * ALOG(W * (W - ONE) / (SD * SD))
      GAMMA = Y
      XI = U - EXP((HALF / X - Y) / X)
      RETURN
C
C        SB OR SU DISTRIBUTION
C
   90 IF (X .GT. ZERO) GOTO 100
      ITYPE = 2
      CALL SUFIT(XBAR, SD, RB1, B2, GAMMA, DELTA, XLAM, XI)
      RETURN
  100 ITYPE = 3
      CALL SBFIT(XBAR, SD, RB1, B2, GAMMA, DELTA, XLAM, XI, FAULT)
      IF (.NOT.FAULT) RETURN
C
C        FAILURE - TRY TO FIT APPROXIMATE RESULT
C
      IFAULT = 3
      IF (B2 .GT. B1 + TWO) GOTO 20
      GOTO 40
      END
C
      SUBROUTINE SUFIT(XBAR, SD, RB1, B2, GAMMA, DELTA, XLAM, XI)
C
C        ALGORITHM AS 99.1  APPL. STATIST. (1976) VOL.25, NO.2
C
C        FINDS PARAMETERS OF JOHNSON SU CURVE WITH
C        GIVEN FIRST FOUR MOMENTS
C
      REAL XBAR, SD, RB1, B2, GAMMA, DELTA, XLAM, XI, TOL, B1, B3, W, Y,
     *  W1, WM1, Z, V, A, B, X, ZERO, ONE, TWO, THREE, FOUR, SIX,
     *  SEVEN, EIGHT, NINE, TEN, HALF, ONE5, TWO8
      DATA TOL /0.01/
      DATA ZERO, ONE, TWO, THREE, FOUR, SIX, SEVEN, EIGHT, NINE, TEN,
     *  HALF, ONE5, TWO8 /0.0, 1.0, 2.0, 3.0, 4.0, 6.0, 7.0, 8.0, 9.0,
     *  10.0, 0.5, 1.5, 2.8/
      B1 = RB1 * RB1
      B3 = B2 - THREE
C
C        W IS FIRST ESTIMATE OF EXP(DELTA ** (-2))
C
      W = SQRT(SQRT(TWO * B2 - TWO8 * B1 - TWO) - ONE)
      IF (ABS(RB1) .GT. TOL) GOTO 10
```

```
C         SYMMETRICAL CASE - RESULTS ARE KNOWN
C
      Y = ZERO
      GOTO 20
C
C         JOHNSON ITERATION (USING Y FOR HIS M)
C
   10 W1 = W + ONE
      WM1 = W - ONE
      Z = W1 * B3
      V = W * (SIX + W * (THREE + W))
      A = EIGHT * (WM1 * (THREE + W * (SEVEN + V)) - Z)
      B = 16.0 * (WM1 * (SIX + V) - B3)
      Y = (SQRT(A * A - TWO * B * (WM1 * (THREE + W *
     *  (NINE + W * (TEN + V))) - TWO * W1 * Z)) - A) / B
      Z = Y * WM1 * (FOUR * (W + TWO) * Y + THREE * W1 * W1) ** 2 /
     *  (TWO * (TWO * Y + W1) ** 3)
      V = W * W
      W = SQRT(SQRT(ONE - TWO * (ONE5 - B2 + (B1 *
     *  (B2 - ONE5 - V * (ONE + HALF * V))) / Z)) - ONE)
      IF (ABS(B1 - Z) .GT. TOL) GOTO 10
C
C         END OF ITERATION
C
      Y = Y / W
      Y = ALOG(SQRT(Y) + SQRT(Y + ONE))
      IF (RB1 .GT. ZERO) Y = -Y
   20 X = SQRT(ONE / ALOG(W))
      DELTA = X
      GAMMA = Y * X
      Y = EXP(Y)
      Z = Y * Y
      X = SD / SQRT(HALF * (W - ONE) * (HALF * W * (Z + ONE / Z) + ONE))
      XLAM = X
      XI = (HALF * SQRT(W) * (Y - ONE / Y)) * X + XBAR
      RETURN
      END
C
      SUBROUTINE SBFIT(XBAR, SIGMA, RTB1, B2, GAMMA, DELTA, XLAM, XI,
     *  FAULT)
C
C         ALGORITHM AS 99.2  APPL. STATIST. (1976) VOL.25, NO.2
C
C         FINDS PARAMETERS OF JOHNSON SB CURVE WITH
C         GIVEN FIRST FOUR MOMENTS
C
      REAL HMU(6), DERIV(4), DD(4), XBAR, SIGMA, RTB1, B2, GAMMA, DELTA,
     *  XLAM, XI, TT, TOL, RB1, B1, E, U, X, Y, W, F, D, G, S, H2, T,
     *  H2A, H2B, H3, H4, RBET, BET2, ZERO, ONE, TWO, THREE, FOUR, SIX,
     *  HALF, QUART, ONE5
      LOGICAL NEG, FAULT
      DATA TT, TOL, LIMIT /1.0E-4, 0.01, 50/
      DATA ZERO, ONE, TWO, THREE, FOUR, SIX, HALF, QUART, ONE5
     *  /0.0, 1.0, 2.0, 3.0, 4.0, 6.0, 0.5, 0.25, 1.5/
      RB1 = ABS(RTB1)
      B1 = RB1 * RB1
      NEG = RTB1 .LT. ZERO
C
C         GET D AS FIRST ESTIMATE OF DELTA
C
      E = B1 + ONE
      U = ONE / THREE
      X = HALF * B1 + ONE
      Y = RB1 * SQRT(QUART * B1 + ONE)
      W = (X + Y) ** U + (X - Y) ** U - ONE
      F = W * W * (THREE + W * (TWO + W)) - THREE
      E = (B2 - E) / (F - E)
      IF (ABS(RB1) .GT. TOL) GOTO 5
      F = TWO
      GOTO 20
    5 D = ONE / SQRT(ALOG(W))
```

```
      IF (D .LT. 0.64) GOTO 10
      F = TWO - 8.5245 / (D * (D * (D - 2.163) + 11.346))
      GOTO 20
   10 F = 1.25 * D
   20 F = E * F + ONE
      IF (F .LT. 1.8) GOTO 25
      D = (0.626 * F - 0.408) * (THREE - F) ** (-0.479)
      GOTO 30
   25 D = 0.8 * (F - ONE)
C
C         GET G AS FIRST ESTIMATE OF GAMMA
C
   30 G = ZERO
      IF (B1 .LT. TT) GOTO 70
      IF (D .GT. ONE) GOTO 40
      G = (0.7466 * D ** 1.7973 + 0.5955) * B1 ** 0.485
      GOTO 70
   40 IF (D .LE. 2.5) GOTO 50
      U = 0.0124
      Y = 0.5291
      GOTO 60
   50 U = 0.0623
      Y = 0.4043
   60 G = B1 ** (U * D + Y) * (0.9281 + D * (1.0614 * D - 0.7077))
   70 M = 0
C
C         MAIN ITERATION STARTS HERE
C
   80 M = M + 1
      FAULT = M .GT. LIMIT
      IF (FAULT) RETURN
C
C         GET FIRST SIX MOMENTS FOR LATEST G AND D VALUES
C
      CALL MOM(G, D, HMU, FAULT)
      IF (FAULT) RETURN
      S = HMU(1) * HMU(1)
      H2 = HMU(2) - S
      FAULT = H2 .LE. ZERO
      IF (FAULT) RETURN
      T = SQRT(H2)
      H2A = T * H2
      H2B = H2 * H2
      H3 = HMU(3) - HMU(1) * (THREE * HMU(2) - TWO * S)
      RBET = H3 / H2A
      H4 = HMU(4) - HMU(1) * (FOUR * HMU(3) - HMU(1) *
     *  (SIX * HMU(2) - THREE * S))
      BET2 = H4 / H2B
      W = G * D
      U = D * D
C
C         GET DERIVATIVES
C
      DO 120 J = 1, 2
      DO 110 K = 1, 4
      T = K
      IF (J .EQ. 1) GOTO 90
      S = ((W - T) * (HMU(K) - HMU(K + 1)) + (T + ONE) *
     *  (HMU(K + 1) - HMU(K + 2))) / U
      GOTO 100
   90 S = HMU(K + 1) - HMU(K)
  100 DD(K) = T * S / D
  110 CONTINUE
      T = TWO * HMU(1) * DD(1)
      S = HMU(1) * DD(2)
      Y = DD(2) - T
      DERIV(J) = (DD(3) - THREE * (S + HMU(2) * DD(1) - T * HMU(1))
     *  - ONE5 * H3 * Y / H2) / H2A
      DERIV(J + 2) = (DD(4) - FOUR * (DD(3) * HMU(1) + DD(1) * HMU(3))
     *  + SIX * (HMU(2) * T + HMU(1) * (S - T * HMU(1)))
     *  - TWO * H4 * Y / H2) / H2B
```

```
  120 CONTINUE
      T = ONE / (DERIV(1) * DERIV(4) - DERIV(2) * DERIV(3))
      U = (DERIV(4) * (RBET - RB1) - DERIV(2) * (BET2 - B2)) * T
      Y = (DERIV(1) * (BET2 - B2) - DERIV(3) * (RBET - RB1)) * T
C
C         FORM NEW ESTIMATES OF G AND D
C
      G = G - U
      IF (B1 .EQ. ZERO .OR. G .LT. ZERO) G = ZERO
      D = D - Y
      IF (ABS(U) .GT. TT .OR. ABS(Y) .GT. TT) GOTO 80
C
C         END OF ITERATION
C
      DELTA = D
      XLAM = SIGMA / SQRT(H2)
      IF (NEG) GOTO 130
      GAMMA = G
      GOTO 140
  130 GAMMA = -G
      HMU(1) = ONE - HMU(1)
  140 XI = XBAR - XLAM * HMU(1)
      RETURN
      END
C
      SUBROUTINE MOM(G, D, A, FAULT)
C
C         ALGORITHM AS 99.3  APPL. STATIST. (1976) VOL.25, NO.2
C
C         EVALUATES FIRST SIX MOMENTS OF A JOHNSON
C         SB DISTRIBUTION, USING GOODWIN METHOD
C
      REAL A(6), B(6), C(6), G, D, ZZ, VV, RTTWO, RRTPI, W, E, R, H, T,
     *  U, Y, X, V, F, Z, S, P, Q, AA, AB, EXPA, EXPB, ZERO, ONE, TWO,
     *  THREE
      LOGICAL L, FAULT
      DATA ZZ, VV, LIMIT /1.0E-5, 1.0E-8, 500/
C
C         RTTWO IS SQRT(2.0)
C         RRTPI IS RECIPROCAL OF SQRT(PI)
C         EXPA IS A VALUE SUCH THAT EXP(EXPA) DOES NOT QUITE
C           CAUSE OVERFLOW
C         EXPB IS A VALUE SUCH THAT 1.0 + EXP(-EXPB) MAY BE
C           TAKEN TO BE 1.0
C
      DATA RTTWO, RRTPI, EXPA, EXPB
     *  /1.414213562, 0.5641895835, 80.0, 23.7/
      DATA ZERO, ONE, TWO, THREE /0.0, 1.0, 2.0, 3.0/
      FAULT = .FALSE.
      DO 10 I = 1, 6
   10 C(I) = ZERO
      W = G / D
C
C         TRIAL VALUE OF H
C
      IF (W .GT. EXPA) GOTO 140
      E = EXP(W) + ONE
      R = RTTWO / D
      H = 0.75
      IF (D .LT. THREE) H = 0.25 * D
      K = 1
      GOTO 40
C
C         START OF OUTER LOOP
C
   20 K = K + 1
      IF (K .GT. LIMIT) GOTO 140
      DO 30 I = 1, 6
   30 C(I) = A(I)
C
C         NO CONVERGENCE YET - TRY SMALLER H
```

```
      H = 0.5 * H
   40 T = W
      U = T
      Y = H * H
      X = TWO * Y
      A(1) = ONE / E
      DO 50 I = 2, 6
   50 A(I) = A(I - 1) / E
      V = Y
      F = R * H
      M = 0
C
C         START OF INNER LOOP
C         TO EVALUATE INFINITE SERIES
C
   60 M = M + 1
      IF (M .GT. LIMIT) GOTO 140
      DO 70 I = 1, 6
   70 B(I) = A(I)
      U = U - F
      Z = ONE
      IF (U .GT. -EXPB) Z = EXP(U) + Z
      T = T + F
      L = T .GT. EXPB
      IF (.NOT.L) S = EXP(T) + ONE
      P = EXP(-V)
      Q = P
      DO 90 I = 1, 6
      AA = A(I)
      P = P / Z
      AB = AA
      AA = AA + P
      IF (AA .EQ. AB) GOTO 100
      IF (L) GOTO 80
      Q = Q / S
      AB = AA
      AA = AA + Q
      L = AA .EQ. AB
   80 A(I) = AA
   90 CONTINUE
  100 Y = Y + X
      V = V + Y
      DO 110 I = 1, 6
      IF (A(I) .EQ. ZERO) GOTO 140
      IF (ABS((A(I) - B(I)) / A(I)) .GT. VV) GOTO 60
  110 CONTINUE
C
C         END OF INNER LOOP
C
      V = RRTPI * H
      DO 120 I = 1, 6
  120 A(I) = V * A(I)
      DO 130 I = 1, 6
      IF (A(I) .EQ. ZERO) GOTO 140
      IF (ABS((A(I) - C(I)) / A(I)) .GT. ZZ) GOTO 20
  130 CONTINUE
C
C         END OF OUTER LOOP
C
      RETURN
  140 FAULT = .TRUE.
      RETURN
      END
```