plot

Normal stats R Documentation

The Normal Distribution

Description

Description
Density, distribution function, quantile function and random generation for the normal distribution with mean equal to mean and standard deviation equal to sd.
Usage
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)

plot

Normal stats R Documentation

The Normal Distribution

Arguments
x, q vector of quantiles.
p vector of probabilities.
n number of observations. If length(n) ¿ 1, the length is taken to be the number required.
mean vector of means.
sd vector of standard deviations.
log, log.p logical; if TRUE, probabilities p are given as log(p).
lower.tail logical; if TRUE (default), probabilities are P[X x] otherwise, P[X ¿ x].

plot

Normal stats R Documentation

The Normal Distribution

Details

If mean or sd are not specified they assume the default values of 0 and 1, respectively.

The normal distribution has density

$$If(x) = 1/((2)?)e^{-}((x-?)^{2}/(2?^{2}))$$

where ? is the mean of the distribution and ? the standard deviation.

The Normal Distribution

Value

dnorm gives the density, pnorm gives the distribution function, qnorm gives the quantile function, and rnorm generates random deviates.

The length of the result is determined by n for rnorm, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than n are recycled to the length of the result. Only the first elements of the logical arguments are used.

For sd = 0 this gives the limit as sd decreases to 0, a point mass at mu. sd ¡ 0 is an error and returns NaN.

Source

For pnorm, based on

Cody, W. D. (1993) Algorithm 715: SPECFUN - A portable FORTRAN package of special function routines and test drivers. ACM Transactions on Mathematical Software 19, 22-32.

For qnorm, the code is a C translation of

Wichura, M. J. (1988) Algorithm AS 241: The percentage points of the normal distribution. Applied Statistics, 37, 477484.

which provides precise results up to about 16 digits.

For rnorm, see RNG for how to select the algorithm and for references to the supplied methods.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) The New S Language. Wadsworth & Brooks/Cole

The Normal Distribution

Examples

```
require(graphics)

dnorm(0) == 1/sqrt(2*pi)
dnorm(1) == exp(-1/2)/sqrt(2*pi)
dnorm(1) == 1/sqrt(2*pi*exp(1))

## Using ”log = TRUE”for an extended range :
par(mfrow = c(2,1))
plot(function(x) dnorm(x, log = TRUE), -60, 50, main = ”log  Normal density
”)
curve(log(dnorm(x)), add = TRUE, col = red”, lwd = 2)
mtext(”dnorm(x, log=TRUE)”, adj = 0)
mtext(”log(dnorm(x))”, col = red”, adj = 1)
```

The Normal Distribution

```
plot(function(x) pnorm(x, log.p = TRUE), -50, 10, main = ”log  Normal
Cumulative ”)
curve(log(pnorm(x)), add = TRUE, col = red”, lwd = 2)
mtext(”pnorm(x, log=TRUE)”, adj = 0)
mtext(”log(pnorm(x))”, col = red”, adj = 1)
par(mfrow = c(2,1))
## if you want the so-called 'error function'
erf ¡- function(x) 2 * pnorm(x * sqrt(2)) - 1
## (see Abramowitz and Stegun 29.2.29)
## and the so-called 'complementary error function'
erfc ¡- function(x) 2 * pnorm(x * sqrt(2), lower = FALSE)
## and the inverses
erfinv ¡- function (x) qnorm((1 + x)/2)/sqrt(2)
erfcinv ¡- function (x) qnorm(x/2, lower = FALSE)/sqrt(2)
```

The Normal Distribution

Examples

```
require(graphics)
## Using ”log = TRUE”for an extended range :
par(mfrow = c(2,1))
plot(function(x) dnorm(x, log = TRUE), -60, 50, main = ”log  Normal density
”)
curve(log(dnorm(x)), add = TRUE, col = red”, lwd = 2)
mtext(”dnorm(x, log=TRUE)”, adj = 0)
mtext(”log(dnorm(x))”, col = red”, adj = 1)
plot(function(x) pnorm(x, log.p = TRUE), -50, 10, main = ”log  Normal
Cumulative ”)
curve(log(pnorm(x)), add = TRUE, col = red”, lwd = 2)
mtext(”pnorm(x, log=TRUE)”, adj = 0)
mtext(”log(pnorm(x))”, col = red”, adj = 1)
## if you want the so-called 'error function'
erf ¡- function(x) 2 * pnorm(x * sqrt(2)) - 1
## (see Abramowitz and Stegun 29.2.29)
## and the so-called 'complementary error function'
erfc ¡- function(x) 2 * pnorm(x * sqrt(2), lower = FALSE)
## and the inverses
erfinv ¡- function (x) qnorm((1 + x)/2)/sqrt(2)
erfcinv ¡- function (x) qnorm(x/2, lower = FALSE)/sqrt(2)
```