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

\[
\begin{align*}
    d &= \text{normalden}(z) \\
    d &= \text{normalden}(x, sd) \\
    d &= \text{normalden}(x, mean, sd) \\
    p &= \text{normal}(z) \\
    z &= \text{invnormal}(p) \\
    \ln(d) &= \ln\text{normalden}(z) \\
    \ln(d) &= \ln\text{normalden}(x, sd) \\
    \ln(d) &= \ln\text{normalden}(x, mean, sd) \\
    \ln(p) &= \ln\text{normal}(z)
\end{align*}
\]

Binormal

\[
p = \text{binormal}(z_1, z_2, rho)
\]

Multivariate normal

\[
\ln(d) = \ln\text{mvnormalden}(M, V, X)
\]

Beta

\[
\begin{align*}
    d &= \text{betaden}(a, b, x) \\
    p &= \text{ibeta}(a, b, x) \\
    q &= \text{ibetatail}(a, b, x) \\
    x &= \text{invibeta}(a, b, p) \\
    x &= \text{invibetatail}(a, b, q)
\end{align*}
\]
Binomial

\[ pk = \text{binomialp}(n, k, pi) \]
\[ p = \text{binomial}(n, k, pi) \]
\[ q = \text{binomialtail}(n, k, pi) \]
\[ pi = \text{invbinomial}(n, k, p) \]
\[ pi = \text{invbinomialtail}(n, k, q) \]

Cauchy

\[ d = \text{cauchyden}(a, b, x) \]
\[ p = \text{cauchy}(a, b, x) \]
\[ q = \text{cauchytail}(a, b, x) \]
\[ x = \text{invcauchy}(a, b, p) \]
\[ x = \text{invcauchytail}(a, b, q) \]
\[ \ln(d) = \text{lncauchyden}(a, b, x) \]

Chi-squared

\[ d = \text{chi2den}(df, x) \]
\[ p = \text{chi2}(df, x) \]
\[ q = \text{chi2tail}(df, x) \]
\[ x = \text{invchi2}(df, p) \]
\[ x = \text{invchi2tail}(df, q) \]

Dunnett’s multiple range

\[ p = \text{dunnettprob}(k, df, x) \]
\[ x = \text{invdunnettprob}(k, df, p) \]

Exponential

\[ d = \text{exponentialden}(b, x) \]
\[ p = \text{exponential}(b, x) \]
\[ q = \text{exponentialtail}(b, x) \]
\[ x = \text{invexponential}(b, p) \]
\[ x = \text{invexponentialtail}(b, q) \]

F

\[ d = \text{Fden}(df_1, df_2, f) \]
\[ p = \text{F}(df_1, df_2, f) \]
\[ q = \text{Ftail}(df_1, df_2, f) \]
\[ f = \text{invF}(df_1, df_2, p) \]
\[ f = \text{invFtail}(df_1, df_2, q) \]
Gamma and inverse gamma

\[ d = \text{gammaden}(a, b, g, x) \]
\[ p = \text{gammap}(a, x) \]
\[ q = \text{gammaptail}(a, x) \]
\[ x = \text{invgammap}(a, p) \]
\[ x = \text{invgammaptail}(a, q) \]
\[ \frac{dg}{da} = \text{dgammapda}(a, x) \]
\[ \frac{dg}{dx} = \text{dgammapdx}(a, x) \]
\[ \frac{d^2g}{da^2} = \text{dgammapdada}(a, x) \]
\[ \frac{d^2g}{dadx} = \text{dgammapdadx}(a, x) \]
\[ \frac{d^2g}{d^2x} = \text{dgammapdxdx}(a, x) \]
\[ \ln(d) = \lnigammapda(a, b, x) \]

Hypergeometric

\[ pk = \text{hypergeometricp}(N, K, n, k) \]
\[ p = \text{hypergeometric}(N, K, n, k) \]

Inverse Gaussian

\[ d = \text{igaussianden}(m, a, x) \]
\[ p = \text{igaussian}(m, a, x) \]
\[ q = \text{igaussiantail}(m, a, x) \]
\[ x = \text{invgaussian}(m, a, p) \]
\[ x = \text{invgaussiantail}(m, a, q) \]
\[ \ln(d) = \lnigaussianden(m, a, x) \]

Laplace

\[ d = \text{laplaceden}(m, b, x) \]
\[ p = \text{laplace}(m, b, x) \]
\[ q = \text{laplacetail}(m, b, x) \]
\[ x = \text{invlaplace}(m, b, p) \]
\[ x = \text{invlaplacetail}(m, b, q) \]
\[ \ln(d) = \lnlaplaceden(m, b, x) \]
Logistic

\[ d = \text{logisticden}(x) \]
\[ d = \text{logisticden}(s, x) \]
\[ d = \text{logisticden}(m, s, x) \]
\[ p = \text{logistic}(x) \]
\[ p = \text{logistic}(s, x) \]
\[ p = \text{logistic}(m, s, x) \]
\[ q = \text{logistictail}(x) \]
\[ q = \text{logistictail}(s, x) \]
\[ q = \text{logistictail}(m, s, x) \]
\[ x = \text{invlogistic}(p) \]
\[ x = \text{invlogistic}(s, p) \]
\[ x = \text{invlogistic}(m, s, p) \]
\[ x = \text{invlogistictail}(q) \]
\[ x = \text{invlogistictail}(s, q) \]
\[ x = \text{invlogistictail}(m, s, q) \]

Negative binomial

\[ pk = \text{nbinomialp}(n, k, pi) \]
\[ p = \text{nbinomial}(n, k, pi) \]
\[ q = \text{nbinomialtail}(n, k, pi) \]
\[ pi = \text{invnbinomial}(n, k, p) \]
\[ pi = \text{invnbinomialtail}(n, k, q) \]

Noncentral beta

\[ d = \text{nbetaden}(a, b, np, x) \]
\[ p = \text{nibeta}(a, b, np, x) \]
\[ x = \text{invnibeta}(a, b, np, p) \]

Noncentral chi-squared

\[ d = \text{nchi2den}(df, np, x) \]
\[ p = \text{nchi2}(df, np, x) \]
\[ q = \text{nchi2tail}(df, np, x) \]
\[ x = \text{invnchi2}(df, np, p) \]
\[ x = \text{invnchi2tail}(df, np, q) \]
\[ np = \text{nnpchi2}(df, x, p) \]
Noncentral F

\[ d = nF\text{den}(df_1, df_2, np, f) \]
\[ p = nF(df_1, df_2, np, f) \]
\[ q = nF\text{tail}(df_1, df_2, np, f) \]
\[ f = nF^{-1}(df_1, df_2, np, f) \]
\[ f = nF\text{tail}^{-1}(df_1, df_2, np, q) \]
\[ np = npnF(df_1, df_2, f, p) \]

Noncentral Student’s t

\[ d = nt\text{den}(df, np, t) \]
\[ p = nt(df, np, t) \]
\[ q = nt\text{tail}(df, np, t) \]
\[ t = nt^{-1}(df, np, p) \]
\[ t = nt\text{tail}^{-1}(df, np, q) \]
\[ np = npnt(df, t, p) \]

Poisson

\[ pk = \text{poissonp}(mean, k) \]
\[ p = \text{poisson}(mean, k) \]
\[ q = \text{poissons}\text{tail}(mean, k) \]
\[ m = inv\text{poisson}(k, p) \]
\[ m = inv\text{poisson}\text{tail}(k, q) \]

Student’s t

\[ d = t\text{den}(df, t) \]
\[ p = t(df, t) \]
\[ q = tt\text{ail}(df, t) \]
\[ t = invt(df, p) \]
\[ t = invtt\text{ail}(df, q) \]

Tukey’s Studentized range

\[ p = \text{tukeyprob}(k, df, x) \]
\[ x = invtukeyprob(k, df, p) \]
Weibull

\[ d = \text{weibullden}(a, b, x) \]
\[ d = \text{weibullden}(a, b, g, x) \]
\[ p = \text{weibull}(a, b, x) \]
\[ p = \text{weibull}(a, b, g, x) \]
\[ q = \text{weibulltail}(a, b, x) \]
\[ q = \text{weibulltail}(a, b, g, x) \]
\[ x = \text{invweibull}(a, b, p) \]
\[ x = \text{invweibull}(a, b, g, p) \]
\[ x = \text{invweibulltail}(a, b, q) \]
\[ x = \text{invweibulltail}(a, b, g, q) \]

Weibull (proportional hazards)

\[ d = \text{weibullphden}(a, b, x) \]
\[ d = \text{weibullphden}(a, b, g, x) \]
\[ p = \text{weibullph}(a, b, x) \]
\[ p = \text{weibullph}(a, b, g, x) \]
\[ q = \text{weibullphtail}(a, b, x) \]
\[ q = \text{weibullphtail}(a, b, g, x) \]
\[ x = \text{invweibullph}(a, b, p) \]
\[ x = \text{invweibullph}(a, b, g, p) \]
\[ x = \text{invweibullphtail}(a, b, q) \]
\[ x = \text{invweibullphtail}(a, b, g, q) \]

Wishart and inverse Wishart

\[ ln(d) = \text{lnwishartden}(df, V, X) \]
\[ ln(d) = \text{lniwishartden}(df, V, X) \]
normal() — Cumulatives, reverse cumulatives, and densities

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:

   \[ d \quad \text{= density value} \]
   \[ pk \quad \text{= probability of discrete outcome} \quad K = \Pr(K = k) \]
   \[ p \quad \text{= left cumulative} \]
   \[ = \Pr\left(\text{statistic} < \text{infinity} < -\infty < \text{statistic} \leq x\right) \quad \text{(continuous)} \]
   \[ = \Pr(0 \leq K \leq k) \quad \text{(discrete)} \]
   \[ q \quad \text{= right cumulative} \]
   \[ = 1 - p \quad \text{(continuous)} \]
   \[ = \Pr(K \geq k) = 1 - p + pk \quad \text{(discrete)} \]
   \[ np \quad \text{= noncentrality parameter} \]
   \[ ln(p) \quad \text{= log cumulative} \]
   \[ ln(d) \quad \text{= log density} \]

3. Hypergeometric distribution:

   \[ N = \text{number of objects in the population} \]
   \[ K = \text{number of objects in the population with the characteristic of interest,} \quad K < N \]
   \[ n = \text{sample size,} \quad n < N \]
   \[ k = \text{number of objects in the sample with the characteristic of interest,} \]
   \[ \max(0, n - N + K) \leq k \leq \min(K, n) \]

4. Negative binomial distribution: \( n > 0 \) and may be nonintegral.

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

   \[ M = \text{mean vector} \]
   \[ V = \text{covariance or scale matrix} \]
   \[ X = \text{random vector for multivariate normal or} \]
   \[ \text{random matrix for Wishart and inverse Wishart} \]

Remarks and examples

Remarks are presented under the following headings:

- **R-conformability**
- **A note concerning invbinomial() and invbinomiaitl()**
- **A note concerning ibeta( )**
- **A note concerning gammap( )**

R-conformability

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

: \( x = \text{chi2(10, 12)} \)

: \( x \)

.7149434997
The arguments may, however, be vectors or matrices. For instance,

```python
: x = chi2((10, 11, 12), 12)
: x

1   2   3
1 0.7149434997 0.6363567795 0.5543203586

: x = chi2(10, (12, 12.5, 13))
: x

1   2   3
1 0.7149434997 0.7470146767 0.7763281832

: x = chi2((10, 11, 12), (12, 12.5, 13))
: x

1   2   3
1 0.7149434997 0.6727441644 0.6309593164
```

In the last example, the numbers correspond to \( \chi^2(10, 12) \), \( \chi^2(11, 12.5) \), and \( \chi^2(12, 13) \).

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

```python
: x = chi2((10, 11, 12), (12, 12.5, 13))
: x

1   2   3
1 0.7149434997 0.7470146767 0.7763281832
2 0.6363567795 0.6727441644 0.7066745906
3 0.5543203586 0.5935959664 0.6309593164
```

which corresponds to

```
1   2   3
1 \chi^2(10, 12) \chi^2(10, 12.5) \chi^2(10, 13)
2 \chi^2(11, 12) \chi^2(11, 12.5) \chi^2(11, 13)
3 \chi^2(12, 12) \chi^2(12, 12.5) \chi^2(12, 13)
```

### A note concerning invbinomial() and invbinomialtail()

`invbinomial(n, k, p)` and `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 $\max(\text{argument rows})$ rows and $\max(\text{argument columns})$ 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