normal() — Cumulatives, reverse cumulatives, and densities

Description

The below functions return density values, cumulatives, reverse cumulatives, inverse cumulatives, and in one case, derivatives of the indicated probability density function. These functions mirror the Stata functions of the same name and in fact are the Stata functions.

See [FN] **Statistical functions** for details. In the syntax diagram above, some arguments have been renamed in hope of aiding understanding, but the function arguments match one to one with the underlying Stata functions.

Syntax

Gaussian normal

d = `normalden(`z`)`

d = `normalden(`x, sd`)`

d = `normalden(`x, $mean$, sd`)`

p = `normal(`z`)`

z = `invnormal(`p`)`

$ln(d)$ = `lnnormalden(`z`)`

$ln(d)$ = `lnnormalden(`x, sd`)`

$ln(d)$ = `lnnormalden(`x, $mean$, sd`)`

$ln(p)$ = `lnnormal(`z`)`

Binormal

p = `binormal(`z_1, z_2, rho`)`

Multivariate normal

$ln(d)$ = `lnmvnormalden(`M, V, X`)`

Beta

d = `betaden(`a, b, x`)`

p = `ibeta(`a, b, x`)`

q = `ibetatail(`a, b, x`)`

x = `invibeta(`a, b, p`)`

x = `invibetatail(`a, b, q`)`

Binomial

pk = `binomialp(`n, k, pi`)`
p = `binomial(`n, k, pi`)`
q = `binomialtail(`n, k, pi`)`
pi = `invbinomial(`n, k, p`)`
pi = `invbinomialtail(`n, k, q`)`

Cauchy

d = `cauchyden(`a, b, x`)`
p = `cauchy(`a, b, x`)`
q = `cauchytail(`a, b, x`)`
x = `invcauchy(`a, b, p`)`
x = `invcauchytail(`a, b, q`)`
$ln(d)$ = `lncauchyden(`a, b, x`)`

χ^2

d = `chi2den(`df, x`)`
p = `chi2(`df, x`)`
q = `chi2tail(`df, x`)`
x = `invchi2(`df, p`)`
x = `invchi2tail(`df, q`)`

Dunnett's multiple range

p = `dunnettprob(`k, df, x`)`
x = `invdunnettprob(`k, df, p`)`

Exponential

d = `exponentialden(`b, x`)`
p = `exponential(`b, x`)`
q = `exponentialtail(`b, x`)`
x = `invexponential(`b, p`)`
x = `invexponentialtail(`b, q`)`

F

d = `Fden(`df_1, df_2, f`)`
p = `F(`df_1, df_2, f`)`
q = `Ftail(`df_1, df_2, f`)`
f = `invF(`df_1, df_2, p`)`
f = `invFtail(`df_1, df_2, q`)`

Gamma and inverse gamma

d = `gammaden(`a, b, g, x`)`

p = `gammap(`a, x`)`

q = `gammaptail(`a, x`)`

x = `invgammap(`a, p`)`

x = `invgammaptail(`a, q`)`

dg/da = `dgammapda(`a, x`)`

dg/dx = `dgammapdx(`a, x`)`

d2g/da2 = `dgammapdada(`a, x`)`

d2g/dadx = `dgammapdadx(`a, x`)`

d2g/dx2 = `dgammapdxdx(`a, x`)`

ln(d) = `lnigammaden(`a, b, x`)`

Hypergeometric

pk = `hypergeometricp(`N, K, n, k`)`

p = `hypergeometric(`N, K, n, k`)`

Inverse Gaussian

d = `igaussianden(`m, a, x`)`

p = `igaussian(`m, a, x`)`

q = `igaussiantail(`m, a, x`)`

x = `invigaussian(`m, a, p`)`

x = `invigaussiantail(`m, a, q`)`

ln(d) = `lnigaussianden(`m, a, x`)`

Laplace

d = `laplaceden(`m, b, x`)`

p = `laplace(`m, b, x`)`

q = `laplacetail(`m, b, x`)`

x = `invlaplace(`m, b, p`)`

x = `invlaplacetail(`m, b, q`)`

ln(d) = `lnlaplaceden(`m, b, x`)`

Logistic

d = `logisticden(`x`)`
d = `logisticden(`s, x`)`
d = `logisticden(`m, s, x`)`
p = `logistic(`x`)`
p = `logistic(`s, x`)`
p = `logistic(`m, s, x`)`
q = `logistictail(`x`)`
q = `logistictail(`s, x`)`
q = `logistictail(`m, s, x`)`
x = `invlogistic(`p`)`
x = `invlogistic(`s, p`)`
x = `invlogistic(`m, s, p`)`
x = `invlogistictail(`q`)`
x = `invlogistictail(`s, q`)`
x = `invlogistictail(`m, s, q`)`

Negative binomial

pk = `nbinomialp(`n, k, pi`)`
p = `nbinomial(`n, k, pi`)`
q = `nbinomialtail(`n, k, pi`)`
pi = `invnbinomial(`n, k, p`)`
pi = `invnbinomialtail(`n, k, q`)`

Noncentral beta

d = `nbetaden(`a, b, np, x`)`
p = `nibeta(`a, b, np, x`)`
x = `invnibeta(`a, b, np, p`)`

Noncentral χ^2

d = `nchi2den(`df, np, x`)`
p = `nchi2(`df, np, x`)`
q = `nchi2tail(`df, np, x`)`
x = `invnchi2(`df, np, p`)`
x = `invnchi2tail(`df, np, q`)`
np = `npnchi2(`df, x, p`)`

Noncentral F

d = `nFden(`df_1, df_2, np, f`)`

p = `nF(`df_1, df_2, np, f`)`

q = `nFtail(`df_1, df_2, np, f`)`

f = `invnF(`df_1, df_2, np, p`)`

f = `invnFtail(`df_1, df_2, np, q`)`

np = `npnF(`df_1, df_2, f, p`)`

Noncentral Student's t

d = `ntden(`df, np, t`)`

p = `nt(`df, np, t`)`

q = `nttail(`df, np, t`)`

t = `invnt(`df, np, p`)`

t = `invnttail(`df, np, q`)`

np = `npnt(`df, t, p`)`

Poisson

pk = `poissonp(`$mean$, k`)`

p = `poisson(`$mean$, k`)`

q = `poissontail(`$mean$, k`)`

m = `invpoisson(`k, p`)`

m = `invpoissontail(`k, q`)`

Student's t

d = `tden(`df, t`)`

p = `t(`df, t`)`

q = `ttail(`df, t`)`

t = `invt(`df, p`)`

t = `invttail(`df, q`)`

Tukey's Studentized range

p = `tukeyprob(`k, df, x`)`

x = `invtukeyprob(`k, df, p`)`

Weibull

d = `weibullden(`a, b, x`)`
d = `weibullden(`a, b, g, x`)`
p = `weibull(`a, b, x`)`
p = `weibull(`a, b, g, x`)`
q = `weibulltail(`a, b, x`)`
q = `weibulltail(`a, b, g, x`)`
x = `invweibull(`a, b, p`)`
x = `invweibull(`a, b, g, p`)`
x = `invweibulltail(`a, b, q`)`
x = `invweibulltail(`a, b, g, q`)`

Weibull (proportional hazards)

d = `weibullphden(`a, b, x`)`
d = `weibullphden(`a, b, g, x`)`
p = `weibullph(`a, b, x`)`
p = `weibullph(`a, b, g, x`)`
q = `weibullphtail(`a, b, x`)`
q = `weibullphtail(`a, b, g, x`)`
x = `invweibullph(`a, b, p`)`
x = `invweibullph(`a, b, g, p`)`
x = `invweibullphtail(`a, b, q`)`
x = `invweibullphtail(`a, b, g, q`)`

Wishart and inverse Wishart

ln(d) = `lnwishartden(`*df*, V, X`)`
ln(d) = `lniwishartden(`*df*, V, X`)`

where

1. All functions return real and all arguments are real or real matrices.

2. The left-hand-side notation is used to assist in interpreting the meaning of the returned value:

d	= density value
pk	= probability of discrete outcome $K = \Pr(K = k)$
p	= left cumulative
	$= \Pr(-\text{infinity} < \textit{statistic} \le x)$ (continuous)
	$= \Pr(0 \le K \le k)$ (discrete)
q	= right cumulative
	$= 1 - p$ (continuous)
	$= \Pr(K \ge k) = 1 - p + pk$ (discrete)
np	= noncentrality parameter
ln(p)	= log cumulative
ln(d)	= log density

3. Hypergeometric distribution:

N	= number of objects in the population
K	= number of objects in the population with the characteristic of interest, $K < N$
n	= sample size, $n < N$
k	= number of objects in the sample with the characteristic of interest, $\max(0, n - N + K) \le k \le \min(K, n)$

4. Negative binomial distribution: $n > 0$ and may be nonintegral.

5. Multivariate normal, Wishart, and inverse Wishart distributions:

M	= mean vector
V	= covariance or scale matrix
X	= random vector for multivariate normal or random matrix for Wishart and inverse Wishart

Remarks and examples

stata.com

Remarks are presented under the following headings:

R-conformability

The above functions are usually used with scalar arguments and then return a scalar result:

```
: x = chi2(10, 12)

: x
  .7149434997
```

The arguments may, however, be vectors or matrices. For instance,

```
        : x = chi2((10,11,12), 12)
        : x
                      1              2              3
            +----------------------------------------------+
          1 |  .7149434997    .6363567795    .5543203586   |
            +----------------------------------------------+

        : x = chi2(10, (12,12.5,13))
        : x
                      1              2              3
            +----------------------------------------------+
          1 |  .7149434997    .7470146767    .7763281832   |
            +----------------------------------------------+

        : x = chi2((10,11,12), (12,12.5,13))
        : x
                      1              2              3
            +----------------------------------------------+
          1 |  .7149434997    .6727441644    .6309593164   |
            +----------------------------------------------+
```

In the last example, the numbers correspond to `chi2(10,12)`, `chi2(11,12.5)`, and `chi2(12,13)`.

Arguments are required to be r-conformable (see [M-6] **Glossary**), and thus,

```
        : x = chi2((10\11\12), (12,12.5,13))
        : x
                      1              2              3
            +----------------------------------------------+
          1 |  .7149434997    .7470146767    .7763281832   |
          2 |  .6363567795    .6727441644    .7066745906   |
          3 |  .5543203586     .593595966    .6309593164   |
            +----------------------------------------------+
```

which corresponds to

```
                      1              2              3
            +----------------------------------------------+
          1 |  chi2(10,12) chi2(10,12.5)   chi2(10,13)     |
          2 |  chi2(11,12) chi2(11,12.5)   chi2(11,13)     |
          3 |  chi2(12,12) chi2(12,12.5)   chi2(12,13)     |
            +----------------------------------------------+
```

A note concerning invbinomial() and invbinomialtail()

`invbinomial(`n`, `k`, `p`)` `invbinomialtail(`n`, `k`, `q`)` are useful for calculating confidence intervals for *pi*, the probability of a success. `invbinomial()` returns the probability *pi* such that the probability of observing k or fewer successes in n trials is p. `invbinomialtail()` returns the probability *pi* such that the probability of observing k or more successes in n trials is q.

A note concerning ibeta()

`ibeta(`a`, `b`, `x`)` is known as the cumulative beta distribution, and it is known as the incomplete beta function $I_x(a, b)$.

A note concerning gammap( )

`gammap(`a`, `x`)` is known as the cumulative gamma distribution, and it is known as the incomplete gamma function $P(a, x)$.

Conformability

All functions require that arguments be r-conformable; see *R-conformability* above. Returned is matrix of `max(`argument rows`)` rows and `max(`argument columns`)` columns containing element-by-element calculated results.

Diagnostics

All functions return missing when arguments are out of range.

Also see

[M-5] **mvnormal( )** — Compute multivariate normal distributions and derivatives

[M-4] **Statistical** — Statistical functions