```
      E(NP) = ZERO
      K = NP * (NP - 1) / 2
      DO 35 J = 1, NP
      K = K + 1
   35 E(NP) = E(NP) + XSSPI(K) * X(J)
C
      A = ZERO
      DO 40 I = 1, NP
   40 A = A + X(I) * E(I)
      A = A + ONE / C
      IF (ZABS(A) .LE. EPS) RETURN
      IFAULT = 0
C
      K = 0
      DO 70 I = 1, NP
      DO 70 J = 1, I
      K = K + 1
      XSSPI(K) = XSSPI(K) - E(I) * E(J) / A
   70 CONTINUE
      RETURN
      END
```

Algorithm AS 241

The Percentage Points of the Normal Distribution

By Michael J. Wichura†

University of Chicago, USA

[Received September 1987. Revised April 1988]

Keywords: Inverse normal; Normal percentage points

Language

Fortran 77

Description and Purpose

Two function routines are given to compute the percentage point z_p of the standard normal distribution corresponding to a prescribed value p for the lower tail area; the relation between p and z_p is

$$p = \int_{-\infty}^{z_p} (2\pi)^{-1/2} \exp(-\zeta^2/2)\,\mathrm{d}\zeta \equiv \Phi(z_p), \qquad z_p = \Phi^{-1}(p).$$

The first routine, PPND7, is accurate to about seven figures (decimal) for $10^{-316} < \min(p, 1-p)$. The second routine, PPND16, is accurate to about 16 figures over the same range.

† *Address for correspondence*: Department of Statistics, University of Chicago, 5734 University Avenue, Chicago, IL 60637, USA.

Numerical Method

The routines set $q = p - 0.5$ and compare q with 0.425. If $|q| \leqslant 0.425$, z_p is obtained as

$$z_p = qA(0.425^2 - q^2)/B(0.425^2 - q^2) \equiv qR_1(q^2),$$

$R_1(t)$ being a minimax rational approximation to $\Phi^{-1}(0.5 + \sqrt{t})/\sqrt{t}$ for $0 \leqslant t \leqslant 0.425^2$. The polynomials A and B are of degree 3 for PPND7 and of degree 7 for PPND16. However, if $|q| > 0.425$, an auxiliary variable $r = \{-\log[\min(p, 1-p)]\}^{1/2}$ is first formed, and z_p is obtained as

$$z_p = \pm C(r - 1.6)/D(r - 1.6) \equiv \pm R_2(r)$$

if $r \leqslant 5$, and as

$$z_p = \pm E(r - 5)/F(r - 5) \equiv \pm R_3(r)$$

if $r > 5$; in each case the sign is taken to be that of q. $R_2(t)$ and $R_3(t)$ are minimax rational approximations to $-\Phi^{-1}(\exp(-t^2))$ over the ranges $1.6 \leqslant t \leqslant 5$ and $5 \leqslant t \leqslant 27$ respectively. For PPND7, the polynomials C and E are of degree 3, while D and F are of degree 2. For PPND16, C, D, E and F are of degree 7. Evaluation of the polynomials A–F involves the addition and multiplication only of positive values; this enhances the numerical stability of the routines.

Related Algorithms

The present algorithms are similar to algorithm PPND of AS 111 (Beasley and Springer, 1977). Whereas that algorithm is accurate to between seven and nine figures for $|z_p| < 3.5$, its performance deteriorates in the tails of the distribution. For example, its accuracy drops to six figures at $z_p = -4$ ($p \approx 3 \times 10^{-5}$), to five figures at $z_p = -5.5$ ($p \approx 2 \times 10^{-8}$) and to four figures at $z_p = 9.5$ ($p \approx 10^{-21}$). As noted later, PPND7 runs as fast as PPND. PPND16 is from $1\frac{1}{3}$ to $1\frac{2}{3}$ times slower than PPND, but produces two to four times as many significant digits. Other algorithms for evaluating z_p are discussed in Kennedy and Gentle (1980).

Structure

REAL FUNCTION PPND7 (P, IFAULT)
REAL FUNCTION PPND16 (P, IFAULT)

Formal parameters

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

Failure indications

IFAULT = 1 if $p \leqslant 0$ or $p \geqslant 1$; *IFAULT* = 0 otherwise.
If *IFAULT* = 1, *PPND7* and *PPND16* return a value of zero.

Time

The present algorithms and PPND were evaluated on a Sun 2 (24-bit mantissa) with the results shown in Table 1. (Whereas 0.425 is the (first) break point in PPND7 and PPND16, the corresponding break point in PPND is 0.42.)

Accuracy

Tables 2 and 3 assess the relative error in the computed value of z_p corresponding to a given input value for p satisfying $\min(p, 1-p) > 10^{-316}$. In the *a priori* bounds θ is the relative error in the computed value of $r = \{-\log[\min(p, 1-p)]\}^{1/2}$, and δ bounds the relative error in a floating point add, multiply or divide. For a computer which has base b and mantissa of fixed length m, and which carries out each arithmetic operation internally in double precision and rounds (or chops) the result to m significant digits, the user may take $\delta = b^{-(m-1)}/2$ (or $\delta = b^{-(m-1)}$). For each of the indicated ranges, the Monte Carlo results are based on 50 000 pseudorandom values of p uniformly distributed on a logit scale; p was constrained to the range $(10^{-35}, 1-10^{-5})$ for PPND7 and to $(10^{-70}, 1-10^{-15})$ for PPND16. PPND7 was evaluated on a Sun 2 with $b = 2$ and $m = 24$, PPND16 on an IBM 3081 with $b = 16$ and $m = 14$.

TABLE 1
Average time for one evaluation

Precision	*Range*	*PPND7 (ms)*	*PPND16 (ms)*	*PPND (ms)*
Single	$\lvert p-0.5\rvert < 0.42$	0.56		0.59
	$\lvert p-0.5\rvert > 0.425$	0.82		0.76
Double	$\lvert p-0.5\rvert < 0.42$		1.62	0.99
	$\lvert p-0.5\rvert > 0.425$		3.68	2.81

TABLE 2
Magnitude of the relative error in z_p for PPND7

Range	*A priori upper bound*	*Monte Carlo Maximum*	*Monte Carlo Root mean square*
$\lvert p-0.5\rvert \leqslant 0.425$	$11.3\delta + 4.6 \times 10^{-8}$	2.9×10^{-7}	8.0×10^{-8}
$\lvert p-0.5\rvert > 0.425$	$12.4\delta + 2\lvert\theta\rvert + 1.2 \times 10^{-7}$	3.5×10^{-7}	1.1×10^{-7}

TABLE 3
Magnitude of the relative error in z_p for PPND16

Range	*A priori upper bound*	*Monte Carlo Maximum*	*Monte Carlo Root mean square*
$\lvert p-0.5\rvert \leqslant 0.425$	$18\delta + 7.4 \times 10^{-17}$	6.0×10^{-16}	1.8×10^{-16}
$\lvert p-0.5\rvert > 0.425$	$22\delta + 2\lvert\theta\rvert + 2.9 \times 10^{-17}$	5.8×10^{-16}	1.6×10^{-16}

Error Propagation

The effect of a perturbation in p can be assessed as follows. Suppose that $p^* = p + \Delta p$ with Δp small. Set $\Delta z = z_{p^*} - z_p$ and write $\phi(z)$ for the normal density at $z = z_p$. Then

$$|\Delta z| \approx \frac{|\Delta p|}{\phi(z)}$$

and

$$\left|\frac{\Delta z}{z}\right| \approx \begin{cases} \left|\dfrac{\Phi(z)}{\phi(z)z}\,\dfrac{\Delta p}{p}\right| \leqslant \dfrac{1}{z^2}\,\dfrac{|\Delta p|}{p}, & \text{if } p < 0.5; \\[2ex] \left|\dfrac{1-\Phi(z)}{\phi(z)z}\,\dfrac{\Delta(1-p)}{1-p}\right| \leqslant \dfrac{1}{z^2}\,\dfrac{|\Delta(1-p)|}{1-p}, & \text{if } p > 0.5. \end{cases}$$

In particular the absolute error in z is never more than 8/3 times as large as the absolute error in p for $|z| \leqslant 0.36$, while the relative error in z is never more than 8/3 times as large as the relative error in $\min(p, 1-p)$ for $|z| \geqslant 0.36$.

Test Data

The following values may be used in checking whether the algorithms have been correctly implemented:

$$z_{0.25} = -0.674\,489\,750\,196\,081\,7,$$

$$z_{0.001} = -3.090\,232\,306\,167\,814,$$

$$z_{10^{-20}} = -9.262\,340\,089\,798\,408.$$

Precision

For double precision:

(a) change REAL to DOUBLE PRECISION on both the FUNCTION statements and the declaration of variables;
(b) change $E0$ to $D0$ in the PARAMETER statements.

(On a machine that uses only 32 bits to represent real variables, PPND16 should be implemented in double precision.)

Additional Comments

If p is very close to unity, a serious loss of significance may be incurred in forming $1-p \equiv c$. In this circumstance the user should, if possible, evaluate c directly (i.e. not by subtracting p from unity) and evaluate z_p as $-z_c$.

The coefficients used in algorithms PPND7 and PPND16 were taken from Wichura (1987), who gives similar sets of coefficients for rational approximations to Φ^{-1} with minimax errors ranging down to 10^{-22}.

Acknowledgement

This paper was prepared using computer facilities supported in part by National

Science Foundation grant DMS-8404941 to the Department of Statistics at the University of Chicago.

References

Beasley, J. D. and Springer, S. G. (1977) The percentage points of the normal distribution. *Appl. Statist.*, **26**, 118–121.
Kennedy, W. J., Jr, and Gentle, J. E. (1980) *Statistical Computing*. New York: Dekker.
Wichura, M. J. (1987) Minimax rational approximations to the percentage points of the normal distribution. *Technical Report 222*. Department of Statistics, University of Chicago.

```
      REAL FUNCTION PPND7 (P, IFAULT)
C
C        ALGORITHM AS241   APPL. STATIST. (1988) VOL. 37, NO. 3
C
C       PRODUCES THE NORMAL DEVIATE  Z  CORRESPONDING TO A GIVEN LOWER
C       TAIL AREA OF  P;  Z  IS ACCURATE TO ABOUT  1  PART IN 10**7.
C
C       THE HASH SUMS BELOW ARE THE SUMS OF THE MANTISSAS OF THE
C       COEFFICIENTS. THEY ARE INCLUDED FOR USE IN CHECKING
C       TRANSCRIPTION.
C
      REAL ZERO, ONE, HALF, SPLIT1, SPLIT2, CONST1, CONST2,
     *  A0, A1, A2, A3, B1, B2, B3, C0, C1, C2, C3, D1, D2,
     *  E0, E1, E2, E3, F1, F2, P, Q, R
      PARAMETER (ZERO = 0.0E0, ONE = 1.0E0, HALF = ONE/2.0E0,
     *  SPLIT1 = 0.425E0,    SPLIT2 = 5.0E0,
     *  CONST1 = 0.180625E0, CONST2 = 1.6E0)
C
C     COEFFICIENTS FOR  P  CLOSE TO  1/2
      PARAMETER (A0 = 3.38713 27179E0,
     *           A1 = 5.04342 71938E1,
     *           A2 = 1.59291 13202E2,
     *           A3 = 5.91093 74720E1,
     *           B1 = 1.78951 69469E1,
     *           B2 = 7.87577 57664E1,
     *           B3 = 6.71875 63600E1)
C     HASH SUM AB    32.31845 77772
C
C     COEFFICIENTS FOR  P  NEITHER CLOSE TO  1/2  NOR  0 OR 1
      PARAMETER (C0 = 1.42343 72777E0,
     *           C1 = 2.75681 53900E0,
     *           C2 = 1.30672 84816E0,
     *           C3 = 1.70238 21103E-1,
     *           D1 = 7.37001 64250E-1,
     *           D2 = 1.20211 32975E-1)
C     HASH SUM CD    15.76149 29821
C
C     COEFFICIENTS FOR  P  NEAR  0 OR 1
      PARAMETER (E0 = 6.65790 51150E0,
     *           E1 = 3.08122 63860E0,
     *           E2 = 4.28682 94337E-1,
     *           E3 = 1.73372 03997E-2,
     *           F1 = 2.41978 94225E-1,
     *           F2 = 1.22582 02635E-2)
C     HASH SUM EF    19.40529 10204
C
      IFAULT = 0
      Q = P - HALF
      IF (ABS(Q) .LE. SPLIT1) THEN
         R = CONST1 - Q * Q
         PPND7 = Q * (((A3 * R + A2) * R + A1) * R + A0) /
     *               (((B3 * R + B2) * R + B1) * R + ONE)
```

```
         RETURN
      ELSE
         IF (Q .LT. 0) THEN
            R = P
         ELSE
            R = ONE - P
         ENDIF
         IF (R .LE. ZERO) THEN
            IFAULT = 1
            PPND7 = ZERO
            RETURN
         ENDIF
         R = SQRT(-LOG(R))
         IF (R .LE. SPLIT2) THEN
            R = R - CONST2
            PPND7 = (((C3 * R + C2) * R + C1) * R + C0) /
     *               ((D2 * R + D1) * R + ONE)
         ELSE
            R = R - SPLIT2
            PPND7 = (((E3 * R + E2) * R + E1) * R + E0) /
     *               ((F2 * R + F1) * R + ONE)
         ENDIF
         IF (Q .LT. 0) PPND7 = -PPND7
         RETURN
      ENDIF
      END
      REAL FUNCTION PPND16 (P, IFAULT)
C
C        ALGORITHM AS241    APPL. STATIST. (1988) VOL. 37, NO. 3
C
C       PRODUCES THE NORMAL DEVIATE  Z  CORRESPONDING TO A GIVEN LOWER
C       TAIL AREA OF  P;  Z  IS ACCURATE TO ABOUT  1  PART IN 10**16.
C
C       THE HASH SUMS BELOW ARE THE SUMS OF THE MANTISSAS OF THE
C       COEFFICIENTS. THEY ARE INCLUDED FOR USE IN CHECKING
C       TRANSCRIPTION.
C
      REAL ZERO, ONE, HALF, SPLIT1, SPLIT2, CONST1, CONST2,
     *  A0, A1, A2, A3, A4, A5, A6, A7, B1, B2, B3, B4, B5, B6, B7,
     *  C0, C1, C2, C3, C4, C5, C6, C7, D1, D2, D3, D4, D5, D6, D7,
     *  E0, E1, E2, E3, E4, E5, E6, E7, F1, F2, F3, F4, F5, F6, F7,
     *  P, Q, R
C
      PARAMETER (ZERO = 0.0E0, ONE = 1.0E0, HALF = ONE/2.0E0,
     *  SPLIT1 = 0.425E0,    SPLIT2 = 5.0E0,
     *  CONST1 = 0.180625E0, CONST2 = 1.6E0)
C
C     COEFFICIENTS FOR  P  CLOSE TO  1/2
      PARAMETER (A0 = 3.38713 28727 96366 6080E0,
     *           A1 = 1.33141 66789 17843 7745E2,
     *           A2 = 1.97159 09503 06551 4427E3,
     *           A3 = 1.37316 93765 50946 1125E4,
     *           A4 = 4.59219 53931 54987 1457E4,
     *           A5 = 6.72657 70927 00870 0853E4,
     *           A6 = 3.34305 75583 58812 8105E4,
     *           A7 = 2.50908 09287 30122 6727E3,
     *           B1 = 4.23133 30701 60091 1252E1,
     *           B2 = 6.87187 00749 20579 0830E2,
     *           B3 = 5.39419 60214 24751 1077E3,
     *           B4 = 2.12137 94301 58659 5867E4,
     *           B5 = 3.93078 95800 09271 0610E4,
     *           B6 = 2.87290 85735 72194 2674E4,
     *           B7 = 5.22649 52788 52854 5610E3)
C     HASH SUM AB    55.88319 28806 14901 4439
C
C     COEFFICIENTS FOR  P  NEITHER CLOSE TO  1/2  NOR  0 OR 1
      PARAMETER (C0 = 1.42343 71107 49683 57734E0,
```

```
     *          C1 = 4.63033 78461 56545 29590E0,
     *          C2 = 5.76949 72214 60691 40550E0,
     *          C3 = 3.64784 83247 63204 60504E0,
     *          C4 = 1.27045 82524 52368 38258E0,
     *          C5 = 2.41780 72517 74506 11770E-1,
     *          C6 = 2.27238 44989 26918 45833E-2,
     *          C7 = 7.74545 01427 83414 07640E-4,
     *          D1 = 2.05319 16266 37758 82187E0,
     *          D2 = 1.67638 48301 83803 84940E0,
     *          D3 = 6.89767 33498 51000 04550E-1,
     *          D4 = 1.48103 97642 74800 74590E-1,
     *          D5 = 1.51986 66563 61645 71966E-2,
     *          D6 = 5.47593 80849 95344 94600E-4,
     *          D7 = 1.05075 00716 44416 84324E-9)
C     HASH SUM CD    49.33206 50330 16102 89036
C
C     COEFFICIENTS FOR  P  NEAR  0 OR 1
      PARAMETER (E0 = 6.65790 46435 01103 77720E0,
     *          E1 = 5.46378 49111 64114 36990E0,
     *          E2 = 1.78482 65399 17291 33580E0,
     *          E3 = 2.96560 57182 85048 91230E-1,
     *          E4 = 2.65321 89526 57612 30930E-2,
     *          E5 = 1.24266 09473 88078 43860E-3,
     *          E6 = 2.71155 55687 43487 57815E-5,
     *          E7 = 2.01033 43992 92288 13265E-7,
     *          F1 = 5.99832 20655 58879 37690E-1,
     *          F2 = 1.36929 88092 27358 05310E-1,
     *          F3 = 1.48753 61290 85061 48525E-2,
     *          F4 = 7.86869 13114 56132 59100E-4,
     *          F5 = 1.84631 83175 10054 68180E-5,
     *          F6 = 1.42151 17583 16445 88870E-7,
     *          F7 = 2.04426 31033 89939 78564E-15)
C     HASH SUM EF    47.52583 31754 92896 71629
C
      IFAULT = 0
      Q = P - HALF
      IF (ABS(Q) .LE. SPLIT1) THEN
         R = CONST1 - Q * Q
         PPND16 = Q * (((((((A7 * R + A6) * R + A5) * R + A4) * R + A3)
     *     * R + A2) * R + A1) * R + A0) / (((((((B7 * R + B6) * R + B5)
     *     * R + B4) * R + B3) * R + B2) * R + B1) * R + ONE)
         RETURN
      ELSE
         IF (Q .LT. 0) THEN
            R = P
         ELSE
            R = ONE - P
         ENDIF
         IF (R .LE. ZERO) THEN
            IFAULT = 1
            PPND16 = ZERO
            RETURN
         ENDIF
         R = SQRT(-LOG(R))
         IF (R .LE. SPLIT2) THEN
            R = R - CONST2
            PPND16 = (((((((C7 * R + C6) * R + C5) * R + C4) * R
     *         + C3) * R + C2) * R + C1) * R + C0) / (((((((D7 * R
     *         + D6) * R + D5) * R + D4) * R + D3) * R + D2) * R
     *         + D1) * R + ONE)
         ELSE
            R = R - SPLIT2
            PPND16 = (((((((E7 * R + E6) * R + E5) * R + E4) * R
     *         + E3) * R + E2) * R + E1) * R + E0) / (((((((F7 * R
     *         + F6) * R + F5) * R + F4) * R + F3) * R + F2) * R
     *         + F1) * R + ONE)
         ENDIF
```

```
    IF (Q .LT. 0) PPND16 = -PPND16
    RETURN
 ENDIF
 END
```

Remark AS R75

Some Remarks on Algorithm AS 164: Least Squares Subject to Linear Constraints

By William M. Sallas†

IMSL, Houston, USA

[Received July 1987. Revised March 1988]

This paper addresses three items concerning the linear least squares algorithm with linear equality constraints AS 164 (Stirling, 1981). The following items will be discussed in sequence:

(a) a correction to subroutine GIVENC so that rounding errors that can arise from incorporating linear constraints do not lead to totally incorrect results;
(b) an addition to subroutine ALIAS so that the R matrix computed corresponds to a reduced model formed after *all* constraints have been incorporated in the full (unconstrained) model;
(c) a modification to subroutine ALIAS to perform an improved check for linearly dependent regressors (Healy, 1968).

First, subroutine GIVENC can return totally incorrect results because of rounding errors that occur when constraints are incorporated into the R matrix. The incorrect results occur frequently for linearly dependent constraints. Also, incorrect results can occur for linearly independent constraints. These problems occur regardless of the settings of small constants EPS0 and EPS1 that are suggested for remedying these problems.

The problem is that GIVENC needs to determine whether any element of a new constraint that is reduced by the previous constraints in R should be regarded as zero. For example, a new constraint that is linearly dependent on previous constraints is usually not zeroed out exactly due to rounding, and the remaining garbage is treated incorrectly as an additional linearly independent constraint. Even for linearly independent constraints, a small number as the leading non-zero element in a reduced constraint causes a totally different constraint to be incorporated in the fit. For example, the following two linearly independent constraints

$$3.3\beta_1 + 6.6\beta_2 + 3.3\beta_3 + 6.6\beta_4 = 3.3$$

$$1.65\beta_1 + 3.3\beta_2 + 1.65\beta_3 + 6.6\beta_4 = 6.6$$

are reduced using the original version of GIVENC (on a computer with a machine epsilon in single precision approximately equal to 10^{-6}) to

$$\beta_1 = 1.000\,00 - 2.000\,00\beta_2 - 1.000\,00\beta_3 - 2.000\,00\beta_4$$

$$\beta_2 = 173\,015\,1 - 0.333\,33\beta_3 - 115\,343\,5\beta_4$$

† *Address for correspondence*: IMSL, Park West Tower One, 2500 CityWest Boulevard, Houston, TX 77042, USA.