Algorithm AS 70

The Percentage Points of the Normal Distribution

By R. E. ODEH and J. O. EVANS

University of Victoria, Victoria, B.C., Canada

Keywords: INVERSE NORMAL; NORMAL PERCENTAGE POINTS

LANGUAGE

ISO Fortran

DESCRIPTION AND PURPOSE

Given a value of p, this function routine computes the value of x_p for the standard normal distribution. The symbols p and x_p are respectively the lower tail area and its corresponding percentage point. Thus

$$p = \int_{-\infty}^{x_p} (2\pi)^{-\frac{1}{2}} \exp(-t^2/2)\, dt.$$

NUMERICAL METHOD

The subroutine evaluates a rational approximation to $|x_p|$ for $10^{-20} < p < 0{\cdot}5$. The approximation used is of the form

$$x_p = y + S_4(y)/T_4(y),$$

where $S_4(y)$, $T_4(y)$ are polynomials of degree 4, and $y = \sqrt{\{\ln(1/p^2)\}}$. If $0{\cdot}5 < p < 1$, $1-p$ is used in place of p; otherwise the value of x_p is negated. The approximation was determined by first discretizing the interval $[10^{-20}, 0{\cdot}5]$ with 85 values of p. The algorithm described by Evans (1972) was then used to find the minimal degree rational approximation to $x_p - y$ with maximum error less than 10^{-7}. The best approximation was found using the Modified Differential Correction Algorithm. A Fibonacci search was then used to determine the maximum error on the interval.

STRUCTURE

FUNCTION GAUINV (P, IFAULT)

Formal parameters

P	Real	input:	value of lower tail area p
IFAULT	Integer	output:	fault indicator

Failure indications

IFAULT = 0 if $10^{-20} \leqslant p \leqslant 1 - 10^{-20}$.
IFAULT = 1 otherwise.
If *IFAULT* = 1 the value of *GAUINV* is set equal to zero.

TIME

The function evaluation takes 0·002 sec on an IBM 370/145.

Accuracy

The value obtained for x_p is accurate to seven decimal places. The maximum error of approximation is $1{\cdot}5 \times 10^{-8}$. On an IBM 370/145 full accuracy is obtained if the function is evaluated in double precision, and the accuracy of the approximation is reduced to $1{\cdot}8 \times 10^{-6}$ if the function is evaluated in single precision.

Additional Comments

Several algorithms have been published for finding percentage points to greater accuracy but these depend upon evaluation of the Normal Integral. In particular the approximation given here can be used to advantage to replace the Hastings approximation used in Cunningham (1969) or Milton and Hotchkiss (1969).

The approximation of the form $x_p = y + S_2(y)/T_3(y)$ by Hastings (1955) gives a maximum error of $4{\cdot}5 \times 10^{-4}$.

Acknowledgements

The research for this paper was supported by the Defense Research Board of Canada Grant No. 9540–29 (UG). The authors are indebted to the referee for many helpful comments, and to F. W. Mayhood for assistance with the programming.

References

CUNNINGHAM, S. W. (1969). Algorithm AS 24. From normal integral to deviate. *Appl. Statist.*, **18**, 290–293.

EVANS, J. O. (1972). Algorithms for minimal degree polynomial and rational approximation. M.Sc. Thesis, University of Victoria, Victoria, B.C., Canada.

HASTINGS, C. (1955). *Approximations for Digital Computers.* Princeton: University Press.

MILTON, R. C. and HOTCHKISS, R. (1969). Computer evaluation of the normal and inverse normal distribution functions. *Technometrics*, **11**, 817–822.

```
      FUNCTION GAUINV(P, IFAULT)
C
C        ALGORITHM AS 70  APPL. STATIST. (1974) VOL.23, NO.1
C
C        GAUINV FINDS PERCENTAGE POINTS OF THE NORMAL DISTRIBUTION
C
      DATA ZERO, ONE, HALF, ALIMIT /0.0, 1.0, 0.5, 1.0E-20/
C
      DATA                PO,   P1,              P2,                 P3
     *   / -.322232431088, -1.0, -.342242088547, -.204231210245E-1 /
C
      DATA                  P4,             Q0,              Q1
     *   / -.453642210148E-4, .993484626060E-1, .588581570495 /
C
      DATA                Q2,              Q3,              Q4
     *   / .531103462366, .103537752850, .38560700634E-2 /
C
      IFAULT = 1
      GAUINV = ZERO
      PS = P
      IF (PS .GT. HALF) PS = ONE - PS
      IF (PS .LT. ALIMIT) RETURN
      IFAULT = 0
      IF (PS .EQ. HALF) RETURN
      YI = SQRT(ALOG(ONE / (PS * PS)))
      GAUINV = YI + ((((YI * P4 + P3) * YI + P2) * YI + P1) * YI + PO)
     *            / ((((YI * Q4 + Q3) * YI + Q2) * YI + Q1) * YI + QO)
      IF (P .LT. HALF) GAUINV = -GAUINV
      RETURN
      END
```