Package ‘extraDistr’

October 10, 2016

Type Package

Title Additional Univariate and Multivariate Distributions

Version 1.8.2

Date 2016-10-10

Author Tymoteusz Wolodzko

Maintainer Tymoteusz Wolodzko <twolodzko+extraDistr@gmail.com>

Description Density, distribution function, quantile function and random generation for a number of univariate and multivariate distributions. This package implements the following distributions: Bernoulli, beta-binomial, beta-negative binomial, beta prime, Bhattacharjee, Birnbaum-Saunders, bivariate normal, bivariate Poisson, categorical, Dirichlet, Dirichlet-multinomial, discrete Laplace, discrete normal, discrete uniform, discrete Weibull, Frechet, gamma-Poisson, generalized extreme value, Gompertz, generalized Pareto, Gumbel, half-Cauchy, half-normal, half-t, Huber density, inverse chi-squared, inverse-gamma, Kumaraswamy, Laplace, logarithmic, Lomax, multivariate hypergeometric, multinomial, non-standard t, non-standard beta, normal mixture, Poisson mixture, Pareto, power, reparametrized beta, Rayleigh, Skellam, slash, triangular, truncated normal, truncated Poisson, Tukey lambda, Wald, zero-inflated binomial, zero-inflated negative binomial, zero-inflated Poisson.

License GPL-2

URL https://github.com/twolodzko/extraDistr

BugReports https://github.com/twolodzko/extraDistr/issues

LazyData TRUE

LinkingTo Rcpp

Imports Rcpp

Suggests testthat

RoxygenNote 5.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-10-10 08:28:35

R topics documented:

`Bernoulli` *Bernoulli distribution*

Description

Probability mass function, distribution function, quantile function and random generation for the Bernoulli distribution.

Usage

```
dbern(x, prob = 0.5, log = FALSE)

pbern(q, prob = 0.5, lower.tail = TRUE, log.p = FALSE)

qbern(p, prob = 0.5, lower.tail = TRUE, log.p = FALSE)

rbern(n, prob = 0.5)
```

Arguments

`x, q`	vector of quantiles.
`prob`	probability of success; (`0 < prob < 1`).
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

See Also

`Binomial`

Examples

```
prop.table(table(rbern(1e5, 0.5)))
```

BetaBinom *Beta-binomial distribution*

Description

Probability mass function and random generation for the beta-binomial distribution.

Usage

```
dbbinom(x, size, alpha = 1, beta = 1, log = FALSE)

pbbinom(q, size, alpha = 1, beta = 1, lower.tail = TRUE, log.p = FALSE)

rbbinom(n, size, alpha = 1, beta = 1)
```

Arguments

`x, q`	vector of quantiles.
`size`	number of trials (zero or more).
`alpha, beta`	non-negative parameters of the beta distribution.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

If $p \sim \mathrm{Beta}(\alpha, \beta)$ and $X \sim \mathrm{Binomial}(n, p)$, then $X \sim \mathrm{BetaBinomial}(n, \alpha, \beta)$.

Probability mass function

$$f(x) = \binom{n}{x} \frac{\mathrm{B}(x + \alpha, n - x + \beta)}{\mathrm{B}(\alpha, \beta)}$$

Warning: cumulative distribution function is defined as

$$F(x) = \sum_{k=0}^{x} f(k)$$

so it may be slow for large datasets.

See Also

`Beta`, `Binomial`

Examples

```
x <- rbbinom(1e5, 1000, 5, 13)
xx <- 0:1000
hist(x, 100, freq = FALSE)
lines(xx-0.5, dbbinom(xx, 1000, 5, 13), col = "red")
hist(pbbinom(x, 1000, 5, 13))
plot(ecdf(x))
lines(xx, pbbinom(xx, 1000, 5, 13), col = "red", lwd = 2)
```

`BetaNegBinom` *Beta-negative binomial distribution*

Description

Probability mass function and random generation for the beta-negative binomial distribution.

Usage

```
dbnbinom(x, size, alpha = 1, beta = 1, log = FALSE)

pbnbinom(q, size, alpha = 1, beta = 1, lower.tail = TRUE, log.p = FALSE)

rbnbinom(n, size, alpha = 1, beta = 1)
```

Arguments

`x, q`	vector of quantiles.
`size`	number of trials (zero or more).
`alpha, beta`	non-negative parameters of the beta distribution.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

If $p \sim \mathrm{Beta}(\alpha, \beta)$ and $X \sim \mathrm{NegBinomial}(r, p)$, then $X \sim \mathrm{BetaNegBinomial}(r, \alpha, \beta)$.

Probability mass function

$$f(x) = \frac{\Gamma(r+x)}{x!\Gamma(r)} \frac{\mathrm{B}(\alpha + r, \beta + x)}{\mathrm{B}(\alpha, \beta)}$$

Warning: cumulative distribution function is defined as

$$F(x) = \sum_{k=0}^{x} f(k)$$

so it may be slow for large datasets.

See Also

Beta, NegBinomial

Examples

```
x <- rbnbinom(1e5, 1000, 5, 13)
xx <- 0:1e5
hist(x, 100, freq = FALSE)
lines(xx-0.5, dbnbinom(xx, 1000, 5, 13), col = "red")
hist(pbnbinom(x, 1000, 5, 13))
plot(ecdf(x))
lines(xx, pbnbinom(xx, 1000, 5, 13), col = "red", lwd = 2)
```

BetaPrime *Beta prime distribution*

Description

Density, distribution function, quantile function and random generation for the beta prime distribution.

Usage

```
dbetapr(x, shape1, shape2, scale = 1, log = FALSE)

pbetapr(q, shape1, shape2, scale = 1, lower.tail = TRUE, log.p = FALSE)

qbetapr(p, shape1, shape2, scale = 1, lower.tail = TRUE, log.p = FALSE)

rbetapr(n, shape1, shape2, scale = 1)
```

Arguments

`x, q`	vector of quantiles.
`shape1, shape2`	non-negative parameters.
`scale`	positive valued scale parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

If $X \sim \mathrm{Beta}(\alpha, \beta)$, then $\frac{X}{1-X} \sim \mathrm{BetaPrime}(\alpha, \beta)$.

Probability density function

$$f(x) = \frac{(x/\sigma)^{\alpha-1}(1+x/\sigma)^{-\alpha-\beta}}{B(\alpha,\beta)\sigma}$$

Cumulative distribution function

$$F(x) = I_{\frac{x/\sigma}{1+x/\sigma}}(\alpha, \beta)$$

See Also

Beta

Examples

```
x <- rbetapr(1e5, 5, 3, 2)
xx <- seq(0, 100, by = 0.1)
hist(x, 350, freq = FALSE, xlim = c(0, 100))
lines(xx, dbetapr(xx, 5, 3, 2), col = "red")
hist(pbetapr(x, 5, 3, 2))
plot(ecdf(x), xlim = c(0, 100))
lines(xx, pbetapr(xx, 5, 3, 2), col = "red")
```

Bhattacharjee *Bhattacharjee distribution*

Description

Density, distribution function, and random generation for the Bhattacharjee distribution.

Usage

```
dbhatt(x, mu = 0, sigma = 1, a = sigma, log = FALSE)

pbhatt(q, mu = 0, sigma = 1, a = sigma, lower.tail = TRUE,
  log.p = FALSE)

rbhatt(n, mu = 0, sigma = 1, a = sigma)
```

Arguments

`x, q`	vector of quantiles.
`mu, sigma, a`	location, scale and shape parameters. Scale and shape must be positive.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

If $Z \sim \mathrm{Normal}(0, 1)$ and $U \sim \mathrm{Uniform}(0, 1)$, then $Z + U$ follows Bhattacharjee distribution.

Probability density function

$$f(z) = \frac{1}{2a}\left[\Phi\left(\frac{x-\mu+a}{\sigma}\right) - \Phi\left(\frac{x-\mu-a}{\sigma}\right)\right]$$

Cumulative distribution function

$$F(z) = \frac{\sigma}{2a}\left[(x-\mu)\Phi\left(\frac{x-\mu+a}{\sigma}\right) - (x-\mu)\Phi\left(\frac{x-\mu-a}{\sigma}\right) + \phi\left(\frac{x-\mu+a}{\sigma}\right) - \phi\left(\frac{x-\mu-a}{\sigma}\right)\right]$$

References

Bhattacharjee, G.P., Pandit, S.N.N., and Mohan, R. (1963). Dimensional chains involving rectangular and normal error-distributions. Technometrics, 5, 404-406.

Examples

```
x <- rbhatt(1e5, 5, 3, 5)
xx <- seq(-20, 20, by = 0.01)
hist(x, 100, freq = FALSE)
lines(xx, dbhatt(xx, 5, 3, 5), col = "red")
hist(pbhatt(x, 5, 3, 5))
plot(ecdf(x))
lines(xx, pbhatt(xx, 5, 3, 5), col = "red", lwd = 2)
```

`BirnbaumSaunders` *Birnbaum-Saunders (fatigue life) distribution*

Description

Density, distribution function, quantile function and random generation for the Birnbaum-Saunders (fatigue life) distribution.

Usage

```
dfatigue(x, alpha, beta = 1, mu = 0, log = FALSE)

pfatigue(q, alpha, beta = 1, mu = 0, lower.tail = TRUE, log.p = FALSE)

qfatigue(p, alpha, beta = 1, mu = 0, lower.tail = TRUE, log.p = FALSE)

rfatigue(n, alpha, beta = 1, mu = 0)
```

Arguments

`x, q`	vector of quantiles.
`alpha, beta, mu`	shape, scale and location parameters. Scale and shape must be positive.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \left(\frac{\sqrt{\frac{x-\mu}{\beta}} + \sqrt{\frac{\beta}{x-\mu}}}{2\alpha(x-\mu)} \right) \phi \left(\frac{1}{\alpha} \left(\sqrt{\frac{x-\mu}{\beta}} - \sqrt{\frac{\beta}{x-\mu}} \right) \right)$$

Cumulative distribution function

$$F(x) = \Phi \left(\frac{1}{\alpha} \left(\sqrt{\frac{x-\mu}{\beta}} - \sqrt{\frac{\beta}{x-\mu}} \right) \right)$$

Quantile function

$$F^{-1}(p) = \left[\frac{\alpha}{2}\Phi^{-1}(p) + \sqrt{\left(\frac{\alpha}{2}\Phi^{-1}(p)\right)^2 + 1} \right]^2 \beta + \mu$$

References

Birnbaum, Z. W. and Saunders, S. C. (1969). A new family of life distributions. Journal of Applied Probability, 6(2), 637-652.

Desmond, A. (1985) Stochastic models of failure in random environments. Canadian Journal of Statistics, 13, 171-183.

Vilca-Labra, F., and Leiva-Sanchez, V. (2006). A new fatigue life model based on the family of skew-elliptical distributions. Communications in Statistics-Theory and Methods, 35(2), 229-244.

Leiva, V., Sanhueza, A., Sen, P. K., and Paula, G. A. (2008). Random number generators for the generalized Birnbaum-Saunders distribution. Journal of Statistical Computation and Simulation, 78(11), 1105-1118.

Examples

```
x <- rfatigue(1e5, .5, 2, 5)
xx <- seq(0, 1000, by = 0.1)
hist(x, 100, freq = FALSE)
lines(xx, dfatigue(xx, .5, 2, 5), col = "red")
hist(pfatigue(x, .5, 2, 5))
plot(ecdf(x))
lines(xx, pfatigue(xx, .5, 2, 5), col = "red", lwd = 2)
```

BivNormal — *Bivariate normal distribution*

Description

Density, distribution function and random generation for the bivariate normal distribution.

Usage

```
dbvnorm(x, y = NULL, mean1 = 0, mean2 = mean1, sd1 = 1, sd2 = sd1,
  cor = 0, log = FALSE)

rbvnorm(n, mean1 = 0, mean2 = mean1, sd1 = 1, sd2 = sd1, cor = 0)
```

Arguments

`x, y`	vectors of quantiles; alternativelly x may be a two-column matrix (or data.frame) and y may be omitted.
`mean1, mean2`	vectors of means.
`sd1, sd2`	vectors of standard deviations.
`cor`	vector of correlations (`-1 < cor < 1`).
`log`	logical; if TRUE, probabilities p are given as log(p).
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \frac{1}{2\pi\sqrt{1-\rho^2}\sigma_1\sigma_2} \exp\left(-\frac{1}{2(1-\rho^2)}(z_1^2 - 2\rho z_1 z_2 + z_2^2)\right)$$

where $z_1 = \frac{x_1-\mu_1}{\sigma_1}$ and $z_2 = \frac{x_2-\mu_2}{\sigma_2}$.

References

Krishnamoorthy, K. (2006). Handbook of Statistical Distributions with Applications. Chapman & Hall/CRC

Mukhopadhyay, N. (2000). Probability and statistical inference. Chapman & Hall/CRC

See Also

Normal

Examples

```
y <- x <- seq(-4, 4, by = 0.25)
z <- outer(x, y, function(x, y) dbvnorm(x, y, cor = -0.75))
persp(x, y, z)

y <- x <- seq(-4, 4, by = 0.25)
z <- outer(x, y, function(x, y) dbvnorm(x, y, cor = -0.25))
persp(x, y, z)
```

BivPoiss *Bivariate Poisson distribution*

Description

Probability mass function and random generation for the bivariate Poisson distribution.

Usage

```
dbvpois(x, y = NULL, a, b, c, log = FALSE)

rbvpois(n, a, b, c)
```

Arguments

`x, y`	vectors of quantiles; alternativelly x may be a two-column matrix (or data.frame) and y may be omitted.
`a, b, c`	positive valued parameters.
`log`	logical; if TRUE, probabilities p are given as log(p).
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability mass function

$$f(x) = \exp\{-(a+b+c)\} \frac{a^x}{x!} \frac{b^y}{y!} \sum_{k=0}^{\min(x,y)} \binom{x}{k} \binom{y}{k} k! \left(\frac{c}{ab}\right)^k$$

References

Karlis, D. and Ntzoufras, I. (2003). Analysis of sports data by using bivariate Poisson models. Journal of the Royal Statistical Society: Series D (The Statistician), 52(3), 381-393.

Kocherlakota, S. and Kocherlakota, K. (1992) Bivariate Discrete Distributions. New York: Dekker.

Johnson, N., Kotz, S. and Balakrishnan, N. (1997). Discrete Multivariate Distributions. New York: Wiley.

Holgate, P. (1964). Estimation for the bivariate Poisson distribution. Biometrika, 51(1-2), 241-287.

Kawamura, K. (1984). Direct calculation of maximum likelihood estimator for the bivariate Poisson distribution. Kodai mathematical journal, 7(2), 211-221.

See Also

`Poisson`

Examples

```
x <- rbvpois(5000, 7, 8, 5)
image(prop.table(table(x[,1], x[,2])))
colMeans(x)
```

Categorical *Categorical distribution*

Description

Probability mass function, distribution function, quantile function and random generation for the categorical distribution.

Usage

```
dcat(x, prob, log = FALSE)

pcat(q, prob, lower.tail = TRUE, log.p = FALSE)

qcat(p, prob, lower.tail = TRUE, log.p = FALSE, labels)

rcat(n, prob, labels)
```

Arguments

`x, q`	vector of quantiles.
`prob`	vector of length k, or k-column matrix of probabilities. Probabilities need to sum up to 1.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`labels`	if provided, labeled `factor` vector is returned. Number of labels needs to be the same as number of categories (number of columns in prob).
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Examples

```
# Generating 10 random draws from categorical distribution
# with k=3 categories occuring with equal probabilities
# parametrized using a vector

rcat(10, c(1/3, 1/3, 1/3))

# or with k=5 categories parametrized using a matrix of probabilities
# (generated from Dirichlet distribution)

p <- rdirichlet(10, c(1, 1, 1, 1, 1))
rcat(10, p)

x <- rcat(1e5, c(0.2, 0.4, 0.3, 0.1))
plot(prop.table(table(x)), type = "h")
lines(0:5, dcat(0:5, c(0.2, 0.4, 0.3, 0.1)), col = "red")
```

`Dirichlet` *Dirichlet distribution*

Description

Density function, cumulative distribution function and random generation for the Dirichlet distribution.

Usage

```
ddirichlet(x, alpha, log = FALSE)

rdirichlet(n, alpha)
```

Arguments

- `x` — k-column matrix of quantiles.
- `alpha` — k-values vector or k-column matrix; concentration parameter. Must be positive.
- `log` — logical; if TRUE, probabilities p are given as log(p).
- `n` — number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k x_k^{k-1}$$

References

Devroye, L. (1986). Non-Uniform Random Variate Generation. Springer-Verlag.

Examples

```
# Generating 10 random draws from Dirichlet distribution
# parametrized using a vector

rdirichlet(10, c(1, 1, 1, 1))

# or parametrized using a matrix where each row
# is a vector of parameters

alpha <- matrix(c(1, 1, 1, 1:3, 7:9), ncol = 3, byrow = TRUE)
rdirichlet(10, alpha)
```

DirMnom — *Dirichlet-multinomial (multivariate Polya) distribution*

Description

Density function, cumulative distribution function and random generation for the Dirichlet-multinomial (multivariate Polya) distribution.

Usage

```
ddirmnom(x, size, alpha, log = FALSE)

rdirmnom(n, size, alpha)
```

Arguments

`x`	k-column matrix of quantiles.
`size`	numeric vector; number of trials (zero or more).
`alpha`	k-values vector or k-column matrix; concentration parameter. Must be positive.
`log`	logical; if TRUE, probabilities p are given as log(p).
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

If $(p_1, \ldots, p_k) \sim \mathrm{Dirichlet}(\alpha_1, \ldots, \alpha_k)$ and $(x_1, \ldots, x_k) \sim \mathrm{Multinomial}(n, p_1, \ldots, p_k)$, then $(x_1, \ldots, x_k) \sim \mathrm{DirichletMultinomial}(\mathrm{n}, \alpha_1, \ldots, \alpha_\mathrm{k})$.

Probability density function

$$f(x) = \frac{(n!)\,\Gamma\left(\sum \alpha_k\right)}{\Gamma\left(n + \sum \alpha_k\right)} \prod_{k=1}^{K} \frac{\Gamma(x_k + \alpha_k)}{(x_k!)\,\Gamma(\alpha_k)}$$

References

Gentle, J.E. (2006). Random number generation and Monte Carlo methods. Springer.

Kvam, P. and Day, D. (2001) The multivariate Polya distribution in combat modeling. Naval Research Logistics, 48, 1-17.

See Also

`Dirichlet`, `Multinomial`

`DiscreteLaplace` *Discrete Laplace distribution*

Description

Probability mass, distribution function and random generation for the discrete Laplace distribution parametrized by location and scale.

Usage

```
ddlaplace(x, scale, location = 0, log = FALSE)

pdlaplace(q, scale, location = 0, lower.tail = TRUE, log.p = FALSE)

rdlaplace(n, scale, location = 0)
```

Arguments

`x, q`	vector of quantiles.
`scale`	scale parameter; `0 < scale < 1`.
`location`	location parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

If $U \sim \mathrm{Geometric}(1-p)$ and $V \sim \mathrm{Geometric}(1-p)$, then $U - V \sim \mathrm{DiscreteLaplace}(p)$, where geometric distribution is related to discrete Laplace distribution in similar way as exponential distribution is related to Laplace distribution.

Probability mass function

$$f(x) = \frac{1-p}{1+p} p^{|x-\mu|}$$

Cumulative distribution function

$$F(x) = \begin{cases} \frac{p^{-|x-\mu|}}{1+p} & x < 0 \\ 1 - \frac{p^{|x-\mu|+1}}{1+p} & x \ge 0 \end{cases}$$

References

Inusah, S., & Kozubowski, T.J. (2006). A discrete analogue of the Laplace distribution. Journal of statistical planning and inference, 136(3), 1090-1102.

Kotz, S., Kozubowski, T., & Podgorski, K. (2012). The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance. Springer Science & Business Media.

Examples

```
p <- 0.45
x <- rdlaplace(1e5, p)
xx <- seq(-200, 200, by = 1)
plot(prop.table(table(x)))
lines(xx, ddlaplace(xx, p), col = "red")
hist(pdlaplace(x, p))
plot(ecdf(x))
lines(xx, pdlaplace(xx, p), col = "red")
```

`DiscreteNormal` *Discrete normal distribution*

Description

Probability mass function, distribution function and random generation for discrete normal distribution.

Usage

```
ddnorm(x, mean = 0, sd = 1, log = FALSE)

pdnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)

qdnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)

rdnorm(n, mean = 0, sd = 1)
```

Arguments

`x, q`	vector of quantiles.
`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 $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability mass function

$$f(x) = \Phi\left(\frac{x - \mu + 1}{\sigma}\right) - \Phi\left(\frac{x - \mu}{\sigma}\right)$$

References

Roy, D. (2003). The discrete normal distribution. Communications in Statistics-Theory and Methods, 32, 1871-1883.

See Also

Normal

Examples

```
x <- rdnorm(1e5, 7, 35)
xx <- -150:150
hist(x, 100, freq = FALSE)
lines(xx-0.5, ddnorm(xx, 7, 35), col = "red")
hist(pdnorm(x, 7, 35))
plot(ecdf(x))
lines(xx, pdnorm(xx, 7, 35), col = "red", lwd = 2)
```

DiscreteUniform *Discrete uniform distribution*

Description

Probability mass function, distribution function, quantile function and random generation for the discrete uniform distribution.

Usage

```
ddunif(x, min, max, log = FALSE)

pdunif(q, min, max, lower.tail = TRUE, log.p = FALSE)

qdunif(p, min, max, lower.tail = TRUE, log.p = FALSE)

rdunif(n, min, max)
```

Arguments

`x, q`	vector of quantiles.
`min, max`	lower and upper limits of the distribution. Must be finite.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

If `min == max`, then discrete uniform distribution is a degenerate distribution.

Examples

```
x <- rdunif(1e5, 1, 10)
xx <- -1:11
plot(prop.table(table(x)), type = "h")
lines(xx, ddunif(xx, 1, 10), col = "red")
hist(pdunif(x, 1, 10))
plot(ecdf(x))
lines(xx, pdunif(xx, 1, 10), col = "red")
```

DiscreteWeibull — *Discrete Weibull distribution (type I)*

Description

Density, distribution function, quantile function and random generation for the discrete Weibull (type I) distribution.

Usage

```
ddweibull(x, shape1, shape2, log = FALSE)

pdweibull(q, shape1, shape2, lower.tail = TRUE, log.p = FALSE)

qdweibull(p, shape1, shape2, lower.tail = TRUE, log.p = FALSE)

rdweibull(n, shape1, shape2)
```

Arguments

x, q	vector of quantiles.
shape1, shape2	parameters (named q, β). Values of shape2 need to be positive and 0 < shape1 < 1.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.

Details

Probability mass function

$$f(x) = q^{x^\beta} - q^{(x+1)^\beta}$$

Cumulative distribution function

$$F(x) = 1 - q^{(x+1)^\beta}$$

Quantile function

$$F^{-1}(p) = \left\lceil \left(\frac{\log(1-p)}{\log(q)} \right)^{1/\beta} - 1 \right\rceil$$

References

Nakagawa, T. and Osaki, S. (1975). The Discrete Weibull Distribution. IEEE Transactions on Reliability, R-24, 300-301.

Kulasekera, K.B. (1994). Approximate MLE's of the parameters of a discrete Weibull distribution with type I censored data. Microelectronics Reliability, 34(7), 1185-1188.

Khan, M.A., Khalique, A. and Abouammoh, A.M. (1989). On estimating parameters in a discrete Weibull distribution. IEEE Transactions on Reliability, 38(3), 348-350.

See Also

Weibull

Examples

```
x <- rdweibull(1e5, 0.32, 1)
xx <- seq(-2, 100, by = 1)
plot(prop.table(table(x)), type = "h")
lines(xx, ddweibull(xx, .32, 1), col = "red")

# Notice: distribution of F(X) is far from uniform:
hist(pdweibull(x, .32, 1), 50)

plot(ecdf(x))
lines(xx, pdweibull(xx, .32, 1), col = "red", lwd = 2)
```

---

extraDistr *Additional univariate and multivariate distributions*

Description

Density, distribution function, quantile function and random generation for a number of univariate and multivariate distributions.

Details

This package follows naming convention that is consistent with base R, where density (or probability mass) functions, distribution functions, quantile functions and random generation functions names are followed by `d*`, `p*`, `q*`, and `r*` prefixes.

Behaviour of the functions is consistent with base R, where for not valid parameters values `NaN`'s are returned, while for values beyond function support `0`'s are returned (e.g. for non-integers in discrete distributions, or for negative values in functions with non-negative support).

All the functions vectorized and coded in C++ using **Rcpp**.

Frechet *Frechet distribution*

Description

Density, distribution function, quantile function and random generation for the Frechet distribution.

Usage

```
dfrechet(x, lambda = 1, mu = 0, sigma = 1, log = FALSE)

pfrechet(q, lambda = 1, mu = 0, sigma = 1, lower.tail = TRUE,
  log.p = FALSE)

qfrechet(p, lambda = 1, mu = 0, sigma = 1, lower.tail = TRUE,
  log.p = FALSE)

rfrechet(n, lambda = 1, mu = 0, sigma = 1)
```

Arguments

`x, q`	vector of quantiles.
`lambda, sigma, mu`	shape, scale, and location parameters. Scale and shape must be positive.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \frac{\lambda}{\sigma} \left(\frac{x-\mu}{\sigma}\right)^{-1-\lambda} \exp\left(-\left(\frac{x-\mu}{\sigma}\right)^{-\lambda}\right)$$

Cumulative distribution function

$$F(x) = \exp\left(-\left(\frac{x-\mu}{\sigma}\right)^{-\lambda}\right)$$

Quantile function

$$F^{-1}(p) = \mu + \sigma - \log(p)^{-1/\lambda}$$

References

Bury, K. (1999). Statistical Distributions in Engineering. Cambridge University Press.

Examples

```
x <- rfrechet(1e5, 5, 2, 1.5)
xx <- seq(0, 1000, by = 0.1)
hist(x, 200, freq = FALSE)
lines(xx, dfrechet(xx, 5, 2, 1.5), col = "red")
hist(pfrechet(x, 5, 2, 1.5))
plot(ecdf(x))
lines(xx, pfrechet(xx, 5, 2, 1.5), col = "red", lwd = 2)
```

GammaPoiss *Gamma-Poisson distribution*

Description

Probability mass function and random generation for the gamma-Poisson distribution.

Usage

```
dgpois(x, shape, rate, scale = 1/rate, log = FALSE)

pgpois(q, shape, rate, scale = 1/rate, lower.tail = TRUE, log.p = FALSE)

rgpois(n, shape, rate, scale = 1/rate)
```

Arguments

`x, q`	vector of quantiles.
`shape, scale`	shape and scale parameters. Must be positive, scale strictly.
`rate`	an alternative way to specify the scale.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Gamma-Poisson distribution arises as a continuous mixture of Poisson distributions, where the mixing distribution of the Poisson rate λ is a gamma distribution. When $X \sim \mathrm{Poisson}(\lambda)$ and $\lambda \sim \mathrm{Gamma}(\alpha, \beta)$, then $X \sim \mathrm{GammaPoisson}(\alpha, \beta)$.

Probability density function (parametrized by scale)

$$f(x) = \frac{\Gamma(\alpha + x)}{x!\Gamma(\alpha)} \left(\frac{\beta}{1+\beta} \right)^k \left(1 - \frac{\beta}{1+\beta} \right)^\alpha$$

Warning: cumulative distribution function is defined as

$$F(x) = \sum_{k=0}^{x} f(k)$$

so it may be slow for large datasets.

See Also

Gamma, Poisson

Examples

```
x <- rgpois(1e5, 7, 0.002)
xx <- seq(0, 12000, by = 1)
hist(x, 100, freq = FALSE)
lines(xx, dgpois(xx, 7, 0.002), col = "red")
hist(pgpois(x, 7, 0.002))
plot(ecdf(x))
lines(xx, pgpois(xx, 7, 0.002), col = "red", lwd = 2)
```

GEV — *Generalized extreme value distribution*

Description

Density, distribution function, quantile function and random generation for the generalized extreme value distribution.

Usage

```
dgev(x, mu = 0, sigma = 1, xi = 0, log = FALSE)

pgev(q, mu = 0, sigma = 1, xi = 0, lower.tail = TRUE, log.p = FALSE)

qgev(p, mu = 0, sigma = 1, xi = 0, lower.tail = TRUE, log.p = FALSE)

rgev(n, mu = 0, sigma = 1, xi = 0)
```

Arguments

`x, q`	vector of quantiles.
`mu, sigma, xi`	location, scale, and shape parameters. Scale must be positive.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \begin{cases} \frac{1}{\sigma}\left(1 - \xi\frac{x-\mu}{\sigma}\right)^{-1-1/\xi} \exp\left(-\left(1 - \xi\frac{x-\mu}{\sigma}\right)^{-1/\xi}\right) & \xi \neq 0 \\ \frac{1}{\sigma}\exp\left(-\frac{x-\mu}{\sigma}\right)\exp\left(-\exp\left(-\frac{x-\mu}{\sigma}\right)\right) & \xi = 0 \end{cases}$$

Cumulative distribution function

$$F(x) = \begin{cases} \exp\left(-\left(1 + \xi\frac{x-\mu}{\sigma}\right)^{1/\xi}\right) & \xi \neq 0 \\ \exp\left(-\exp\left(-\frac{x-\mu}{\sigma}\right)\right) & \xi = 0 \end{cases}$$

Quantile function

$$F^{-1}(p) = \begin{cases} \mu - \frac{\sigma}{\xi}(1 - (-\log(p))^{\xi}) & \xi \neq 0 \\ \mu - \sigma\log(-\log(p)) & \xi = 0 \end{cases}$$

References

Coles, S. (2001). An Introduction to Statistical Modeling of Extreme Values. Springer.

Examples

```
x <- rgev(1e5, 5, 2, .5)
xx <- seq(0, 1000, by = 0.1)
hist(x, 1000, freq = FALSE, xlim = c(0, 50))
lines(xx, dgev(xx, 5, 2, .5), col = "red")
hist(pgev(x, 5, 2, .5))
plot(ecdf(x))
lines(xx, pgev(xx, 5, 2, .5), col = "red", lwd = 2)
```

`Gompertz` *Gompertz distribution*

Description

Density, distribution function, quantile function and random generation for the Gompertz distribution.

Usage

```
dgompertz(x, a = 1, b = 1, log = FALSE)

pgompertz(q, a = 1, b = 1, lower.tail = TRUE, log.p = FALSE)

qgompertz(p, a = 1, b = 1, lower.tail = TRUE, log.p = FALSE)

rgompertz(n, a = 1, b = 1)
```

Arguments

`x, q`	vector of quantiles.
`a, b`	positive valued scale and location parameters.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = a \exp\left(bx - \frac{a}{b}(\exp(bx) - 1)\right)$$

Cumulative distribution function

$$F(x) = 1 - \exp\left(-\frac{a}{b}(exp(bx) - 1)\right)$$

Quantile function

$$F^{-1}(p) = \frac{1}{b} \log\left(1 - \frac{b}{a}\log(1 - p)\right)$$

References

Lenart, A. (2012). The Gompertz distribution and Maximum Likelihood Estimation of its parameters - a revision. MPIDR WORKING PAPER WP 2012-008. http://www.demogr.mpg.de/papers/working/wp-2012-008.pdf

Examples

```
x <- rgompertz(1e5, 5, 2)
xx <- seq(0, 1000, by = 0.1)
hist(x, 100, freq = FALSE)
lines(xx, dgompertz(xx, 5, 2), col = "red")
hist(pgompertz(x, 5, 2))
plot(ecdf(x))
lines(xx, pgompertz(xx, 5, 2), col = "red", lwd = 2)
```

GPD *Generalized Pareto distribution*

Description

Density, distribution function, quantile function and random generation for the generalized Pareto distribution.

Usage

```
dgpd(x, mu = 0, sigma = 1, xi = 0, log = FALSE)

pgpd(q, mu = 0, sigma = 1, xi = 0, lower.tail = TRUE, log.p = FALSE)

qgpd(p, mu = 0, sigma = 1, xi = 0, lower.tail = TRUE, log.p = FALSE)

rgpd(n, mu = 0, sigma = 1, xi = 0)
```

Arguments

`x, q`	vector of quantiles.
`mu, sigma, xi`	location, scale, and shape parameters. Scale must be positive.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \begin{cases} \frac{1}{\sigma} \left(1 + \xi \frac{x-\mu}{\sigma}\right)^{-(\xi+1)/\xi} & \xi \neq 0 \\ \frac{1}{\sigma} \exp\left(-\frac{x-\mu}{\sigma}\right) & \xi = 0 \end{cases}$$

Cumulative distribution function

$$F(x) = \begin{cases} 1 - \left(1 + \xi \frac{x-\mu}{\sigma}\right)^{-1/\xi} & \xi \neq 0 \\ 1 - \exp\left(-\frac{x-\mu}{\sigma}\right) & \xi = 0 \end{cases}$$

Quantile function

$$F^{-1}(x) = \begin{cases} \mu + \sigma \frac{(1-p)^{-\xi} - 1}{\xi} & \xi \neq 0 \\ \mu - \sigma \log(1-p) & \xi = 0 \end{cases}$$

References

Coles, S. (2001). An Introduction to Statistical Modeling of Extreme Values. Springer.

Examples

```
x <- rgpd(1e5, 5, 2, .1)
xx <- seq(0, 1000, by = 0.1)
hist(x, 100, freq = FALSE, xlim = c(0, 50))
lines(xx, dgpd(xx, 5, 2, .1), col = "red")
hist(pgpd(x, 5, 2, .1))
plot(ecdf(x))
lines(xx, pgpd(xx, 5, 2, .1), col = "red", lwd = 2)
```

`Gumbel` *Gumbel distribution*

Description

Density, distribution function, quantile function and random generation for the Gumbel distribution.

Usage

```
dgumbel(x, mu = 0, sigma = 1, log = FALSE)

pgumbel(q, mu = 0, sigma = 1, lower.tail = TRUE, log.p = FALSE)

qgumbel(p, mu = 0, sigma = 1, lower.tail = TRUE, log.p = FALSE)

rgumbel(n, mu = 0, sigma = 1)
```

Arguments

`x, q`	vector of quantiles.
`mu, sigma`	location and scale parameters. Scale must be positive.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).

lower.tail	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
p	vector of probabilities.
n	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \frac{1}{\sigma} \exp\left(-\left(\frac{x-\mu}{\sigma} + \exp\left(-\frac{x-\mu}{\sigma}\right)\right)\right)$$

Cumulative distribution function

$$F(x) = \exp\left(-\exp\left(-\frac{x-\mu}{\sigma}\right)\right)$$

Quantile function

$$F^{-1}(p) = \mu - \sigma \log(-\log(p))$$

References

Bury, K. (1999). Statistical Distributions in Engineering. Cambridge University Press.

Examples

```
x <- rgumbel(1e5, 5, 2)
xx <- seq(0, 1000, by = 0.1)
hist(x, 100, freq = FALSE)
lines(xx, dgumbel(xx, 5, 2), col = "red")
hist(pgumbel(x, 5, 2))
plot(ecdf(x))
lines(xx, pgumbel(xx, 5, 2), col = "red", lwd = 2)
```

HalfCauchy *Half-Cauchy distribution*

Description

Density, distribution function, quantile function and random generation for the half-Cauchy distribution.

Usage

```
dhcauchy(x, sigma = 1, log = FALSE)

phcauchy(q, sigma = 1, lower.tail = TRUE, log.p = FALSE)

qhcauchy(p, sigma = 1, lower.tail = TRUE, log.p = FALSE)

rhcauchy(n, sigma = 1)
```

Arguments

x, q	vector of quantiles.
sigma	positive valued scale parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
p	vector of probabilities.
n	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

If X follows Cauchy centered at 0 and parametrized by scale σ, then $|X|$ follows half-Cauchy distribution parametrized by scale σ. Half-Cauchy distribution is a special case of half-t distribution with $\nu = 1$ degrees of freedom.

References

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). Bayesian analysis, 1(3), 515-534.

Jacob, E. and Jayakumar, K. (2012). On Half-Cauchy Distribution and Process. International Journal of Statistika and Mathematika, 3(2), 77-81.

See Also

HalfT

Examples

```
x <- rhcauchy(1e5, 2)
xx <- seq(-1, 100, by = 0.01)
hist(x, 2e5, freq = FALSE, xlim = c(0, 100))
lines(xx, dhcauchy(xx, 2), col = "red")
hist(phcauchy(x, 2))
plot(ecdf(x), xlim = c(0, 100))
lines(xx, phcauchy(xx, 2), col = "red", lwd = 2)
```

`HalfNormal` *Half-normal distribution*

Description

Density, distribution function, quantile function and random generation for the half-normal distribution.

Usage

```
dhnorm(x, sigma = 1, log = FALSE)

phnorm(q, sigma = 1, lower.tail = TRUE, log.p = FALSE)

qhnorm(p, sigma = 1, lower.tail = TRUE, log.p = FALSE)

rhnorm(n, sigma = 1)
```

Arguments

`x, q`	vector of quantiles.
`sigma`	positive valued scale parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

If X follows normal distribution centered at 0 and parametrized by scale σ, then $|X|$ follows half-normal distribution parametrized by scale σ. Half-t distribution with $\nu = \infty$ degrees of freedom converges to half-normal distribution.

References

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). Bayesian analysis, 1(3), 515-534.

Jacob, E. and Jayakumar, K. (2012). On Half-Cauchy Distribution and Process. International Journal of Statistika and Mathematika, 3(2), 77-81.

See Also

HalfT

Examples

```
x <- rhnorm(1e5, 2)
xx <- seq(-1, 100, by = 0.01)
hist(x, 100, freq = FALSE)
lines(xx, dhnorm(xx, 2), col = "red")
hist(phnorm(x, 2))
plot(ecdf(x))
lines(xx, phnorm(xx, 2), col = "red", lwd = 2)
```

HalfT *Half-t distribution*

Description

Density, distribution function, quantile function and random generation for the half-t distribution.

Usage

```
dht(x, nu, sigma = 1, log = FALSE)

pht(q, nu, sigma = 1, lower.tail = TRUE, log.p = FALSE)

qht(p, nu, sigma = 1, lower.tail = TRUE, log.p = FALSE)

rht(n, nu, sigma = 1)
```

Arguments

`x, q`	vector of quantiles.
`nu, sigma`	positive valued degrees of freedom and scale parameters.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

If X follows t distribution parametrized by degrees of freedom ν and scale σ, then $|X|$ follows half-t distribution parametrized by degrees of freedom ν and scale σ.

References

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). Bayesian analysis, 1(3), 515-534.

Jacob, E. and Jayakumar, K. (2012). On Half-Cauchy Distribution and Process. International Journal of Statistika and Mathematika, 3(2), 77-81.

See Also

HalfNormal, HalfCauchy

Examples

```
x <- rht(1e5, 2, 2)
xx <- seq(-1, 100, by = 0.01)
hist(x, 500, freq = FALSE, xlim = c(0, 100))
lines(xx, dht(xx, 2, 2), col = "red")
hist(pht(x, 2, 2))
plot(ecdf(x), xlim = c(0, 100))
lines(xx, pht(xx, 2, 2), col = "red", lwd = 2)
```

Huber *"Huber density" distribution*

Description

Density, distribution function, quantile function and random generation for the "Huber density" distribution.

Usage

```
dhuber(x, mu = 0, sigma = 1, epsilon = 1.345, log = FALSE)

phuber(q, mu = 0, sigma = 1, epsilon = 1.345, lower.tail = TRUE,
  log.p = FALSE)

qhuber(p, mu = 0, sigma = 1, epsilon = 1.345, lower.tail = TRUE,
  log.p = FALSE)

rhuber(n, mu = 0, sigma = 1, epsilon = 1.345)
```

Arguments

`x, q`	vector of quantiles.
`mu, sigma, epsilon`	location, and scale, and shape parameters. Scale and shape must be positive.

`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Huber density is connected to Huber loss and can be defined as:

$$f(x) = \frac{1-\epsilon}{\sqrt{2\pi}} e^{-\rho_k(x)}$$

where

$$\rho_k(x) = \left\{ \begin{array}{ll} \frac{1}{2}x^2 & |x| \le k \\ k|x| - \frac{1}{2}k^2 & |x| > k \end{array} \right.$$

and ϵ satisfies

$$\frac{2\phi(k)}{k} - 2\Phi(-k) = \frac{\epsilon}{1-\epsilon}$$

References

Huber, P.J. (1964). Robust Estimation of a Location Parameter. Annals of Statistics, 53(1), 73-101.

Huber, P.J. (1981). Robust Statistics. Wiley.

Schumann, D. (2009). Robust Variable Selection. ProQuest.

Examples

```
x <- rhuber(1e5, 5, 2, 3)
xx <- seq(-20, 20, by = 0.1)
hist(x, 100, freq = FALSE)
lines(xx, dhuber(xx, 5, 2, 3), col = "red")
hist(phuber(x, 5, 2, 3))
plot(ecdf(x))
lines(xx, phuber(xx, 5, 2, 3), col = "red", lwd = 2)
```

InvChiSq *Inverse chi-squared and scaled chi-squared distributions*

Description

Density, distribution function and random generation for the inverse chi-squared distribution and scaled chi-squared distribution.

Usage

```
dinvchisq(x, nu, tau, log = FALSE)

pinvchisq(q, nu, tau, lower.tail = TRUE, log.p = FALSE)

qinvchisq(p, nu, tau, lower.tail = TRUE, log.p = FALSE)

rinvchisq(n, nu, tau)
```

Arguments

`x, q`	vector of quantiles.
`nu`	positive valued shape parameter.
`tau`	positive valued scaling parameter; if provided it returns values for scaled chi-squared distributions.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

If X follows $\chi^2(\nu)$ distribution, then $1/X$ follows inverse chi-squared distribution parametrized by ν. Inverse chi-squared distribution is a special case of inverse gamma distribution with parameters $\alpha = \frac{\nu}{2}$ and $\beta = \frac{1}{2}$; or $\alpha = \frac{\nu}{2}$ and $\beta = \frac{\nu\tau^2}{2}$ for scaled inverse chi-squared distribution.

See Also

`Chisquare`, `GammaDist`

Examples

```
x <- rinvchisq(1e5, 20)
xx <- seq(0, 6, by = 0.001)
hist(x, 100, freq = FALSE)
lines(xx, dinvchisq(xx, 20), col = "red")
hist(pinvchisq(x, 20))
plot(ecdf(x))
lines(xx, pinvchisq(xx, 20), col = "red", lwd = 2)

# scaled

x <- rinvchisq(1e5, 10, 5)
xx <- seq(0, 700, by = 0.01)
hist(x, 100, freq = FALSE)
lines(xx, dinvchisq(xx, 10, 5), col = "red")
hist(pinvchisq(x, 10, 5))
plot(ecdf(x))
lines(xx, pinvchisq(xx, 10, 5), col = "red", lwd = 2)
```

InvGamma — *Inverse-gamma distribution*

Description

Density, distribution function and random generation for the inverse-gamma distribution.

Usage

```
dinvgamma(x, alpha, beta = 1, log = FALSE)

pinvgamma(q, alpha, beta = 1, lower.tail = TRUE, log.p = FALSE)

qinvgamma(p, alpha, beta = 1, lower.tail = TRUE, log.p = FALSE)

rinvgamma(n, alpha, beta = 1)
```

Arguments

`x, q`	vector of quantiles.
`alpha, beta`	positive valued shape and scale parameters.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability mass function

$$f(x) = \frac{x^{-\alpha-1}\exp(-\frac{1}{\beta x})}{\Gamma(\alpha)\beta^{\alpha}}$$

Cumulative distribution function

$$F(x) = \frac{\gamma(\alpha, \frac{1}{\beta x})}{\Gamma(\alpha)}$$

References

Witkovsky, V. (2001). Computing the distribution of a linear combination of inverted gamma variables. Kybernetika 37(1), 79-90.

Leemis, L.M. and McQueston, L.T. (2008). Univariate Distribution Relationships. American Statistician 62(1): 45-53.

See Also

GammaDist

Examples

```
x <- rinvgamma(1e5, 20, 3)
xx <- seq(0, 1, by = 0.001)
hist(x, 100, freq = FALSE)
lines(xx, dinvgamma(xx, 20, 3), col = "red")
hist(pinvgamma(x, 20, 3))
plot(ecdf(x))
lines(xx, pinvgamma(xx, 20, 3), col = "red", lwd = 2)
```

Kumaraswamy *Kumaraswamy distribution*

Description

Density, distribution function, quantile function and random generation for the Kumaraswamy distribution.

Usage

```
dkumar(x, a = 1, b = 1, log = FALSE)

pkumar(q, a = 1, b = 1, lower.tail = TRUE, log.p = FALSE)

qkumar(p, a = 1, b = 1, lower.tail = TRUE, log.p = FALSE)

rkumar(n, a = 1, b = 1)
```

Arguments

`x, q`	vector of quantiles.
`a, b`	positive valued parameters.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = abx^{a-1}(1 - x^a)^{b-1}$$

Cumulative distribution function

$$F(x) = 1 - (1 - x^a)^b$$

Quantile function

$$F^{-1}(p) = 1 - (1 - p^{1/b})^{1/a}$$

References

Jones, M. C. (2009). Kumaraswamy's distribution: A beta-type distribution with some tractability advantages. Statistical Methodology, 6, 70-81.

Cordeiro, G.M. and de Castro, M. (2009). A new family of generalized distributions. Journal of Statistical Computation & Simulation, 1-17.

Examples

```
x <- rkumar(1e5, 5, 16)
xx <- seq(0, 1, by = 0.001)
hist(x, 100, freq = FALSE)
lines(xx, dkumar(xx, 5, 16), col = "red")
hist(pkumar(x, 5, 16))
plot(ecdf(x))
lines(xx, pkumar(xx, 5, 16), col = "red", lwd = 2)
```

`Laplace` *Laplace distribution*

Description

Density, distribution function, quantile function and random generation for the Laplace distribution.

Usage

```
dlaplace(x, mu = 0, sigma = 1, log = FALSE)

plaplace(q, mu = 0, sigma = 1, lower.tail = TRUE, log.p = FALSE)

qlaplace(p, mu = 0, sigma = 1, lower.tail = TRUE, log.p = FALSE)

rlaplace(n, mu = 0, sigma = 1)
```

Arguments

`x, q`	vector of quantiles.
`mu, sigma`	location and scale parameters. Scale must be positive.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \frac{1}{2\sigma} \exp\left(-\left|\frac{x-\mu}{\sigma}\right|\right)$$

Cumulative distribution function

$$F(x) = \left\{ \begin{array}{ll} \frac{1}{2} \exp\left(\frac{x-\mu}{\sigma}\right) & x < \mu \\ 1 - \frac{1}{2} \exp\left(\frac{x-\mu}{\sigma}\right) & x \ge \mu \end{array} \right.$$

Quantile function

$$F^{-1}(p) = \left\{ \begin{array}{ll} \mu + \sigma \log(2p) & p < 0.5 \\ \mu + \sigma \log(2(1-p)) & p \ge 0.5 \end{array} \right.$$

References

Krishnamoorthy, K. (2006). Handbook of Statistical Distributions with Applications. Chapman & Hall/CRC

Forbes, C., Evans, M. Hastings, N., & Peacock, B. (2011). Statistical Distributions. John Wiley & Sons.

Examples

```
x <- rlaplace(1e5, 5, 16)
xx <- seq(-100, 100, by = 0.01)
hist(x, 100, freq = FALSE)
lines(xx, dlaplace(xx, 5, 16), col = "red")
hist(plaplace(x, 5, 16))
plot(ecdf(x))
lines(xx, plaplace(xx, 5, 16), col = "red", lwd = 2)
```

`LogSeries` *Logarythmic series distribution*

Description

Density, distribution function, quantile function and random generation for the logarythmic series distribution.

Usage

```
dlgser(x, theta, log = FALSE)

plgser(q, theta, lower.tail = TRUE, log.p = FALSE)

qlgser(p, theta, lower.tail = TRUE, log.p = FALSE)

rlgser(n, theta)
```

Arguments

`x, q`	vector of quantiles.
`theta`	vector; concentration parameter; (`0 < theta < 1`).
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability mass function

$$f(x) = \frac{-1}{\log(1-\theta)} \frac{\theta^x}{x}$$

Cumulative distribution function

$$F(x) = \frac{-1}{\log(1-\theta)} \sum_{k=1}^{x} \frac{\theta^x}{x}$$

Quantile function and random generation are computed using algorithm described in Krishnamoorthy (2006).

References

Krishnamoorthy, K. (2006). Handbook of Statistical Distributions with Applications. Chapman & Hall/CRC

Forbes, C., Evans, M. Hastings, N., & Peacock, B. (2011). Statistical Distributions. John Wiley & Sons.

Examples

```
x <- rlgser(1e5, 0.66)
xx <- seq(0, 100, by = 1)
plot(prop.table(table(x)), type = "h")
lines(xx, dlgser(xx, 0.66), col = "red")

# Notice: distribution of F(X) is far from uniform:
hist(plgser(x, 0.66), 50)

plot(ecdf(x))
lines(xx, plgser(xx, 0.66), col = "red", lwd = 2)
```

Lomax *Lomax distribution*

Description

Density, distribution function, quantile function and random generation for the Lomax distribution.

Usage

```
dlomax(x, lambda, kappa, log = FALSE)

plomax(q, lambda, kappa, lower.tail = TRUE, log.p = FALSE)

qlomax(p, lambda, kappa, lower.tail = TRUE, log.p = FALSE)

rlomax(n, lambda, kappa)
```

Arguments

`x, q`	vector of quantiles.
`lambda, kappa`	positive valued parameters.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \frac{\lambda\kappa}{(1+\lambda x)^{\kappa+1}}$$

Cumulative distribution function

$$F(x) = 1 - (1+\lambda x)^{-\kappa}$$

Quantile function

$$F^{-1}(p) = \frac{(1-p)^{-1/\kappa} - 1}{\lambda}$$

Examples

```
x <- rlomax(1e5, 5, 16)
xx <- seq(-100, 100, by = 0.001)
hist(x, 100, freq = FALSE)
lines(xx, dlomax(xx, 5, 16), col = "red")
hist(plomax(x, 5, 16))
plot(ecdf(x))
lines(xx, plomax(xx, 5, 16), col = "red", lwd = 2)
```

`MultiHypergeometric` *Multivariate hypergeometric distribution*

Description

Probability mass function and random generation for the multivariate hypergeometric distribution.

Usage

```
dmvhyper(x, n, k, log = FALSE)

rmvhyper(nn, n, k)
```

Arguments

`x`	m-column matrix of quantiles.
`n`	m-length vector or m-column matrix of numbers of balls in m colors.
`k`	the number of balls drawn from the urn.
`log`	logical; if TRUE, probabilities p are given as log(p).
`nn`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability mass function

$$f(x) = \frac{\prod_{i=1}^m \binom{n_i}{x_i}}{\binom{N}{k}}$$

The multivariate hypergeometric distribution is generalization of hypergeometric distribution. It is used for sampling *without* replacement k out of N marbles in m colors, where each of the colors appears n_i times. Where $k = \sum_{i=1}^m x_i$, $N = \sum_{i=1}^m n_i$ and $k \leq N$.

References

Gentle, J.E. (2006). Random number generation and Monte Carlo methods. Springer.

See Also

Hypergeometric

Examples

```
# Generating 10 random draws from multivariate hypergeometric
# distribution parametrized using a vector

rmvhyper(10, c(10, 12, 5, 8, 11), 33)
```

Multinomial *Multinomial distribution*

Description

Probability mass function and random generation for the multinomial distribution.

Usage

```
dmnom(x, size, prob, log = FALSE)

rmnom(n, size, prob)
```

Arguments

x	k-column matrix of quantiles.
size	numeric vector; number of trials (zero or more).
prob	k-column numeric matrix; probability of success on each trial.
log	logical; if TRUE, probabilities p are given as log(p).
n	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability mass function

$$f(x) = \frac{n!}{\prod_{i=1}^k x_i} \prod_{i=1}^k p_i^{x_i}$$

References

Gentle, J.E. (2006). Random number generation and Monte Carlo methods. Springer.

See Also

`Binomial`, `Multinomial`

Examples

```
# Generating 10 random draws from multinomial distribution
# parametrized using a vector

(x <- rmnom(10, 3, c(1/3, 1/3, 1/3)))

# Results are consistent with dmultinom() from stats:

all.equal(dmultinom(x[1,], 3, c(1/3, 1/3, 1/3)),
          dmnom(x[1, , drop = FALSE], 3, c(1/3, 1/3, 1/3)))
```

NonStandardT *Non-standard t-distribution*

Description

Probability mass function, distribution function and random generation for non-standard t-distribution. Non-standard t-distribution besides degrees of freedom ν, is parametrized using additional parameters μ for location and σ for scale ($\mu = 0$ and $\sigma = 1$ for standard t-distribution).

Usage

```
dnst(x, df, mu = 0, sigma = 1, log = FALSE)

pnst(q, df, mu = 0, sigma = 1, lower.tail = TRUE, log.p = FALSE)

qnst(p, df, mu = 0, sigma = 1, lower.tail = TRUE, log.p = FALSE)

rnst(n, df, mu = 0, sigma = 1)
```

Arguments

`x, q`	vector of quantiles.
`df`	degrees of freedom (> 0, maybe non-integer). `df = Inf` is allowed.
`mu`	vector of locations
`sigma`	vector of positive valued scale parameters.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

See Also

`TDist`

Examples

```
x <- rnst(1e5, 1000, 5, 13)
xx <- seq(-60, 60, by = 0.01)
hist(x, 100, freq = FALSE)
lines(xx-0.5, dnst(xx, 1000, 5, 13), col = "red")
hist(pnst(x, 1000, 5, 13))
plot(ecdf(x))
lines(xx, pnst(xx, 1000, 5, 13), col = "red", lwd = 2)
```

`NormalMix` *Mixture of normal distributions*

Description

Density, distribution function and random generation for the mixture of normal distributions.

Usage

```
dmixnorm(x, mean, sd, alpha, log = FALSE)

pmixnorm(q, mean, sd, alpha, lower.tail = TRUE, log.p = FALSE)

rmixnorm(n, mean, sd, alpha)
```

Arguments

`x, q`	vector of quantiles.
`mean`	matrix (or vector) of means.
`sd`	matrix (or vector) of standard deviations.
`alpha`	matrix (or vector) of mixing proportions; mixing proportions need to sum up to 1.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`p`	vector of probabilities.

Details

Probability density function

$$f(x) = \alpha_1 f_1(x; \mu_1, \sigma_1) + \ldots + \alpha_k f_k(x; \mu_k, \sigma_k)$$

Cumulative distribution function

$$F(x) = \alpha_1 F_1(x; \mu_1, \sigma_1) + \ldots + \alpha_k F_k(x; \mu_k, \sigma_k)$$

where $\sum_i \alpha_i = 1$.

Examples

```
x <- rmixnorm(1e5, c(0.5, 3, 6), c(3, 1, 1), c(1/3, 1/3, 1/3))
xx <- seq(-20, 20, by = 0.1)
hist(x, 100, freq = FALSE)
lines(xx, dmixnorm(xx, c(0.5, 3, 6), c(3, 1, 1), c(1/3, 1/3, 1/3)), col = "red")
hist(pmixnorm(x, c(0.5, 3, 6), c(3, 1, 1), c(1/3, 1/3, 1/3)))
plot(ecdf(x))
lines(xx, pmixnorm(xx, c(0.5, 3, 6), c(3, 1, 1), c(1/3, 1/3, 1/3)), col = "red", lwd = 2)
```

NSBeta *Non-standard beta distribution*

Description

Non-standard form of beta distribution with lower and upper bounds denoted as `min` and `max`. By default `min=0` and `max=1` what leads to standard beta distribution.

Usage

```
dnsbeta(x, shape1, shape2, min = 0, max = 1, log = FALSE)

pnsbeta(q, shape1, shape2, min = 0, max = 1, lower.tail = TRUE,
  log.p = FALSE)

qnsbeta(p, shape1, shape2, min = 0, max = 1, lower.tail = TRUE,
  log.p = FALSE)

rnsbeta(n, shape1, shape2, min = 0, max = 1)
```

Arguments

`x, q`	vector of quantiles.
`shape1, shape2`	non-negative parameters of the Beta distribution.
`min, max`	lower and upper bounds.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$, otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

See Also

Beta

Examples

```
x <- rnsbeta(1e5, 5, 13, -4, 8)
xx <- seq(-20, 20, by = 0.1)
hist(x, 100, freq = FALSE)
lines(xx, dnsbeta(xx, 5, 13, -4, 8), col = "red")
hist(pnsbeta(x, 5, 13, -4, 8))
plot(ecdf(x))
lines(xx, pnsbeta(xx, 5, 13, -4, 8), col = "red", lwd = 2)
```

`Pareto` *Pareto distribution*

Description

Density, distribution function, quantile function and random generation for the Pareto distribution.

Usage

```
dpareto(x, a = 1, b = 1, log = FALSE)

ppareto(q, a = 1, b = 1, lower.tail = TRUE, log.p = FALSE)

qpareto(p, a = 1, b = 1, lower.tail = TRUE, log.p = FALSE)

rpareto(n, a = 1, b = 1)
```

Arguments

`x, q`	vector of quantiles.
`a, b`	positive valued scale and location parameters.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \frac{ab^a}{x^{a+1}}$$

Cumulative distribution function

$$F(x) = 1 - \left(\frac{b}{x}\right)^a$$

Quantile function

$$F^{-1}(p) = \frac{b}{(1-p)^{1-a}}$$

References

Krishnamoorthy, K. (2006). Handbook of Statistical Distributions with Applications. Chapman & Hall/CRC

Examples

```
x <- rpareto(1e5, 5, 16)
xx <- seq(-100, 100, by = 0.001)
hist(x, 100, freq = FALSE)
lines(xx, dpareto(xx, 5, 16), col = "red")
hist(ppareto(x, 5, 16))
plot(ecdf(x))
lines(xx, ppareto(xx, 5, 16), col = "red", lwd = 2)
```

PoissonMix *Mixture of Poisson distributions*

Description

Density, distribution function and random generation for the mixture of Poisson distributions.

Usage

```
dmixpois(x, lambda, alpha, log = FALSE)

pmixpois(q, lambda, alpha, lower.tail = TRUE, log.p = FALSE)

rmixpois(n, lambda, alpha)
```

Arguments

`x, q`	vector of quantiles.
`lambda`	matrix (or vector) of (non-negative) means.
`alpha`	matrix (or vector) of mixing proportions; mixing proportions need to sum up to 1.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`p`	vector of probabilities.

Details

Probability density function

$$f(x) = \alpha_1 f_1(x; \lambda_1) + \ldots + \alpha_k f_k(x; \lambda_k)$$

Cumulative distribution function

$$F(x) = \alpha_1 F_1(x; \lambda_1) + \ldots + \alpha_k F_k(x; \lambda_k)$$

where $\sum_i \alpha_i = 1$.

Examples

```
x <- rmixpois(1e5, c(5, 12, 19), c(1/3, 1/3, 1/3))
xx <- seq(-1, 50)
plot(prop.table(table(x)))
lines(xx, dmixpois(xx, c(5, 12, 19), c(1/3, 1/3, 1/3)), col = "red")
hist(pmixpois(x, c(5, 12, 19), c(1/3, 1/3, 1/3)))
plot(ecdf(x))
lines(xx, pmixpois(xx, c(5, 12, 19), c(1/3, 1/3, 1/3)), col = "red", lwd = 2)
```

`PowerDist` *Power distribution*

Description

Density, distribution function, quantile function and random generation for the power distribution.

Usage

```
dpower(x, alpha, beta, log = FALSE)

ppower(q, alpha, beta, lower.tail = TRUE, log.p = FALSE)

qpower(p, alpha, beta, lower.tail = TRUE, log.p = FALSE)

rpower(n, alpha, beta)
```

Arguments

`x, q`	vector of quantiles.
`alpha, beta`	parameters.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \frac{\beta x^{\beta-1}}{\alpha^\beta}$$

Cumulative distribution function

$$F(x) = \frac{x^\beta}{\alpha^\beta}$$

Quantile function

$$F^{-1}(p) = \alpha p^{1/\beta}$$

Examples

```
x <- rpower(1e5, 5, 16)
xx <- seq(-100, 100, by = 0.001)
hist(x, 100, freq = FALSE)
lines(xx, dpower(xx, 5, 16), col = "red")
hist(ppower(x, 5, 16))
plot(ecdf(x))
lines(xx, ppower(xx, 5, 16), col = "red", lwd = 2)
```

PropBeta *Beta distribution of proportions*

Description

Probability mass function, distribution function and random generation for the reparametrized beta distribution.

Usage

```
dprop(x, size, mean, log = FALSE)

pprop(q, size, mean, lower.tail = TRUE, log.p = FALSE)

qprop(p, size, mean, lower.tail = TRUE, log.p = FALSE)

rprop(n, size, mean)
```

Arguments

`x, q`	vector of quantiles.
`size`	precision or number of binomial trials (zero or more).
`mean`	mean proportion or probability of success on each trial; `0 < mean < 1`.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Beta can be understood as a distribution of $x = k/n$ proportions in n trials where the average proportion is denoted as μ, so it's parameters become $\alpha = n\mu + 1$ and $\beta = n(1 - \mu) + 1$ and it's density function becomes:

$$f(x) = \frac{1}{\mathrm{B}(n\mu + 1, n(1 - \mu) + 1)} x^{n\mu} (1 - x)^{n(1-\mu)}$$

Alternatively n may be understood as precision parameter as described Ferrari and Cribari-Neto (2004).

References

Ferrari, S., & Cribari-Neto, F. (2004). Beta regression for modelling rates and proportions. Journal of Applied Statistics, 31(7), 799-815.

Examples

```
x <- rprop(1e5, 100, 0.33)
xx <- seq(0, 1, by = 0.01)
hist(x, 100, freq = FALSE)
lines(xx, dprop(xx, 100, 0.33), col = "red")
hist(pprop(x, 100, 0.33))
plot(ecdf(x))
lines(xx, pprop(xx, 100, 0.33), col = "red", lwd = 2)
```

`Rayleigh` *Rayleigh distribution*

Description

Density, distribution function, quantile function and random generation for the Rayleigh distribution.

Usage

```
drayleigh(x, sigma = 1, log = FALSE)

prayleigh(q, sigma = 1, lower.tail = TRUE, log.p = FALSE)

qrayleigh(p, sigma = 1, lower.tail = TRUE, log.p = FALSE)

rrayleigh(n, sigma = 1)
```

Arguments

`x, q`	vector of quantiles.
`sigma`	positive valued parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \frac{x}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

Cumulative distribution function

$$F(x) = 1 - \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

Quantile function

$$F^{-1}(p) = \sqrt{-2\sigma^2 \log(1-p)}$$

References

Krishnamoorthy, K. (2006). Handbook of Statistical Distributions with Applications. Chapman & Hall/CRC.

Forbes, C., Evans, M. Hastings, N., & Peacock, B. (2011). Statistical Distributions. John Wiley & Sons.

Examples

```
x <- rrayleigh(1e5, 13)
xx <- seq(-100, 100, by = 0.001)
hist(x, 100, freq = FALSE)
lines(xx, drayleigh(xx, 13), col = "red")
hist(prayleigh(x, 13))
plot(ecdf(x))
lines(xx, prayleigh(xx, 13), col = "red", lwd = 2)
```

Skellam *Skellam distribution*

Description

Probability mass function and random generation for the Skellam distribution.

Usage

```
dskellam(x, mu1, mu2, log = FALSE)

rskellam(n, mu1, mu2)
```

Arguments

x	vector of quantiles.
mu1, mu2	positive valued parameters.
log	logical; if TRUE, probabilities p are given as log(p).
n	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

If X and Y follow Poisson distributions with means μ_1 and μ_2, than $X - Y$ follows Skellam distribution parametrized by μ_1 and μ_2.

Probability mass function

$$f(x) = e^{-(\mu_1+\mu_2)} \left(\frac{\mu_1}{\mu_2}\right)^{k/2} I_k(2\sqrt{\mu_1\mu_2})$$

References

Karlis, D., & Ntzoufras, I. (2006). Bayesian analysis of the differences of count data. Statistics in medicine, 25(11), 1885-1905.

Examples

```
x <- rskellam(1e5, 5, 13)
xx <- -40:40
plot(prop.table(table(x)), type = "h")
lines(xx, dskellam(xx, 5, 13), col = "red")
```

`Slash` *Slash distribution*

Description

Probability mass function, distribution function and random generation for slash distribution.

Usage

```
dslash(x, mu = 0, sigma = 1, log = FALSE)

pslash(q, mu = 0, sigma = 1, lower.tail = TRUE, log.p = FALSE)

rslash(n, mu = 0, sigma = 1)
```

Arguments

`x, q`	vector of quantiles.
`mu`	vector of locations
`sigma`	vector of positive valued scale parameters.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

If $Z \sim \mathrm{Normal}(0, 1)$ and $U \sim \mathrm{Uniform}(0, 1)$, then Z/U follows slash distribution.

Probability density function

$$f(x) = \frac{\phi(0) - \phi(x)}{x^2}$$

Cumulative distribution function

$$F(x) = \left\{ \begin{array}{ll} \Phi(x) - \frac{\phi(0)-\phi(x)}{x} & x \ne 0 \\ \frac{1}{2} & x = 0 \end{array} \right.$$

Examples

```
x <- rslash(1e5, 5, 3)
xx <- seq(-100, 100, by = 0.001)
hist(x, 1e5, freq = FALSE, xlim = c(-100, 100))
lines(xx, dslash(xx, 5, 3), col = "red")
hist(pslash(x, 5, 3))
plot(ecdf(x), xlim = c(-100, 100))
lines(xx, pslash(xx, 5, 3), col = "red", lwd = 2)
```

`Triangular` *Triangular distribution*

Description

Density, distribution function, quantile function and random generation for the triangular distribution.

Usage

```
dtriang(x, a = -1, b = 1, c = (a + b)/2, log = FALSE)

ptriang(q, a = -1, b = 1, c = (a + b)/2, lower.tail = TRUE,
  log.p = FALSE)

qtriang(p, a = -1, b = 1, c = (a + b)/2, lower.tail = TRUE,
  log.p = FALSE)

rtriang(n, a = -1, b = 1, c = (a + b)/2)
```

Arguments

`x, q`	vector of quantiles.
`a, b, c`	minimum, maximum and mode of the distribution.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \begin{cases} \frac{2(x-a)}{(b-a)(c-a)} & x < c \\ \frac{2}{b-a} & x = c \\ \frac{2(b-x)}{(b-a)(b-c)} & x > c \end{cases}$$

Cumulative distribution function

$$F(x) = \begin{cases} \frac{(x-a)^2}{(b-a)(c-a)} & x \le c \\ 1 - \frac{(b-x)^2}{(b-a)(b-c)} & x > c \end{cases}$$

Quantile function

$$F^{-1}(p) = \begin{cases} a + \sqrt{p \times (b-a)(c-a)} & p \le \frac{c-a}{b-a} \\ b - \sqrt{(1-p)(b-a)(b-c)} & p > \frac{c-a}{b-a} \end{cases}$$

For random generation MINMAX method described by Stein and Keblis (2009) is used.

References

Forbes, C., Evans, M. Hastings, N., & Peacock, B. (2011). Statistical Distributions. John Wiley & Sons.

Stein, W. E., & Keblis, M. F. (2009). A new method to simulate the triangular distribution. Mathematical and computer modelling, 49(5), 1143-1147.

Examples

```
x <- rtriang(1e5, 5, 7, 6)
xx <- seq(-10, 10, by = 0.001)
hist(x, 100, freq = FALSE)
lines(xx, dtriang(xx, 5, 7, 6), col = "red")
hist(ptriang(x, 5, 7, 6))
plot(ecdf(x))
lines(xx, ptriang(xx, 5, 7, 6), col = "red", lwd = 2)
```

TruncNormal *Truncated normal distribution*

Description

Density, distribution function, quantile function and random generation for the truncated normal distribution.

Usage

```
dtnorm(x, mean = 0, sd = 1, a = -Inf, b = Inf, log = FALSE)

ptnorm(q, mean = 0, sd = 1, a = -Inf, b = Inf, lower.tail = TRUE,
  log.p = FALSE)

qtnorm(p, mean = 0, sd = 1, a = -Inf, b = Inf, lower.tail = TRUE,
  log.p = FALSE)

rtnorm(n, mean = 0, sd = 1, a = -Inf, b = Inf)
```

Arguments

x, q	vector of quantiles.
mean, sd	location and scale parameters. Scale must be positive.
a, b	minimal and maximal boundries for truncation (-Inf and Inf by default).
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \frac{\phi(\frac{x-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})}$$

Cumulative distribution function

$$F(x) = \frac{\Phi(\frac{x-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})}$$

Quantile function

$$F^{-1}(p) = \Phi^{-1}\left(\Phi\left(\frac{a-\mu}{\sigma}\right) + p \times \left[\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)\right]\right)$$

For random generation algorithm described by Robert (1995) is used.

References

Robert, C.P. (1995). Simulation of truncated normal variables. Statistics and Computing 5(2): 121-125. http://arxiv.org/abs/0907.4010

Burkardt, J. (17 October 2014). The Truncated Normal Distribution. Florida State University. http://people.sc.fsu.edu/~jburkardt/presentations/truncated_normal.pdf

Examples

```
x <- rtnorm(1e5, 5, 3, b = 7)
xx <- seq(-10, 10, by = 0.001)
hist(x, 100, freq = FALSE)
lines(xx, dtnorm(xx, 5, 3, b = 7), col = "red")
hist(ptnorm(x, 5, 3, b = 7))
plot(ecdf(x))
lines(xx, ptnorm(xx, 5, 3, b = 7), col = "red", lwd = 2)

R <- 1e5
partmp <- par(mfrow = c(2,4), mar = c(2,2,2,2))

hist(rtnorm(R), freq= FALSE, main = "", xlab = "", ylab = "")
lines(xx, dtnorm(xx), col = "red")

hist(rtnorm(R, a = 0), freq= FALSE, main = "", xlab = "", ylab = "")
lines(xx, dtnorm(xx, a = 0), col = "red")

hist(rtnorm(R, b = 0), freq= FALSE, main = "", xlab = "", ylab = "")
lines(xx, dtnorm(xx, b = 0), col = "red")

hist(rtnorm(R, a = 0, b = 1), freq= FALSE, main = "", xlab = "", ylab = "")
lines(xx, dtnorm(xx, a = 0, b = 1), col = "red")
```

```
hist(rtnorm(R, a = -1, b = 0), freq= FALSE, main = "", xlab = "", ylab = "")
lines(xx, dtnorm(xx, a = -1, b = 0), col = "red")

hist(rtnorm(R, mean = -6, a = 0), freq= FALSE, main = "", xlab = "", ylab = "")
lines(xx, dtnorm(xx, mean = -6, a = 0), col = "red")

hist(rtnorm(R, mean = 8, b = 0), freq= FALSE, main = "", xlab = "", ylab = "")
lines(xx, dtnorm(xx, mean = 8, b = 0), col = "red")

hist(rtnorm(R, a = 3, b = 5), freq= FALSE, main = "", xlab = "", ylab = "")
lines(xx, dtnorm(xx, a = 3, b = 5), col = "red")

par(partmp)
```

TruncPoisson — *Truncated Poisson distribution*

Description

Density, distribution function, quantile function and random generation for the truncated Poisson distribution.

Usage

```
dtpois(x, lambda, s = 0, log = FALSE)

ptpois(q, lambda, s = 0, lower.tail = TRUE, log.p = FALSE)

qtpois(p, lambda, s = 0, lower.tail = TRUE, log.p = FALSE)

rtpois(n, lambda, s = 0)
```

Arguments

`x, q`	vector of quantiles.
`lambda`	vector of (non-negative) means.
`s`	truncation point (non-negtive); `s=0` (default) for zero-truncated Poisson, otherwise values greater than `s` are truncated.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

References

Plackett, R.L. (1953). The truncated Poisson distribution. Biometrics, 9(4), 485-488.

Singh, J. (1978). A characterization of positive Poisson distribution and its statistical application. SIAM Journal on Applied Mathematics, 34(3), 545-548.

Dalgaard, P. (May 1, 2005). [R] simulate zero-truncated Poisson distribution. R-help mailing list. https://stat.ethz.ch/pipermail/r-help/2005-May/070680.html

Examples

```
x <- rtpois(1e5, 14, 16)
xx <- seq(-1, 20)
plot(prop.table(table(x)))
lines(xx, dtpois(xx, 14, 16), col = "red")
hist(ptpois(x, 14, 16))
plot(ecdf(x))
lines(xx, ptpois(xx, 14, 16), col = "red", lwd = 2)
uu <- seq(0, 1, by = 0.001)
lines(qtpois(uu, 14, 16), uu, col = "blue")

# Zero-truncated Poisson

x <- rtpois(1e5, 5, 0)
xx <- seq(-1, 50)
plot(prop.table(table(x)))
lines(xx, dtpois(xx, 5, 0), col = "red")
hist(ptpois(x, 5, 0))
plot(ecdf(x))
lines(xx, ptpois(xx, 5, 0), col = "red", lwd = 2)
lines(qtpois(uu, 5, 0), uu, col = "blue")
```

TuckeyLambda *Tuckey lambda distribution*

Description

Quantile function, and random generation for the Tuckey lambda distribution.

Usage

```
qtlambda(p, lambda, lower.tail = TRUE, log.p = FALSE)

rtlambda(n, lambda)
```

Arguments

- `p` — vector of probabilities.
- `lambda` — shape parameter.
- `lower.tail` — logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
- `log.p` — logical; if TRUE, probabilities p are given as log(p).
- `n` — number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Tukey lambda distribution is a continuous probability distribution defined in terms of its quantile function. It is typically used to identify other distributions.

Quantile function:

$$F^{-1}(p) = \begin{cases} \frac{1}{\lambda}[p^\lambda - (1-p)^\lambda] & \lambda \neq 0 \\ \log(\frac{p}{1-p}) & \lambda = 0 \end{cases}$$

References

Joiner, B.L., & Rosenblatt, J.R. (1971). Some properties of the range in samples from Tukey's symmetric lambda distributions. Journal of the American Statistical Association, 66(334), 394-399.

Hastings Jr, C., Mosteller, F., Tukey, J.W., & Winsor, C.P. (1947). Low moments for small samples: a comparative study of order statistics. The Annals of Mathematical Statistics, 413-426.

Examples

```
pp = seq(0, 1, by = 0.001)
partmp <- par(mfrow = c(2,3))
plot(qtlambda(pp, -1), pp, type = "l", main = "lambda = -1 (Cauchy)")
plot(qtlambda(pp, 0), pp, type = "l", main = "lambda = 0 (logistic)")
plot(qtlambda(pp, 0.14), pp, type = "l", main = "lambda = 0.14 (normal)")
plot(qtlambda(pp, 0.5), pp, type = "l", main = "lambda = 0.5 (concave)")
plot(qtlambda(pp, 1), pp, type = "l", main = "lambda = 1 (uniform)")
plot(qtlambda(pp, 2), pp, type = "l", main = "lambda = 2 (uniform)")

hist(rtlambda(1e5, -1), freq = FALSE, main = "lambda = -1 (Cauchy)")
hist(rtlambda(1e5, 0), freq = FALSE, main = "lambda = 0 (logistic)")
hist(rtlambda(1e5, 0.14), freq = FALSE, main = "lambda = 0.14 (normal)")
hist(rtlambda(1e5, 0.5), freq = FALSE, main = "lambda = 0.5 (concave)")
hist(rtlambda(1e5, 1), freq = FALSE, main = "lambda = 1 (uniform)")
hist(rtlambda(1e5, 2), freq = FALSE, main = "lambda = 2 (uniform)")
par(partmp)
```

`Wald` *Wald (inverse Gaussian) distribution*

Description

Density, distribution function and random generation for the Wald distribution.

Usage

```
dwald(x, mu, lambda, log = FALSE)

pwald(q, mu, lambda, lower.tail = TRUE, log.p = FALSE)

rwald(n, mu, lambda)
```

Arguments

`x, q`	vector of quantiles.
`mu, lambda`	location and shape parameters. Scale must be positive.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`p`	vector of probabilities.

Details

Probability density function

$$f(x) = \sqrt{\frac{\lambda}{2\pi x^3}} \exp\left(\frac{-\lambda(x-\mu)^2}{2\mu^2 x}\right)$$

Cumulative distribution function

$$F(x) = \Phi\left(\sqrt{\frac{\lambda}{x}}\left(\frac{x}{\mu} - 1\right)\right) + \exp\left(\frac{2\lambda}{\mu}\right)\Phi\left(\sqrt{\frac{\lambda}{x}}\left(\frac{x}{\mu} + 1\right)\right)$$

Examples

```
x <- rwald(1e5, 5, 16)
xx <- seq(0, 100, by = 0.001)
hist(x, 100, freq = FALSE)
lines(xx, dwald(xx, 5, 16), col = "red")
hist(pwald(x, 5, 16))
plot(ecdf(x))
lines(xx, pwald(xx, 5, 16), col = "red", lwd = 2)
```

ZIB *Zero-inflated binomial distribution*

Description

Probability mass function and random generation for the zero-inflated binomial distribution.

Usage

```
dzib(x, size, prob, pi, log = FALSE)

pzib(q, size, prob, pi, lower.tail = TRUE, log.p = FALSE)

qzib(p, size, prob, pi, lower.tail = TRUE, log.p = FALSE)

rzib(n, size, prob, pi)
```

Arguments

`x, q`	vector of quantiles.
`size`	number of trials (zero or more).
`prob`	probability of success in each trial. `0 < prob <= 1`.
`pi`	probability of extra zeros.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \begin{cases} \pi + (1-\pi)(1-p)^n & x = 0 \\ (1-\pi)\binom{n}{x}p^x(1-p)^{n-x} & x > 0 \end{cases}$$

See Also

Binomial

Examples

```
x <- rzib(1e5, 10, 0.6, 0.33)
xx <- -2:20
plot(prop.table(table(x)), type = "h")
lines(xx, dzib(xx, 10, 0.6, 0.33), col = "red")
plot(ecdf(x))
lines(xx, pzib(xx, 10, 0.6, 0.33), col = "red")
```

ZINB *Zero-inflated negative binomial distribution*

Description

Probability mass function and random generation for the zero-inflated nagative binomial distribution.

Usage

```
dzinb(x, size, prob, pi, log = FALSE)

pzinb(q, size, prob, pi, lower.tail = TRUE, log.p = FALSE)

qzinb(p, size, prob, pi, lower.tail = TRUE, log.p = FALSE)

rzinb(n, size, prob, pi)
```

Arguments

x, q	vector of quantiles.
size	target for number of successful trials, or dispersion parameter (the shape parameter of the gamma mixing distribution). Must be strictly positive, need not be integer.
prob	probability of success in each trial. 0 < prob <= 1.
pi	probability of extra zeros.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \begin{cases} \pi + (1-\pi)p^r & x = 0 \\ (1-\pi)\binom{x+r-1}{x}p^r(1-p)^x & x > 0 \end{cases}$$

See Also

NegBinomial

Examples

```
x <- rzinb(1e5, 100, 0.6, 0.33)
xx <- -2:200
plot(prop.table(table(x)), type = "h")
lines(xx, dzinb(xx, 100, 0.6, 0.33), col = "red")
plot(ecdf(x))
lines(xx, pzinb(xx, 100, 0.6, 0.33), col = "red")
```

ZIP *Zero-inflated Poisson distribution*

Description

Probability mass function and random generation for the zero-inflated Poisson distribution.

Usage

```
dzip(x, lambda, pi, log = FALSE)

pzip(q, lambda, pi, lower.tail = TRUE, log.p = FALSE)

qzip(p, lambda, pi, lower.tail = TRUE, log.p = FALSE)

rzip(n, lambda, pi)
```

Arguments

`x, q`	vector of quantiles.
`lambda`	vector of (non-negative) means.
`pi`	probability of extra zeros.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(x) = \left\{ \begin{array}{ll} \pi + (1-\pi)e^{-\lambda} & x = 0 \\ (1-\pi)\frac{\lambda^x e^{-\lambda}}{x!} & x > 0 \end{array} \right.$$

See Also

Poisson

Examples

```
x <- rzip(1e5, 6, 0.33)
xx <- -2:20
plot(prop.table(table(x)), type = "h")
lines(xx, dzip(xx, 6, 0.33), col = "red")
plot(ecdf(x))
lines(xx, pzip(xx, 6, 0.33), col = "red")
```

Index