Package ‘distributional’

May 8, 2026

Title Vectorised Probability Distributions

Version 0.7.0

Description Vectorised distribution objects with tools for manipulating, visualising, and using probability distributions. Designed to allow model prediction outputs to return distributions rather than their parameters, allowing users to directly interact with predictive distributions in a data-oriented workflow. In addition to providing generic replacements for p/d/q/r functions, other useful statistics can be computed including means, variances, intervals, and highest density regions.

License GPL-3

Depends R (>= 4.0.0)

Imports vctrs (>= 0.3.0), rlang (>= 0.4.5), generics, stats, numDeriv, utils, lifecycle, pillar

Suggests testthat (>= 2.1.0), covr, mvtnorm, actuar (>= 2.0.0), evd, ggdist, ggplot2, gk, LaplacesDemon, pkgdown

RdMacros lifecycle

URL https://pkg.mitchelloharawild.com/distributional/, https://github.com/mitchelloharawild/distributional

BugReports https://github.com/mitchelloharawild/distributional/issues

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.3

NeedsCompilation no

Author Mitchell O'Hara-Wild [aut, cre] (ORCID: <https://orcid.org/0000-0001-6729-7695>),
Matthew Kay [aut] (ORCID: <https://orcid.org/0000-0001-9446-0419>),
Alex Hayes [aut] (ORCID: <https://orcid.org/0000-0002-4985-5160>),
Rob Hyndman [aut] (ORCID: <https://orcid.org/0000-0002-2140-5352>),
Earo Wang [ctb] (ORCID: <https://orcid.org/0000-0001-6448-5260>),
Vencislav Popov [ctb] (ORCID: <https://orcid.org/0000-0002-8073-4199>)

Maintainer Mitchell O'Hara-Wild <mail@mitchelloharawild.com>

Repository CRAN

Date/Publication 2026-03-17 08:30:44 UTC

Contents

`cdf` *The cumulative distribution function*

Description

[Stable]

Usage

```
cdf(x, q, ..., log = FALSE)

## S3 method for class 'distribution'
cdf(x, q, ...)
```

Arguments

`x`	The distribution(s).
`q`	The quantile at which the cdf is calculated.
`...`	Additional arguments passed to methods.
`log`	If `TRUE`, probabilities will be given as log probabilities.

`covariance` *Covariance*

Description

[Stable]

A generic function for computing the covariance of an object.

Usage

```
covariance(x, ...)
```

Arguments

`x`	An object.
`...`	Additional arguments used by methods.

See Also

`covariance.distribution()`, `variance()`

`covariance.distribution` *Covariance of a probability distribution*

Description

[Stable]

Returns the empirical covariance of the probability distribution. If the method does not exist, the covariance of a random sample will be returned.

Usage

```
## S3 method for class 'distribution'
covariance(x, ...)
```

Arguments

`x`	The distribution(s).
`...`	Additional arguments used by methods.

`density.distribution` *The probability density/mass function*

Description

[Stable]

Computes the probability density function for a continuous distribution, or the probability mass function for a discrete distribution.

Usage

```
## S3 method for class 'distribution'
density(x, at, ..., log = FALSE)
```

Arguments

`x`	The distribution(s).
`at`	The point at which to compute the density/mass.
`...`	Additional arguments passed to methods.
`log`	If `TRUE`, probabilities will be given as log probabilities.

`dist_bernoulli` *The Bernoulli distribution*

Description

[Stable]

Bernoulli distributions are used to represent events like coin flips when there is single trial that is either successful or unsuccessful. The Bernoulli distribution is a special case of the `Binomial()` distribution with `n = 1`.

Usage

```
dist_bernoulli(prob)
```

Arguments

`prob`	The probability of success on each trial, `prob` can be any value in `[0, 1]`.

Details

We recommend reading this documentation on pkgdown which renders math nicely. `https://pkg.mitchelloharawild.com/distributional/reference/dist_bernoulli.html`

In the following, let X be a Bernoulli random variable with parameter `prob` = p. Some textbooks also define $q = 1 - p$, or use π instead of p.

The Bernoulli probability distribution is widely used to model binary variables, such as 'failure' and 'success'. The most typical example is the flip of a coin, when p is thought as the probability of flipping a head, and $q = 1 - p$ is the probability of flipping a tail.

Support: $\{0, 1\}$

Mean: p

Variance: $p \cdot (1 - p) = p \cdot q$

Probability mass function (p.m.f):

$$P(X = x) = p^x(1 - p)^{1-x} = p^x q^{1-x}$$

Cumulative distribution function (c.d.f):

$$P(X \le x) = \begin{cases} 0 & x < 0 \\ 1 - p & 0 \le x < 1 \\ 1 & x \ge 1 \end{cases}$$

Moment generating function (m.g.f):

$$E(e^{tX}) = (1 - p) + pe^t$$

Skewness:

$$\frac{1-2p}{\sqrt{p(1-p)}} = \frac{q-p}{\sqrt{pq}}$$

Excess Kurtosis:

$$\frac{1-6p(1-p)}{p(1-p)} = \frac{1-6pq}{pq}$$

See Also

stats::Binomial

Examples

```
dist <- dist_bernoulli(prob = c(0.05, 0.5, 0.3, 0.9, 0.1))

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

`dist_beta` *The Beta distribution*

Description

[Stable]

The Beta distribution is a continuous probability distribution defined on the interval [0, 1], commonly used to model probabilities and proportions.

Usage

```
dist_beta(shape1, shape2)
```

Arguments

`shape1, shape2`	The non-negative shape parameters of the Beta distribution.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_beta.html

In the following, let X be a Beta random variable with parameters `shape1` = α and `shape2` = β.

Support: $x \in [0, 1]$

Mean: $\frac{\alpha}{\alpha+\beta}$

Variance: $\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$

Probability density function (p.d.f):

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

where $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ is the Beta function.

Cumulative distribution function (c.d.f):

$$F(x) = I_x(alpha, beta) = \frac{B(x;\alpha,\beta)}{B(\alpha,\beta)}$$

where $I_x(\alpha,\beta)$ is the regularized incomplete beta function and $B(x;\alpha,\beta) = \int_0^x t^{\alpha-1}(1-t)^{\beta-1}dt$.

Moment generating function (m.g.f):

The moment generating function does not have a simple closed form, but the moments can be calculated as:

$$E(X^k) = \prod_{r=0}^{k-1} \frac{\alpha+r}{\alpha+\beta+r}$$

See Also

stats::Beta

Examples

```
dist <- dist_beta(shape1 = c(0.5, 5, 1, 2, 2), shape2 = c(0.5, 1, 3, 2, 5))

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)
```

```
quantile(dist, 0.7)
```

`dist_binomial` *The Binomial distribution*

Description

[Stable]

Binomial distributions are used to represent situations can that can be thought as the result of n Bernoulli experiments (here the n is defined as the `size` of the experiment). The classical example is n independent coin flips, where each coin flip has probability `p` of success. In this case, the individual probability of flipping heads or tails is given by the Bernoulli(p) distribution, and the probability of having x equal results (x heads, for example), in n trials is given by the Binomial(n, p) distribution. The equation of the Binomial distribution is directly derived from the equation of the Bernoulli distribution.

Usage

```
dist_binomial(size, prob)
```

Arguments

`size`	The number of trials. Must be an integer greater than or equal to one. When `size = 1L`, the Binomial distribution reduces to the Bernoulli distribution. Often called `n` in textbooks.
`prob`	The probability of success on each trial, `prob` can be any value in `[0, 1]`.

Details

We recommend reading this documentation on pkgdown which renders math nicely. `https://pkg.mitchelloharawild.com/distributional/reference/dist_binomial.html`

The Binomial distribution comes up when you are interested in the portion of people who do a thing. The Binomial distribution also comes up in the sign test, sometimes called the Binomial test (see `stats::binom.test()`), where you may need the Binomial C.D.F. to compute p-values.

In the following, let X be a Binomial random variable with parameter `size` = n and `p` = p. Some textbooks define $q = 1 - p$, or called π instead of p.

Support: $\{0, 1, 2, ..., n\}$

Mean: np

Variance: $np \cdot (1 - p) = np \cdot q$

Probability mass function (p.m.f):

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

Cumulative distribution function (c.d.f):

$$P(X \le k) = \sum_{i=0}^{\lfloor k \rfloor} \binom{n}{i} p^i (1-p)^{n-i}$$

Moment generating function (m.g.f):

$$E(e^{tX}) = (1 - p + pe^t)^n$$

Skewness:

$$\frac{1-2p}{\sqrt{np(1-p)}}$$

Excess kurtosis:

$$\frac{1-6p(1-p)}{np(1-p)}$$

See Also

stats::Binomial

Examples

```
dist <- dist_binomial(size = 1:5, prob = c(0.05, 0.5, 0.3, 0.9, 0.1))

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

dist_burr — *The Burr distribution*

Description

[Stable]

The Burr distribution (Type XII) is a flexible continuous probability distribution often used for modeling income distributions, reliability data, and failure times.

Usage

```
dist_burr(shape1, shape2, rate = 1, scale = 1/rate)
```

Arguments

`shape1`, `shape2`, `scale`	parameters. Must be strictly positive.
`rate`	an alternative way to specify the scale.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_burr.html

In the following, let X be a Burr random variable with parameters `shape1` = α, `shape2` = γ, and `rate` = λ.

Support: $x \in (0, \infty)$

Mean: $\frac{\lambda^{-1/\alpha}\gamma B(\gamma-1/\alpha, 1+1/\alpha)}{\gamma}$ (for $\alpha\gamma > 1$)

Variance: $\frac{\lambda^{-2/\alpha}\gamma B(\gamma-2/\alpha, 1+2/\alpha)}{\gamma} - \mu^2$ (for $\alpha\gamma > 2$)

Probability density function (p.d.f):

$$f(x) = \alpha\gamma\lambda x^{\alpha-1}(1 + \lambda x^{\alpha})^{-\gamma-1}$$

Cumulative distribution function (c.d.f):

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

Quantile function:

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

Moment generating function (m.g.f):

Does not exist in closed form.

See Also

actuar::Burr

Examples

```
dist <- dist_burr(shape1 = c(1,1,1,2,3,0.5), shape2 = c(1,2,3,1,1,2))
dist


mean(dist)
variance(dist)
support(dist)
generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

---

dist_categorical *The Categorical distribution*

Description

[Stable]

Categorical distributions are used to represent events with multiple outcomes, such as what number appears on the roll of a dice. This is also referred to as the 'generalised Bernoulli' or 'multinoulli' distribution. The Categorical distribution is a special case of the `Multinomial()` distribution with `n = 1`.

Usage

```
dist_categorical(prob, outcomes = NULL)
```

Arguments

`prob`	A list of probabilities of observing each outcome category.
`outcomes`	The list of vectors where each value represents each outcome.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_categorical.html

In the following, let X be a Categorical random variable with probability parameters `prob` = $\{p_1, p_2, \ldots, p_k\}$.

The Categorical probability distribution is widely used to model the occurance of multiple events. A simple example is the roll of a dice, where $p = \{1/6, 1/6, 1/6, 1/6, 1/6, 1/6\}$ giving equal chance of observing each number on a 6 sided dice.

Support: $\{1, \ldots, k\}$

Mean: Not defined for unordered categories. For ordered categories with integer outcomes $\{1, 2, \ldots, k\}$, the mean is:

$$E(X) = \sum_{i=1}^{k} i \cdot p_i$$

Variance: Not defined for unordered categories. For ordered categories with integer outcomes $\{1, 2, \ldots, k\}$, the variance is:

$$\mathrm{Var}(X) = \sum_{i=1}^{k} i^2 \cdot p_i - \left(\sum_{i=1}^{k} i \cdot p_i\right)^2$$

Probability mass function (p.m.f):

$$P(X = i) = p_i$$

Cumulative distribution function (c.d.f):

The c.d.f is undefined for unordered categories. For ordered categories with outcomes $x_1 < x_2 < \ldots < x_k$, the c.d.f is:

$$P(X \le x_j) = \sum_{i=1}^{j} p_i$$

Moment generating function (m.g.f):

$$E(e^{tX}) = \sum_{i=1}^{k} e^{tx_i} \cdot p_i$$

Skewness: Approximated numerically for ordered categories.

Kurtosis: Approximated numerically for ordered categories.

See Also

stats::Multinomial

Examples

```
dist <- dist_categorical(prob = list(c(0.05, 0.5, 0.15, 0.2, 0.1), c(0.3, 0.1, 0.6)))

dist

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

# The outcomes aren't ordered, so many statistics are not applicable.
cdf(dist, 0.6)
quantile(dist, 0.7)
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

# Some of these statistics are meaningful for ordered outcomes
dist <- dist_categorical(list(rpois(26, 3)), list(ordered(letters)))
dist
cdf(dist, "m")
quantile(dist, 0.5)

dist <- dist_categorical(
  prob = list(c(0.05, 0.5, 0.15, 0.2, 0.1), c(0.3, 0.1, 0.6)),
  outcomes = list(letters[1:5], letters[24:26])
)

generate(dist, 10)

density(dist, "a")
density(dist, "z", log = TRUE)
```

`dist_cauchy` *The Cauchy distribution*

Description

[Stable]

The Cauchy distribution is the student's t distribution with one degree of freedom. The Cauchy distribution does not have a well defined mean or variance. Cauchy distributions often appear as priors in Bayesian contexts due to their heavy tails.

Usage

```
dist_cauchy(location, scale)
```

Arguments

`location, scale` location and scale parameters.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_cauchy.html

In the following, let X be a Cauchy variable with mean `location` $= x_0$ and `scale` $= \gamma$.

Support: R, the set of all real numbers

Mean: Undefined.

Variance: Undefined.

Probability density function (p.d.f):

$$f(x) = \frac{1}{\pi\gamma \left[1 + \left(\frac{x-x_0}{\gamma}\right)^2\right]}$$

Cumulative distribution function (c.d.f):

$$F(t) = \frac{1}{\pi} \arctan\left(\frac{t-x_0}{\gamma}\right) + \frac{1}{2}$$

Moment generating function (m.g.f):

Does not exist.

See Also

stats::Cauchy

Examples

```
dist <- dist_cauchy(location = c(0, 0, 0, -2), scale = c(0.5, 1, 2, 1))

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

`dist_chisq` *The (non-central) Chi-Squared Distribution*

Description

[Stable]

Chi-square distributions show up often in frequentist settings as the sampling distribution of test statistics, especially in maximum likelihood estimation settings.

Usage

```
dist_chisq(df, ncp = 0)
```

Arguments

df	Degrees of freedom (non-centrality parameter). Can be any positive real number.
ncp	Non-centrality parameter. Can be any non-negative real number. Defaults to 0 (central chi-squared distribution).

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_chisq.html

In the following, let X be a χ^2 random variable with `df` = k and `ncp` = λ.

Support: R^+, the set of positive real numbers

Mean: $k + \lambda$

Variance: $2(k + 2\lambda)$

Probability density function (p.d.f):

For the central chi-squared distribution ($\lambda = 0$):

$$f(x) = \frac{1}{2^{k/2}\Gamma(k/2)} x^{k/2-1} e^{-x/2}$$

For the non-central chi-squared distribution ($\lambda > 0$):

$$f(x) = \frac{1}{2} e^{-(x+\lambda)/2} \left(\frac{x}{\lambda}\right)^{k/4-1/2} I_{k/2-1}\left(\sqrt{\lambda x}\right)$$

where $I_\nu(z)$ is the modified Bessel function of the first kind.

Cumulative distribution function (c.d.f):

For the central chi-squared distribution ($\lambda = 0$):

$$F(x) = \frac{\gamma(k/2, x/2)}{\Gamma(k/2)} = P(k/2, x/2)$$

where $\gamma(s, x)$ is the lower incomplete gamma function and $P(s, x)$ is the regularized gamma function.

For the non-central chi-squared distribution ($\lambda > 0$):

$$F(x) = \sum_{j=0}^{\infty} \frac{e^{-\lambda/2}(\lambda/2)^j}{j!} P(k/2 + j, x/2)$$

This is approximated numerically.

Moment generating function (m.g.f):

For the central chi-squared distribution ($\lambda = 0$):

$$E(e^{tX}) = (1 - 2t)^{-k/2}, \quad t < 1/2$$

For the non-central chi-squared distribution ($\lambda > 0$):

$$E(e^{tX}) = \frac{e^{\lambda t/(1-2t)}}{(1 - 2t)^{k/2}}, \quad t < 1/2$$

Skewness:

$$\gamma_1 = \frac{2^{3/2}(k + 3\lambda)}{(k + 2\lambda)^{3/2}}$$

For the central case ($\lambda = 0$), this simplifies to $\sqrt{8/k}$.

Excess Kurtosis:

$$\gamma_2 = \frac{12(k + 4\lambda)}{(k + 2\lambda)^2}$$

For the central case ($\lambda = 0$), this simplifies to $12/k$.

See Also

stats::Chisquare

Examples

```
dist <- dist_chisq(df = c(1,2,3,4,6,9))

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
```

```
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

`dist_degenerate` *The degenerate distribution*

Description

[Stable]

The degenerate distribution takes a single value which is certain to be observed. It takes a single parameter, which is the value that is observed by the distribution.

Usage

```
dist_degenerate(x)
```

Arguments

x	The value of the distribution (location parameter). Can be any real number.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_degenerate.html

In the following, let X be a degenerate random variable with value x = k_0.

Support: $\{k_0\}$, a single point

Mean: $\mu = k_0$

Variance: $\sigma^2 = 0$

Probability density function (p.d.f):

$$f(x) = 1 \text{ for } x = k_0$$
$$f(x) = 0 \text{ for } x \neq k_0$$

Cumulative distribution function (c.d.f):

$$F(t) = 0 \text{ for } t < k_0$$
$$F(t) = 1 \text{ for } t \geq k_0$$

Moment generating function (m.g.f):

$$E(e^{tX}) = e^{k_0 t}$$

Skewness: Undefined (NA)

Excess Kurtosis: Undefined (NA)

See Also

stats::Distributions

Examples

```
dist <- dist_degenerate(x = 1:5)

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

---

dist_dirichlet *The Dirichlet distribution*

Description

[Stable]

The Dirichlet distribution is a multivariate generalisation of the Beta distribution. It is the conjugate prior of the Categorical and Multinomial distributions, and describes a probability distribution over the $(k-1)$-simplex — the set of k-dimensional vectors whose components are non-negative and sum to one.

Usage

```
dist_dirichlet(alpha)
```

Arguments

alpha	A list of positive numeric concentration vectors.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_dirichlet.html

In the following, let $\mathbf{X} = (X_1, \ldots, X_k)$ be a Dirichlet random variable with concentration parameter alpha = $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$, where each $\alpha_i > 0$.

Support: $\mathbf{x}$ on the $(k-1)$-simplex, i.e. $x_i \geq 0$ and $\sum_{i=1}^{k} x_i = 1$.

Mean: $E(X_i) = \frac{\alpha_i}{\alpha_0}$ where $\alpha_0 = \sum_{i=1}^{k} \alpha_i$.

Variance:

$$\mathrm{Var}(X_i) = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)}$$

Covariance:

$$\mathrm{Cov}(X_i, X_j) = \frac{-\alpha_i \alpha_j}{\alpha_0^2(\alpha_0 + 1)}, \quad i \neq j$$

Probability density function (p.d.f):

$$f(\mathbf{x}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{k} x_i^{\alpha_i - 1}$$

where $B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma(\alpha_0)}$ is the multivariate Beta function.

See Also

`LaplacesDemon::ddirichlet()`, `LaplacesDemon::rdirichlet()`

Examples

```
dist <- dist_dirichlet(alpha = list(c(2, 5, 3)))
dist


mean(dist)
variance(dist)
support(dist)
generate(dist, 10)

density(dist, cbind(0.2, 0.5, 0.3))
density(dist, cbind(0.2, 0.5, 0.3), log = TRUE)
```

`dist_exponential` *The Exponential Distribution*

Description

[Stable]

Exponential distributions are frequently used to model waiting times and the time between events in a Poisson process.

Usage

```
dist_exponential(rate)
```

Arguments

`rate`	vector of rates.

Details

We recommend reading this documentation on pkgdown which renders math nicely. `https://pkg.mitchelloharawild.com/distributional/reference/dist_exponential.html`

In the following, let X be an Exponential random variable with parameter `rate` = λ.

Support: $x \in [0, \infty)$

Mean: $\frac{1}{\lambda}$

Variance: $\frac{1}{\lambda^2}$

Probability density function (p.d.f):

$$f(x) = \lambda e^{-\lambda x}$$

Cumulative distribution function (c.d.f):

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

Moment generating function (m.g.f):

$$E(e^{tX}) = \frac{\lambda}{\lambda - t}, \quad t < \lambda$$

See Also

stats::Exponential

Examples

```
dist <- dist_exponential(rate = c(2, 1, 2/3))

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)
```

```
quantile(dist, 0.7)
```

---

dist_f *The F Distribution*

Description

[Stable]

The F distribution is commonly used in statistical inference, particularly in the analysis of variance (ANOVA), testing the equality of variances, and in regression analysis. It arises as the ratio of two scaled chi-squared distributions divided by their respective degrees of freedom.

Usage

```
dist_f(df1, df2, ncp = NULL)
```

Arguments

df1	Degrees of freedom for the numerator. Can be any positive number.
df2	Degrees of freedom for the denominator. Can be any positive number.
ncp	Non-centrality parameter. If `NULL` (default), the central F distribution is used. If specified, must be non-negative.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_f.html

In the following, let X be an F random variable with numerator degrees of freedom `df1` = d_1 and denominator degrees of freedom `df2` = d_2.

Support: $x \in (0, \infty)$

Mean:

For the central F distribution (`ncp = NULL`):

$$E(X) = \frac{d_2}{d_2 - 2}$$

for $d_2 > 2$, otherwise undefined.

For the non-central F distribution with non-centrality parameter `ncp` = λ:

$$E(X) = \frac{d_2(d_1 + \lambda)}{d_1(d_2 - 2)}$$

for $d_2 > 2$, otherwise undefined.

Variance:

For the central F distribution (`ncp = NULL`):

$$\text{Var}(X) = \frac{2d_2^2(d_1 + d_2 - 2)}{d_1(d_2 - 2)^2(d_2 - 4)}$$

for $d_2 > 4$, otherwise undefined.

For the non-central F distribution with non-centrality parameter `ncp` = λ:

$$\text{Var}(X) = \frac{2d_2^2}{d_1^2} \cdot \frac{(d_1 + \lambda)^2 + (d_1 + 2\lambda)(d_2 - 2)}{(d_2 - 2)^2(d_2 - 4)}$$

for $d_2 > 4$, otherwise undefined.

Skewness:

For the central F distribution (`ncp = NULL`):

$$\text{Skew}(X) = \frac{(2d_1 + d_2 - 2)\sqrt{8(d_2 - 4)}}{(d_2 - 6)\sqrt{d_1(d_1 + d_2 - 2)}}$$

for $d_2 > 6$, otherwise undefined.

For the non-central F distribution, skewness has no simple closed form and is not computed.

Excess Kurtosis:

For the central F distribution (`ncp = NULL`):

$$\text{Kurt}(X) = \frac{12[d_1(5d_2 - 22)(d_1 + d_2 - 2) + (d_2 - 4)(d_2 - 2)^2]}{d_1(d_2 - 6)(d_2 - 8)(d_1 + d_2 - 2)}$$

for $d_2 > 8$, otherwise undefined.

For the non-central F distribution, kurtosis has no simple closed form and is not computed.

Probability density function (p.d.f):

For the central F distribution (`ncp = NULL`):

$$f(x) = \frac{\sqrt{\frac{(d_1 x)^{d_1} d_2^{d_2}}{(d_1 x + d_2)^{d_1 + d_2}}}}{x\, B(d_1/2, d_2/2)}$$

where $B(\cdot, \cdot)$ is the beta function.

For the non-central F distribution, the density involves an infinite series and is approximated numerically.

Cumulative distribution function (c.d.f):

The c.d.f. does not have a simple closed form expression and is approximated numerically using regularized incomplete beta functions and related special functions.

Moment generating function (m.g.f):

The moment generating function for the F distribution does not exist in general (it diverges for $t > 0$).

See Also

stats::FDist

Examples

```
dist <- dist_f(df1 = c(1,2,5,10,100), df2 = c(1,1,2,1,100))

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

`dist_gamma` *The Gamma distribution*

Description

[Stable]

Several important distributions are special cases of the Gamma distribution. When the shape parameter is 1, the Gamma is an exponential distribution with parameter $1/\beta$. When the $shape = n/2$ and $rate = 1/2$, the Gamma is a equivalent to a chi squared distribution with n degrees of freedom. Moreover, if we have X_1 is $Gamma(\alpha_1, \beta)$ and X_2 is $Gamma(\alpha_2, \beta)$, a function of these two variables of the form $\frac{X_1}{X_1+X_2}$ $Beta(\alpha_1, \alpha_2)$. This last property frequently appears in another distributions, and it has extensively been used in multivariate methods. More about the Gamma distribution will be added soon.

Usage

```
dist_gamma(shape, rate = 1/scale, scale = 1/rate)
```

Arguments

`shape, scale`	shape and scale parameters. Must be positive, `scale` strictly.
`rate`	an alternative way to specify the scale.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_gamma.html

In the following, let X be a Gamma random variable with parameters shape = α and rate = β.

Support: $x \in (0, \infty)$

Mean: $\frac{\alpha}{\beta}$

Variance: $\frac{\alpha}{\beta^2}$

Probability density function (p.m.f):

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

Cumulative distribution function (c.d.f):

$$f(x) = \frac{\Gamma(\alpha, \beta x)}{\Gamma \alpha}$$

Moment generating function (m.g.f):

$$E(e^{tX}) = \left(\frac{\beta}{\beta - t}\right)^\alpha, \, t < \beta$$

See Also

stats::GammaDist

Examples

```
dist <- dist_gamma(shape = c(1,2,3,5,9,7.5,0.5), rate = c(0.5,0.5,0.5,1,2,1,1))

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

dist_geometric *The Geometric Distribution*

Description

[Stable]

The Geometric distribution can be thought of as a generalization of the dist_bernoulli() distribution where we ask: "if I keep flipping a coin with probability p of heads, what is the probability I need k flips before I get my first heads?" The Geometric distribution is a special case of Negative Binomial distribution.

Usage

```
dist_geometric(prob)
```

Arguments

prob	probability of success in each trial. 0 < prob <= 1.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_geometric.html

In the following, let X be a Geometric random variable with success probability prob = p. Note that there are multiple parameterizations of the Geometric distribution.

Support: $\{0, 1, 2, 3, ...\}$

Mean: $\frac{1-p}{p}$

Variance: $\frac{1-p}{p^2}$

Probability mass function (p.m.f):

$$P(X = k) = p(1 - p)^k$$

Cumulative distribution function (c.d.f):

$$P(X \le k) = 1 - (1 - p)^{k+1}$$

Moment generating function (m.g.f):

$$E(e^{tX}) = \frac{pe^t}{1 - (1 - p)e^t}$$

Skewness:

$$\frac{2 - p}{\sqrt{1 - p}}$$

Excess Kurtosis:

$$6 + \frac{p^2}{1-p}$$

See Also

stats::Geometric

Examples

```
dist <- dist_geometric(prob = c(0.2, 0.5, 0.8))

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

---

dist_gev *The Generalized Extreme Value Distribution*

Description

[Stable]

The GEV distribution is widely used in extreme value theory to model the distribution of maxima (or minima) of samples. The parametric form encompasses the Gumbel, Frechet, and reverse Weibull distributions.

Usage

```
dist_gev(location, scale, shape)
```

Arguments

`location`	the location parameter μ of the GEV distribution.
`scale`	the scale parameter σ of the GEV distribution. Must be strictly positive.
`shape`	the shape parameter ξ of the GEV distribution. Determines the tail behavior: $\xi = 0$ gives Gumbel, $\xi > 0$ gives Frechet, $\xi < 0$ gives reverse Weibull.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_gev.html

In the following, let X be a GEV random variable with parameters `location` = μ, `scale` = σ, and `shape` = ξ.

Support:

- $x \in \mathbb{R}$ (all real numbers) if $\xi = 0$
- $x \geq \mu - \sigma/\xi$ if $\xi > 0$
- $x \leq \mu - \sigma/\xi$ if $\xi < 0$

Mean:

$$E(X) = \begin{cases} \mu + \sigma\gamma & \text{if } \xi = 0 \\ \mu + \sigma\frac{\Gamma(1-\xi)-1}{\xi} & \text{if } \xi < 1 \\ \infty & \text{if } \xi \geq 1 \end{cases}$$

where $\gamma \approx 0.5772$ is the Euler-Mascheroni constant and $\Gamma(\cdot)$ is the gamma function.

Median:

$$\text{Median}(X) = \begin{cases} \mu - \sigma\log(\log 2) & \text{if } \xi = 0 \\ \mu + \sigma\frac{(\log 2)^{-\xi}-1}{\xi} & \text{if } \xi \neq 0 \end{cases}$$

Variance:

$$\text{Var}(X) = \begin{cases} \frac{\pi^2\sigma^2}{6} & \text{if } \xi = 0 \\ \frac{\sigma^2}{\xi^2}[\Gamma(1-2\xi) - \Gamma(1-\xi)^2] & \text{if } \xi < 0.5 \\ \infty & \text{if } \xi \geq 0.5 \end{cases}$$

Probability density function (p.d.f):

For $\xi = 0$ (Gumbel):

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

For $\xi \neq 0$:

$$f(x) = \frac{1}{\sigma}\left[1+\xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi-1}\exp\left\{-\left[1+\xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}\right\}$$

where $1 + \xi(x-\mu)/\sigma > 0$.

Cumulative distribution function (c.d.f):

For $\xi = 0$ (Gumbel):

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

For $\xi \neq 0$:

$$F(x) = \exp\left\{-\left[1+\xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}\right\}$$

where $1 + \xi(x - \mu)/\sigma > 0$.

Quantile function:

For $\xi = 0$ (Gumbel):

$$Q(p) = \mu - \sigma \log(-\log p)$$

For $\xi \neq 0$:

$$Q(p) = \mu + \frac{\sigma}{\xi} \left[(-\log p)^{-\xi} - 1\right]$$

References

Jenkinson, A. F. (1955) The frequency distribution of the annual maximum (or minimum) of meteorological elements. *Quart. J. R. Met. Soc.*, **81**, 158–171.

See Also

evd::dgev()

Examples

```
# Create GEV distributions with different shape parameters

# Gumbel distribution (shape = 0)
gumbel <- dist_gev(location = 0, scale = 1, shape = 0)

# Frechet distribution (shape > 0, heavy-tailed)
frechet <- dist_gev(location = 0, scale = 1, shape = 0.3)

# Reverse Weibull distribution (shape < 0, bounded above)
weibull <- dist_gev(location = 0, scale = 1, shape = -0.2)

dist <- c(gumbel, frechet, weibull)
dist

# Statistical properties
mean(dist)
median(dist)
variance(dist)

# Generate random samples
generate(dist, 10)

# Evaluate density
density(dist, 2)
density(dist, 2, log = TRUE)

# Evaluate cumulative distribution
cdf(dist, 4)

# Calculate quantiles
quantile(dist, 0.95)
```

dist_gh *The generalised g-and-h Distribution*

Description

[Stable]

The generalised g-and-h distribution is a flexible distribution used to model univariate data, similar to the g-k distribution. It is known for its ability to handle skewness and heavy-tailed behavior.

Usage

```
dist_gh(A, B, g, h, c = 0.8)
```

Arguments

A	Vector of A (location) parameters.
B	Vector of B (scale) parameters. Must be positive.
g	Vector of g parameters.
h	Vector of h parameters. Must be non-negative.
c	Vector of c parameters (used for generalised g-and-h). Often fixed at 0.8 which is the default.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_gh.html

In the following, let X be a g-and-h random variable with parameters `A` = A, `B` = B, `g` = g, `h` = h, and `c` = c.

Support: $(-\infty, \infty)$

Mean: Does not have a closed-form expression. Approximated numerically.

Variance: Does not have a closed-form expression. Approximated numerically.

Probability density function (p.d.f):

The g-and-h distribution does not have a closed-form expression for its density. The density is approximated numerically from the quantile function. The distribution is defined through its quantile function:

$$Q(u) = A + B\left(1 + c\frac{1 - \exp(-gz(u))}{1 + \exp(-gz(u))}\right)\exp(hz(u)^2/2)z(u)$$

where $z(u) = \Phi^{-1}(u)$ is the standard normal quantile function.

Cumulative distribution function (c.d.f):

Does not have a closed-form expression. The cumulative distribution function is approximated numerically by inverting the quantile function.

Quantile function:

$$Q(p) = A + B\left(1 + c\frac{1 - \exp(-g\Phi^{-1}(p))}{1 + \exp(-g\Phi^{-1}(p))}\right)\exp(h(\Phi^{-1}(p))^2/2)\Phi^{-1}(p)$$

where $\Phi^{-1}(p)$ is the standard normal quantile function.

See Also

gk::dgh(), gk::pgh(), gk::qgh(), gk::rgh(), dist_gk()

Examples

```
dist <- dist_gh(A = 0, B = 1, g = 0, h = 0.5)
dist


mean(dist)
variance(dist)
support(dist)
generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

---

dist_gk *The g-and-k Distribution*

Description

[Stable]

The g-and-k distribution is a flexible distribution often used to model univariate data. It is particularly known for its ability to handle skewness and heavy-tailed behavior.

Usage

```
dist_gk(A, B, g, k, c = 0.8)
```

Arguments

A	Vector of A (location) parameters.
B	Vector of B (scale) parameters. Must be positive.
g	Vector of g parameters.
k	Vector of k parameters. Must be at least -0.5.
c	Vector of c parameters. Often fixed at 0.8 which is the default.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_gk.html

In the following, let X be a g-k random variable with parameters A, B, g, k, and c.

Support: $(-\infty, \infty)$

Mean: Not available in closed form.

Variance: Not available in closed form.

Probability density function (p.d.f):

The g-k distribution does not have a closed-form expression for its density. Instead, it is defined through its quantile function:

$$Q(u) = A + B\left(1 + c\frac{1 - \exp(-gz(u))}{1 + \exp(-gz(u))}\right)(1 + z(u)^2)^k z(u)$$

where $z(u) = \Phi^{-1}(u)$, the standard normal quantile of u.

Cumulative distribution function (c.d.f):

The cumulative distribution function is typically evaluated numerically due to the lack of a closed-form expression.

See Also

gk::dgk, dist_gh

Examples

```
dist <- dist_gk(A = 0, B = 1, g = 0, k = 0.5)
dist


mean(dist)
variance(dist)
support(dist)
generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

`dist_gpd` *The Generalized Pareto Distribution*

Description

The GPD distribution is commonly used to model the tails of distributions, particularly in extreme value theory.

The Pickands–Balkema–De Haan theorem states that for a large class of distributions, the tail (above some threshold) can be approximated by a GPD.

Usage

```
dist_gpd(location, scale, shape)
```

Arguments

`location`	the location parameter a of the GPD distribution.
`scale`	the scale parameter b of the GPD distribution.
`shape`	the shape parameter s of the GPD distribution.

Details

We recommend reading this documentation on pkgdown which renders math nicely. `https://pkg.mitchelloharawild.com/distributional/reference/dist_gpd.html`

In the following, let X be a Generalized Pareto random variable with parameters `location` = a, `scale` = $b > 0$, and `shape` = s.

Support: $x \ge a$ if $s \ge 0$, $a \le x \le a - b/s$ if $s < 0$

Mean:

$$E(X) = a + \frac{b}{1-s} \quad \text{for } s < 1$$

$E(X) = \infty$ for $s \ge 1$

Variance:

$$\text{Var}(X) = \frac{b^2}{(1-s)^2(1-2s)} \quad \text{for } s < 0.5$$

$\text{Var}(X) = \infty$ for $s \ge 0.5$

Probability density function (p.d.f):

For $s = 0$:

$$f(x) = \frac{1}{b} \exp\left(-\frac{x-a}{b}\right) \quad \text{for } x \ge a$$

For $s \neq 0$:

$$f(x) = \frac{1}{b} \left(1 + s\frac{x-a}{b}\right)^{-1/s-1}$$

where $1 + s(x-a)/b > 0$

Cumulative distribution function (c.d.f):

For $s = 0$:

$$F(x) = 1 - \exp\left(-\frac{x-a}{b}\right) \quad \text{for } x \geq a$$

For $s \neq 0$:

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

where $1 + s(x-a)/b > 0$

Quantile function:

For $s = 0$:

$$Q(p) = a - b\log(1-p)$$

For $s \neq 0$:

$$Q(p) = a + \frac{b}{s}\left[(1-p)^{-s} - 1\right]$$

Median:

For $s = 0$:

$$\text{Median}(X) = a + b\log(2)$$

For $s \neq 0$:

$$\text{Median}(X) = a + \frac{b}{s}\left(2^s - 1\right)$$

Skewness and Kurtosis: No closed-form expressions; approximated numerically.

See Also

`evd::dgpd()`

Examples

```
dist <- dist_gpd(location = 0, scale = 1, shape = 0)

dist
mean(dist)
variance(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

`dist_gumbel` *The Gumbel distribution*

Description

[Stable]

The Gumbel distribution is a special case of the Generalized Extreme Value distribution, obtained when the GEV shape parameter ξ is equal to 0. It may be referred to as a type I extreme value distribution.

Usage

```
dist_gumbel(alpha, scale)
```

Arguments

`alpha`	location parameter.
`scale`	parameter. Must be strictly positive.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_gumbel.html

In the following, let X be a Gumbel random variable with location parameter `alpha` = α and scale parameter `scale` = σ.

Support: R, the set of all real numbers.

Mean:

$$E(X) = \alpha + \sigma\gamma$$

where γ is the Euler-Mascheroni constant, approximately equal to 0.5772157.

Variance:

$$\text{Var}(X) = \frac{\pi^2\sigma^2}{6}$$

Skewness:

$$\text{Skew}(X) = \frac{12\sqrt{6}\zeta(3)}{\pi^3} \approx 1.1395$$

where $\zeta(3)$ is Apery's constant, approximately equal to 1.2020569. Note that skewness is independent of the distribution parameters.

Kurtosis (excess):

$$\text{Kurt}(X) = \frac{12}{5} = 2.4$$

Note that excess kurtosis is independent of the distribution parameters.

Median:

$$\text{Median}(X) = \alpha - \sigma \ln(\ln 2)$$

Probability density function (p.d.f):

$$f(x) = \frac{1}{\sigma} \exp\left[-\frac{x-\alpha}{\sigma}\right] \exp\left\{-\exp\left[-\frac{x-\alpha}{\sigma}\right]\right\}$$

for x in R, the set of all real numbers.

Cumulative distribution function (c.d.f):

$$F(x) = \exp\left\{-\exp\left[-\frac{x-\alpha}{\sigma}\right]\right\}$$

for x in R, the set of all real numbers.

Quantile function (inverse c.d.f):

$$F^{-1}(p) = \alpha - \sigma \ln(-\ln p)$$

for p in (0, 1).

Moment generating function (m.g.f):

$$E(e^{tX}) = \Gamma(1 - \sigma t)e^{\alpha t}$$

for $\sigma t < 1$, where Γ is the gamma function.

See Also

actuar::Gumbel, `actuar::dgumbel()`, `actuar::pgumbel()`, `actuar::qgumbel()`, `actuar::rgumbel()`, `actuar::mgumbel()`

Examples

```
dist <- dist_gumbel(alpha = c(0.5, 1, 1.5, 3), scale = c(2, 2, 3, 4))
dist


mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)
support(dist)
generate(dist, 10)
```

```
density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

`dist_horseshoe` *The Horseshoe distribution*

Description

[Stable]

The horseshoe distribution (Carvalho et al., 2008) is a heavy-tailed continuous distribution defined as a scale mixture of normals. It is primarily used as a shrinkage prior in sparse Bayesian regression, where it concentrates mass near zero while retaining heavy tails that leave large signals unshrunk.

Usage

```
dist_horseshoe(lambda, tau)
```

Arguments

`lambda`	A positive numeric vector of local scale parameters $\lambda > 0$ (one per observation).
`tau`	A positive scalar global scale parameter $\tau > 0$.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_horseshoe.html

In the following, let X be a horseshoe random variable with local scale parameter `lambda` = $\lambda > 0$ and global scale parameter `tau` = $\tau > 0$.

Support: $x \in \mathbb{R}$, the set of all real numbers.

Mean: $E(X)$ — not available in closed form.

Variance: $\mathrm{Var}(X)$ — not available in closed form.

Probability density function (p.d.f):

The horseshoe density does not have a simple closed form but can be expressed as a scale mixture:

$$X \mid \lambda, \tau \sim \mathcal{N}(0,\, \lambda^2\tau^2)$$

where the half-Cauchy hyperprior $\lambda \sim C^+(0, 1)$ induces the characteristic horseshoe shrinkage behaviour.

References

Carvalho, C.M., Polson, N.G., and Scott, J.G. (2008). "The Horseshoe Estimator for Sparse Signals". *Discussion Paper 2008-31*. Duke University Department of Statistical Science.

Carvalho, C.M., Polson, N.G., and Scott, J.G. (2009). "Handling Sparsity via the Horseshoe". *Journal of Machine Learning Research*, 5, p. 73–80.

See Also

LaplacesDemon::dhs(), LaplacesDemon::rhs()

Examples

```
dist <- dist_horseshoe(lambda = c(0.5, 1, 2), tau = 1)
dist


support(dist)
generate(dist, 10)

density(dist, 0)
density(dist, 0, log = TRUE)
```

---

dist_hypergeometric *The Hypergeometric distribution*

Description

[Stable]

To understand the HyperGeometric distribution, consider a set of r objects, of which m are of the type I and n are of the type II. A sample with size k ($k < r$) with no replacement is randomly chosen. The number of observed type I elements observed in this sample is set to be our random variable X.

Usage

```
dist_hypergeometric(m, n, k)
```

Arguments

m	The number of type I elements available.
n	The number of type II elements available.
k	The size of the sample taken.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_hypergeometric.html

In the following, let X be a HyperGeometric random variable with success probability `p` = $p = m/(m+n)$.

Support: $x \in \{\max(0, k-n), \dots, \min(k, m)\}$

Mean: $\frac{km}{m+n} = kp$

Variance: $\frac{kmn(m+n-k)}{(m+n)^2(m+n-1)} = kp(1-p)\left(1 - \frac{k-1}{m+n-1}\right)$

Probability mass function (p.m.f):

$$P(X = x) = \frac{\binom{m}{x}\binom{n}{k-x}}{\binom{m+n}{k}}$$

Cumulative distribution function (c.d.f):

$$P(X \le x) = \sum_{i=\max(0,k-n)}^{\lfloor x \rfloor} \frac{\binom{m}{i}\binom{n}{k-i}}{\binom{m+n}{k}}$$

Moment generating function (m.g.f):

$$E(e^{tX}) = \frac{\binom{m}{k}}{\binom{m+n}{k}} {}_2F_1(-m, -k; m+n-k+1; e^t)$$

where ${}_2F_1$ is the hypergeometric function.

Skewness:

$$\frac{(m+n-2k)(m+n-1)^{1/2}(m+n-2n)}{[kmn(m+n-k)]^{1/2}(m+n-2)}$$

See Also

stats::Hypergeometric

Examples

```
dist <- dist_hypergeometric(m = rep(500, 3), n = c(50, 60, 70), k = c(100, 200, 300))

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
```

```
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

`dist_inflated` *Inflate a value of a probability distribution*

Description

[Stable]

Inflated distributions add extra probability mass at a specific value, most commonly zero (zero-inflation). These distributions are useful for modeling data with excess observations at a particular value compared to what the base distribution would predict. Common applications include zero-inflated Poisson or negative binomial models for count data with many zeros.

Usage

```
dist_inflated(dist, prob, x = 0)
```

Arguments

`dist`	The distribution(s) to inflate.
`prob`	The added probability of observing `x`.
`x`	The value to inflate. The default of `x = 0` is for zero-inflation.

Details

We recommend reading this documentation on pkgdown which renders math nicely. `https://pkg.mitchelloharawild.com/distributional/reference/dist_inflated.html`

In the following, let Y be an inflated random variable based on a base distribution X, with inflation value `x` = c and inflation probability `prob` = p.

Support: Same as the base distribution, but with additional probability mass at c

Mean: (when `x` is numeric)

$$E(Y) = p \cdot c + (1 - p) \cdot E(X)$$

Variance: (when `x = 0`)

$$\text{Var}(Y) = (1 - p) \cdot \text{Var}(X) + p(1 - p) \cdot [E(X)]^2$$

For non-zero inflation values, the variance is not computed in closed form.

Probability mass/density function (p.m.f/p.d.f):

For discrete distributions:

$$f_Y(y) = \begin{cases} p + (1-p) \cdot f_X(c) & \text{if } y = c \\ (1-p) \cdot f_X(y) & \text{if } y \neq c \end{cases}$$

For continuous distributions:

$$f_Y(y) = \begin{cases} p & \text{if } y = c \\ (1-p) \cdot f_X(y) & \text{if } y \neq c \end{cases}$$

Cumulative distribution function (c.d.f):

$$F_Y(q) = \begin{cases} (1-p) \cdot F_X(q) & \text{if } q < c \\ p + (1-p) \cdot F_X(q) & \text{if } q \geq c \end{cases}$$

Quantile function:

The quantile function is computed numerically by inverting the inflated CDF, accounting for the jump in probability at the inflation point.

Examples

```
# Zero-inflated Poisson
dist <- dist_inflated(dist_poisson(lambda = 2), prob = 0.3, x = 0)

dist
mean(dist)
variance(dist)

generate(dist, 10)

density(dist, 0)
density(dist, 1)

cdf(dist, 2)

quantile(dist, 0.5)
```

`dist_inverse_exponential` *The Inverse Exponential distribution*

Description

[Stable]

The Inverse Exponential distribution is used to model the reciprocal of exponentially distributed variables.

Usage

```
dist_inverse_exponential(rate)
```

Arguments

`rate`	an alternative way to specify the scale.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_inverse_exponential.html

In the following, let X be an Inverse Exponential random variable with parameter `rate` = λ.

Support: $x > 0$

Mean: Does not exist, returns NA

Variance: Does not exist, returns NA

Probability density function (p.d.f):

$$f(x) = \frac{\lambda}{x^2} e^{-\lambda/x}$$

Cumulative distribution function (c.d.f):

$$F(x) = e^{-\lambda/x}$$

Quantile function (inverse c.d.f):

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

Moment generating function (m.g.f):

Does not exist (divergent integral).

See Also

actuar::InverseExponential

Examples

```
dist <- dist_inverse_exponential(rate = 1:5)
dist


mean(dist)
variance(dist)
support(dist)
generate(dist, 10)

density(dist, 2)
```

```
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

`dist_inverse_gamma` *The Inverse Gamma distribution*

Description

[Stable]

The Inverse Gamma distribution is commonly used as a prior distribution in Bayesian statistics, particularly for variance parameters.

Usage

```
dist_inverse_gamma(shape, rate = 1/scale, scale)
```

Arguments

shape, scale	parameters. Must be strictly positive.
rate	an alternative way to specify the scale.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_inverse_gamma.html

In the following, let X be an Inverse Gamma random variable with shape parameter shape = α and rate parameter rate = β (equivalently, scale = $1/\beta$).

Support: $x \in (0, \infty)$

Mean: $\frac{\beta}{\alpha-1}$ for $\alpha > 1$, otherwise undefined

Variance: $\frac{\beta^2}{(\alpha-1)^2(\alpha-2)}$ for $\alpha > 2$, otherwise undefined

Probability density function (p.d.f):

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

Cumulative distribution function (c.d.f):

$$F(x) = \frac{\Gamma(\alpha, \beta/x)}{\Gamma(\alpha)} = Q(\alpha, \beta/x)$$

where $\Gamma(\alpha, z)$ is the upper incomplete gamma function and Q is the regularized incomplete gamma function.

Moment generating function (m.g.f):

$$M_X(t) = \frac{2(-\beta t)^{\alpha/2}}{\Gamma(\alpha)} K_\alpha\left(\sqrt{-4\beta t}\right)$$

for $t < 0$, where K_α is the modified Bessel function of the second kind. The MGF does not exist for $t \geq 0$.

See Also

actuar::InverseGamma

Examples

```
dist <- dist_inverse_gamma(shape = c(1,2,3,3), rate = c(1,1,1,2))
dist


mean(dist)
variance(dist)
support(dist)
generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

`dist_inverse_gaussian` *The Inverse Gaussian distribution*

Description

[Stable]

Usage

```
dist_inverse_gaussian(mean, shape)
```

Arguments

`mean, shape`	parameters. Must be strictly positive. Infinite values are supported.

Details

The inverse Gaussian distribution (also known as the Wald distribution) is commonly used to model positive-valued data, particularly in contexts involving first passage times and reliability analysis.

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_inverse_gaussian.html

In the following, let X be an Inverse Gaussian random variable with parameters `mean` = μ and `shape` = λ.

Support: $(0, \infty)$

Mean: μ

Variance: $\frac{\mu^3}{\lambda}$

Probability density function (p.d.f):

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

Cumulative distribution function (c.d.f):

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

where Φ is the standard normal c.d.f.

Moment generating function (m.g.f):

$$E(e^{tX}) = \exp\left(\frac{\lambda}{\mu}\left(1 - \sqrt{1 - \frac{2\mu^2 t}{\lambda}}\right)\right)$$

for $t < \frac{\lambda}{2\mu^2}$.

Skewness: $3\sqrt{\frac{\mu}{\lambda}}$

Excess Kurtosis: $\frac{15\mu}{\lambda}$

Quantiles: No closed-form expression, approximated numerically.

See Also

actuar::InverseGaussian

Examples

```
dist <- dist_inverse_gaussian(mean = c(1,1,1,3,3), shape = c(0.2, 1, 3, 0.2, 1))
dist


mean(dist)
variance(dist)
support(dist)
```

```
generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

`dist_laplace` *The Laplace distribution*

Description

[Stable]

The Laplace distribution, also known as the double exponential distribution, is a continuous probability distribution that is symmetric around its location parameter.

Usage

```
dist_laplace(mu, sigma)
```

Arguments

`mu`	The location parameter (mean) of the Laplace distribution.
`sigma`	The positive scale parameter of the Laplace distribution.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_laplace.html

In the following, let X be a Laplace random variable with location parameter `mu` = μ and scale parameter `sigma` = σ.

Support: R, the set of all real numbers

Mean: μ

Variance: $2\sigma^2$

Probability density function (p.d.f):

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

Cumulative distribution function (c.d.f):

$$F(x) = \begin{cases} \frac{1}{2}\exp\left(\frac{x-\mu}{\sigma}\right) & \text{if } x < \mu \\ 1 - \frac{1}{2}\exp\left(-\frac{x-\mu}{\sigma}\right) & \text{if } x \geq \mu \end{cases}$$

Moment generating function (m.g.f):

$$E(e^{tX}) = \frac{\exp(\mu t)}{1-\sigma^2 t^2} \text{ for } |t| < \frac{1}{\sigma}$$

See Also

extraDistr::Laplace

Examples

```
dist <- dist_laplace(mu = c(0, 2, -1), sigma = c(1, 2, 0.5))

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 0)
density(dist, 0, log = TRUE)

cdf(dist, 1)

quantile(dist, 0.7)
```

---

dist_logarithmic *The Logarithmic distribution*

Description

[Stable]

The Logarithmic distribution is a discrete probability distribution derived from the logarithmic series. It is useful in modeling the abundance of species and other phenomena where the frequency of an event follows a logarithmic pattern.

Usage

```
dist_logarithmic(prob)
```

Arguments

prob	parameter. `0 <= prob < 1`.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_logarithmic.html

In the following, let X be a Logarithmic random variable with parameter prob = p.

Support: $\{1, 2, 3, ...\}$

Mean: $\frac{-1}{\log(1-p)} \cdot \frac{p}{1-p}$

Variance: $\frac{-(p^2 + p\log(1-p))}{[(1-p)\log(1-p)]^2}$

Probability mass function (p.m.f):

$$P(X = k) = \frac{-1}{\log(1-p)} \cdot \frac{p^k}{k}$$

for $k = 1, 2, 3, \ldots$

Cumulative distribution function (c.d.f):

The c.d.f. does not have a simple closed form. It is computed using the recurrence relationship $P(X = k+1) = \frac{p \cdot k}{k+1} \cdot P(X = k)$ starting from $P(X = 1) = \frac{-p}{\log(1-p)}$.

Moment generating function (m.g.f):

$$E(e^{tX}) = \frac{\log(1 - pe^t)}{\log(1-p)}$$

for $pe^t < 1$

See Also

actuar::Logarithmic

Examples

```
dist <- dist_logarithmic(prob = c(0.33, 0.66, 0.99))
dist


mean(dist)
variance(dist)
support(dist)
generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

`dist_logistic` *The Logistic distribution*

Description

[Stable]

A continuous distribution on the real line. For binary outcomes the model given by $P(Y = 1|X) = F(X\beta)$ where F is the Logistic `cdf()` is called *logistic regression*.

Usage

```
dist_logistic(location, scale)
```

Arguments

`location, scale` location and scale parameters.

Details

We recommend reading this documentation on pkgdown which renders math nicely. `https://pkg.mitchelloharawild.com/distributional/reference/dist_logistic.html`

In the following, let X be a Logistic random variable with `location` = μ and `scale` = s.

Support: R, the set of all real numbers

Mean: μ

Variance: $s^2\pi^2/3$

Probability density function (p.d.f):

$$f(x) = \frac{e^{-\frac{x-\mu}{s}}}{s\left[1 + e^{-\frac{x-\mu}{s}}\right]^2}$$

Cumulative distribution function (c.d.f):

$$F(x) = \frac{1}{1 + e^{-\frac{x-\mu}{s}}}$$

Moment generating function (m.g.f):

$$E(e^{tX}) = e^{\mu t}B(1 - st, 1 + st)$$

for $-1 < st < 1$, where $B(a, b)$ is the Beta function.

See Also

stats::Logistic

Examples

```
dist <- dist_logistic(location = c(5,9,9,6,2), scale = c(2,3,4,2,1))

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

---

dist_lognormal *The log-normal distribution*

Description

[Stable]

The log-normal distribution is a commonly used transformation of the Normal distribution. If X follows a log-normal distribution, then $\ln X$ would be characterised by a Normal distribution.

Usage

```
dist_lognormal(mu = 0, sigma = 1)
```

Arguments

mu	The mean (location parameter) of the distribution, which is the mean of the associated Normal distribution. Can be any real number.
sigma	The standard deviation (scale parameter) of the distribution. Can be any positive number.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_lognormal.html

In the following, let X be a log-normal random variable with `mu` = μ and `sigma` = σ.

Support: R^+, the set of positive real numbers.

Mean: $e^{\mu+\sigma^2/2}$

Variance: $(e^{\sigma^2} - 1)e^{2\mu+\sigma^2}$

Skewness: $(e^{\sigma^2} + 2)\sqrt{e^{\sigma^2} - 1}$

Excess Kurtosis: $e^{4\sigma^2} + 2e^{3\sigma^2} + 3e^{2\sigma^2} - 6$

Probability density function (p.d.f):

$$f(x) = \frac{1}{x\sqrt{2\pi\sigma^2}}e^{-(\ln x-\mu)^2/(2\sigma^2)}$$

Cumulative distribution function (c.d.f):

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

where Φ is the c.d.f. of the standard Normal distribution.

Moment generating function (m.g.f):

Does not exist in closed form.

See Also

stats::Lognormal

Examples

```
dist <- dist_lognormal(mu = 1:5, sigma = 0.1)

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)

# A log-normal distribution X is exp(Y), where Y is a Normal distribution of
# the same parameters. So log(X) will produce the Normal distribution Y.
log(dist)
```

dist_missing *Missing distribution*

Description

[Maturing]

A placeholder distribution for handling missing values in a vector of distributions.

Usage

```
dist_missing(length = 1)
```

Arguments

`length`	The number of missing distributions

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_missing.html

The missing distribution represents the absence of distributional information. It is used as a placeholder when distribution values are not available or not applicable, similar to how `NA` is used for missing scalar values.

Support: Undefined

Mean: NA

Variance: NA

Skewness: NA

Kurtosis: NA

Probability density function (p.d.f): Undefined

$$f(x) = \text{NA}$$

Cumulative distribution function (c.d.f): Undefined

$$F(t) = \text{NA}$$

Quantile function: Undefined

$$Q(p) = \text{NA}$$

Moment generating function (m.g.f): Undefined

$$E(e^{tX}) = \text{NA}$$

All statistical operations on missing distributions return `NA` values of appropriate length, propagating the missingness through calculations.

See Also

base::NA

Examples

```
dist <- dist_missing(3L)

dist
mean(dist)
variance(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

`dist_mixture` *Create a mixture of distributions*

Description

[Maturing]

A mixture distribution combines multiple component distributions with specified weights. The resulting distribution can model complex, multimodal data by representing it as a weighted sum of simpler distributions.

Usage

```
dist_mixture(..., weights = numeric())
```

Arguments

`...`	Distributions to be used in the mixture. Can be any distributional objects.
`weights`	A numeric vector of non-negative weights that sum to 1. The length must match the number of distributions passed to `...`. Each weight w_i represents the probability that a random draw comes from the i-th component distribution.

Details

In the following, let X be a mixture random variable composed of K component distributions $F_1, F_2, \ldots, F_K$ with corresponding weights $w_1, w_2, \ldots, w_K$ where $\sum_{i=1}^K w_i = 1$ and $w_i \geq 0$ for all i.

Support: The union of the supports of all component distributions

Mean:

For univariate mixtures:

$$E(X) = \sum_{i=1}^K w_i \mu_i$$

where μ_i is the mean of the i-th component distribution.

For multivariate mixtures:

$$E(\mathbf{X}) = \sum_{i=1}^K w_i \boldsymbol{\mu}_i$$

where $\boldsymbol{\mu}_i$ is the mean vector of the i-th component distribution.

Variance:

For univariate mixtures:

$$\text{Var}(X) = \sum_{i=1}^K w_i(\mu_i^2 + \sigma_i^2) - \left(\sum_{i=1}^K w_i \mu_i\right)^2$$

where σ_i^2 is the variance of the i-th component distribution.

Covariance:

For multivariate mixtures:

$$\text{Cov}(\mathbf{X}) = \sum_{i=1}^K w_i \left[(\boldsymbol{\mu}_i - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu}_i - \bar{\boldsymbol{\mu}})^T + \boldsymbol{\Sigma}_i\right]$$

where $\bar{\boldsymbol{\mu}} = \sum_{i=1}^K w_i \boldsymbol{\mu}_i$ is the overall mean vector and $\boldsymbol{\Sigma}_i$ is the covariance matrix of the i-th component distribution.

Probability density/mass function (p.d.f/p.m.f):

$$f(x) = \sum_{i=1}^K w_i f_i(x)$$

where $f_i(x)$ is the density or mass function of the i-th component distribution.

Cumulative distribution function (c.d.f):

For univariate mixtures:

$$F(x) = \sum_{i=1}^K w_i F_i(x)$$

where $F_i(x)$ is the c.d.f. of the i-th component distribution.

For multivariate mixtures, the c.d.f. is approximated numerically.

Quantile function:

For univariate mixtures, the quantile function has no closed form and is computed numerically by inverting the c.d.f. using root-finding (`stats::uniroot()`).

For multivariate mixtures, quantiles are not yet implemented.

See Also

`stats::uniroot()`, `vctrs::vec_unique_count()`

Examples

```
# Univariate mixture of two normal distributions
dist <- dist_mixture(dist_normal(0, 1), dist_normal(5, 2), weights = c(0.3, 0.7))
dist

mean(dist)
variance(dist)

density(dist, 2)
cdf(dist, 2)
quantile(dist, 0.5)

generate(dist, 10)
```

`dist_multinomial` *The Multinomial distribution*

Description

[Stable]

The multinomial distribution is a generalization of the binomial distribution to multiple categories. It is perhaps easiest to think that we first extend a `dist_bernoulli()` distribution to include more than two categories, resulting in a `dist_categorical()` distribution. We then extend repeat the Categorical experiment several (n) times.

Usage

```
dist_multinomial(size, prob)
```

Arguments

`size`	The number of draws from the Categorical distribution.
`prob`	The probability of an event occurring from each draw.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_multinomial.html

In the following, let $X = (X_1, ..., X_k)$ be a Multinomial random variable with success probability `prob` = p. Note that p is vector with k elements that sum to one. Assume that we repeat the Categorical experiment `size` = n times.

Support: Each X_i is in $\{0, 1, 2, ..., n\}$.

Mean: The mean of X_i is np_i.

Variance: The variance of X_i is $np_i(1 - p_i)$. For $i \neq j$, the covariance of X_i and X_j is $-np_ip_j$.

Probability mass function (p.m.f):

$$P(X_1 = x_1, ..., X_k = x_k) = \frac{n!}{x_1!x_2! \cdots x_k!} p_1^{x_1} \cdot p_2^{x_2} \cdot \ldots \cdot p_k^{x_k}$$

where $\sum_{i=1}^k x_i = n$ and $\sum_{i=1}^k p_i = 1$.

Cumulative distribution function (c.d.f):

$$P(X_1 \leq q_1, ..., X_k \leq q_k) = \sum_{\substack{x_1, \ldots, x_k \geq 0 \\ x_i \leq q_i \text{ for all } i \\ \sum_{i=1}^k x_i = n}} \frac{n!}{x_1!x_2! \cdots x_k!} p_1^{x_1} \cdot p_2^{x_2} \cdot \ldots \cdot p_k^{x_k}$$

The c.d.f. is computed as a finite sum of the p.m.f. over all integer vectors in the support that satisfy the componentwise inequalities.

Moment generating function (m.g.f):

$$E(e^{t'X}) = \left(\sum_{i=1}^k p_i e^{t_i} \right)^n$$

where $t = (t_1, ..., t_k)$ is a vector of the same dimension as X.

Skewness: The skewness of X_i is

$$\frac{1 - 2p_i}{\sqrt{np_i(1 - p_i)}}$$

Excess Kurtosis: The excess kurtosis of X_i is

$$\frac{1 - 6p_i(1 - p_i)}{np_i(1 - p_i)}$$

See Also

`stats::dmultinom()`, `stats::rmultinom()`

Examples

```
dist <- dist_multinomial(size = c(4, 3), prob = list(c(0.3, 0.5, 0.2), c(0.1, 0.5, 0.4)))

dist
mean(dist)
variance(dist)

generate(dist, 10)

density(dist, list(d = rbind(cbind(1,2,1), cbind(0,2,1))))
density(dist, list(d = rbind(cbind(1,2,1), cbind(0,2,1))), log = TRUE)

cdf(dist, cbind(1,2,1))
```

---

dist_multivariate_normal

The multivariate normal distribution

Description

[Stable]

The multivariate normal distribution is a generalization of the univariate normal distribution to higher dimensions. It is widely used in multivariate statistics and describes the joint distribution of multiple correlated continuous random variables.

Usage

```
dist_multivariate_normal(mu = 0, sigma = list(diag(1)))
```

Arguments

`mu`	A list of numeric vectors for the distribution's mean.
`sigma`	A list of matrices for the distribution's variance-covariance matrix.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_multivariate_normal.html

In the following, let $\mathbf{X}$ be a k-dimensional multivariate normal random variable with mean vector `mu` = $\boldsymbol{\mu}$ and variance-covariance matrix `sigma` = $\boldsymbol{\Sigma}$.

Support: $\mathbf{x} \in \mathbb{R}^k$

Mean: $\boldsymbol{\mu}$

Variance-covariance matrix: $\boldsymbol{\Sigma}$

Probability density function (p.d.f):

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{k/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

where $|\mathbf{\Sigma}|$ is the determinant of $\mathbf{\Sigma}$.

Cumulative distribution function (c.d.f):

$$P(\mathbf{X} \le \mathbf{q}) = P(X_1 \le q_1, \ldots, X_k \le q_k)$$

The c.d.f. does not have a closed-form expression and is computed numerically.

Moment generating function (m.g.f):

$$M(\mathbf{t}) = E(e^{\mathbf{t}^T\mathbf{X}}) = \exp\left(\mathbf{t}^T\boldsymbol{\mu} + \frac{1}{2}\mathbf{t}^T\mathbf{\Sigma}\mathbf{t}\right)$$

See Also

mvtnorm::dmvnorm(), mvtnorm::pmvnorm(), mvtnorm::qmvnorm(), mvtnorm::rmvnorm()

Examples

```
dist <- dist_multivariate_normal(mu = list(c(1,2)), sigma = list(matrix(c(4,2,2,3), ncol=2)))
dimnames(dist) <- c("x", "y")
dist


mean(dist)
variance(dist)
support(dist)
generate(dist, 10)

density(dist, cbind(2, 1))
density(dist, cbind(2, 1), log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7, kind = "equicoordinate")
quantile(dist, 0.7, kind = "marginal")
```

---

dist_multivariate_t *The multivariate t-distribution*

Description

[Stable]

The multivariate t-distribution is a generalization of the univariate Student's t-distribution to multiple dimensions. It is commonly used for modeling heavy-tailed multivariate data and in robust statistics.

Usage

```
dist_multivariate_t(df = 1, mu = 0, sigma = diag(1))
```

Arguments

`df`	A numeric vector of degrees of freedom (must be positive).
`mu`	A list of numeric vectors for the distribution location parameter.
`sigma`	A list of matrices for the distribution scale matrix.

Details

We recommend reading this documentation on pkgdown which renders math nicely. `https://pkg.mitchelloharawild.com/distributional/reference/dist_multivariate_t.html`

In the following, let $\mathbf{X}$ be a multivariate t random vector with degrees of freedom `df` = ν, location parameter `mu` = $\boldsymbol{\mu}$, and scale matrix `sigma` = $\boldsymbol{\Sigma}$.

Support: $\mathbf{x} \in \mathbb{R}^k$, where k is the dimension of the distribution

Mean: $\boldsymbol{\mu}$ for $\nu > 1$, undefined otherwise

Covariance matrix:

$$\mathrm{Cov}(\mathbf{X}) = \frac{\nu}{\nu - 2}\boldsymbol{\Sigma}$$

for $\nu > 2$, undefined otherwise

Probability density function (p.d.f):

$$f(\mathbf{x}) = \frac{\Gamma\left(\frac{\nu+k}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\nu^{k/2}\pi^{k/2}|\boldsymbol{\Sigma}|^{1/2}}\left[1 + \frac{1}{\nu}(\mathbf{x} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right]^{-\frac{\nu+k}{2}}$$

where k is the dimension of the distribution and $\Gamma(\cdot)$ is the gamma function.

Cumulative distribution function (c.d.f):

$$F(\mathbf{t}) = \int_{-\infty}^{t_1} \cdots \int_{-\infty}^{t_k} f(\mathbf{x})\, d\mathbf{x}$$

This integral does not have a closed form solution and is approximated numerically.

Quantile function:

The equicoordinate quantile function finds q such that:

$$P(X_1 \le q, \ldots, X_k \le q) = p$$

This does not have a closed form solution and is approximated numerically.

The marginal quantile function for each dimension i is:

$$Q_i(p) = \mu_i + \sqrt{\Sigma_{ii}} \cdot t_\nu^{-1}(p)$$

where $t_\nu^{-1}(p)$ is the quantile function of the univariate Student's t-distribution with ν degrees of freedom, and Σ_{ii} is the i-th diagonal element of `sigma`.

See Also

mvtnorm::dmvt, mvtnorm::pmvt, mvtnorm::qmvt, mvtnorm::rmvt

Examples

```
dist <- dist_multivariate_t(
  df = 5,
  mu = list(c(1, 2)),
  sigma = list(matrix(c(4, 2, 2, 3), ncol = 2))
)
dimnames(dist) <- c("x", "y")
dist


mean(dist)
variance(dist)
support(dist)
generate(dist, 10)

density(dist, cbind(2, 1))
density(dist, cbind(2, 1), log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
quantile(dist, 0.7, kind = "marginal")
```

---

dist_negative_binomial

The Negative Binomial distribution

Description

[Stable]

A generalization of the geometric distribution. It is the number of failures in a sequence of i.i.d. Bernoulli trials before a specified number of successes (`size`) occur. The probability of success in each trial is given by `prob`.

Usage

```
dist_negative_binomial(size, prob)
```

Arguments

`size`	The number of successful trials (target number of successes). Must be a positive number. Also called the dispersion parameter.
`prob`	The probability of success in each trial. Must be between 0 and 1.

Details

We recommend reading this documentation on pkgdown which renders math nicely. `https://pkg.mitchelloharawild.com/distributional/reference/dist_negative_binomial.html`

In the following, let X be a Negative Binomial random variable with success probability `prob` = p and the number of successes `size` = r.

Support: $\{0, 1, 2, 3, ...\}$

Mean: $\frac{r(1-p)}{p}$

Variance: $\frac{r(1-p)}{p^2}$

Probability mass function (p.m.f):

$$P(X = k) = \binom{k + r - 1}{k}(1 - p)^r p^k$$

Cumulative distribution function (c.d.f):

$$F(k) = \sum_{i=0}^{\lfloor k \rfloor} \binom{i + r - 1}{i}(1 - p)^r p^i$$

This can also be expressed in terms of the regularized incomplete beta function, and is computed numerically.

Moment generating function (m.g.f):

$$E(e^{tX}) = \left(\frac{1 - p}{1 - pe^t}\right)^r, \quad t < -\log p$$

Skewness:

$$\gamma_1 = \frac{2 - p}{\sqrt{r(1 - p)}}$$

Excess Kurtosis:

$$\gamma_2 = \frac{6}{r} + \frac{p^2}{r(1 - p)}$$

See Also

stats::NegBinomial

Examples

```
dist <- dist_negative_binomial(size = 10, prob = 0.5)

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)
support(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

---

dist_normal *The Normal distribution*

Description

[Stable]

The Normal distribution is ubiquitous in statistics, partially because of the central limit theorem, which states that sums of i.i.d. random variables eventually become Normal. Linear transformations of Normal random variables result in new random variables that are also Normal. If you are taking an intro stats course, you'll likely use the Normal distribution for Z-tests and in simple linear regression. Under regularity conditions, maximum likelihood estimators are asymptotically Normal. The Normal distribution is also called the gaussian distribution.

Usage

```
dist_normal(mu = 0, sigma = 1, mean = mu, sd = sigma)
```

Arguments

`mu, mean`	The mean (location parameter) of the distribution, which is also the mean of the distribution. Can be any real number.
`sigma, sd`	The standard deviation (scale parameter) of the distribution. Can be any positive number. If you would like a Normal distribution with **variance** σ^2, be sure to take the square root, as this is a common source of errors.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_normal.html

In the following, let X be a Normal random variable with mean `mu` = μ and standard deviation `sigma` = σ.

Support: R, the set of all real numbers

Mean: μ

Variance: σ^2

Probability density function (p.d.f):

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}$$

Cumulative distribution function (c.d.f):

$$F(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} dx$$

This integral does not have a closed form solution and is approximated numerically. The c.d.f. of a standard Normal is sometimes called the "error function". The notation $\Phi(t)$ also stands for the c.d.f. of a standard Normal evaluated at t. Z-tables list the value of $\Phi(t)$ for various t.

Moment generating function (m.g.f):

$$E(e^{tX}) = e^{\mu t + \sigma^2 t^2/2}$$

See Also

stats::Normal

Examples

```
dist <- dist_normal(mu = 1:5, sigma = 3)

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

`dist_pareto` *The Pareto Distribution*

Description

[Stable]

The Pareto distribution is a power-law probability distribution commonly used in actuarial science to model loss severity and in economics to model income distributions and firm sizes.

Usage

```
dist_pareto(shape, scale)
```

Arguments

`shape, scale`	parameters. Must be strictly positive.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_pareto.html

In the following, let X be a Pareto random variable with parameters `shape` = α and `scale` = θ.

Support: $(0, \infty)$

Mean: $\frac{\theta}{\alpha-1}$ for $\alpha > 1$, undefined otherwise

Variance: $\frac{\alpha\theta^2}{(\alpha-1)^2(\alpha-2)}$ for $\alpha > 2$, undefined otherwise

Probability density function (p.d.f):

$$f(x) = \frac{\alpha\theta^\alpha}{(x+\theta)^{\alpha+1}}$$

for $x > 0$, $\alpha > 0$ and $\theta > 0$.

Cumulative distribution function (c.d.f):

$$F(x) = 1 - \left(\frac{\theta}{x+\theta}\right)^\alpha$$

for $x > 0$.

Moment generating function (m.g.f):

Does not exist in closed form, but the kth raw moment $E[X^k]$ exists for $-1 < k < \alpha$.

Note

There are many different definitions of the Pareto distribution in the literature; see Arnold (2015) or Kleiber and Kotz (2003). This implementation uses the Pareto distribution without a location parameter as described in actuar::Pareto.

References

Kleiber, C. and Kotz, S. (2003), *Statistical Size Distributions in Economics and Actuarial Sciences*, Wiley.

Klugman, S. A., Panjer, H. H. and Willmot, G. E. (2012), *Loss Models, From Data to Decisions, Fourth Edition*, Wiley.

See Also

actuar::Pareto

Examples

```
dist <- dist_pareto(shape = c(10, 3, 2, 1), scale = rep(1, 4))
dist


mean(dist)
variance(dist)
support(dist)
generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

`dist_percentile` *Percentile distribution*

Description

[Stable]

The Percentile distribution is a non-parametric distribution defined by a set of quantiles at specified percentile values. This distribution is useful for representing empirical distributions or elicited expert knowledge when only percentile information is available. The distribution uses linear interpolation between percentiles and can be used to approximate complex distributions that may not have simple parametric forms.

Usage

```
dist_percentile(x, percentile)
```

Arguments

`x`	A list of values
`percentile`	A list of percentiles

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_percentile.html

In the following, let X be a Percentile random variable defined by values $x_1, x_2, \ldots, x_n$ at percentiles $p_1, p_2, \ldots, p_n$ where $0 \le p_i \le 100$.

Support: $[\min(x_i), \max(x_i)]$ if $\min(p_i) > 0$ or $\max(p_i) < 100$, otherwise support is approximated from the specified percentiles.

Mean: Approximated numerically using spline interpolation and numerical integration:

$$E(X) \approx \int_0^1 Q(u)du$$

where $Q(u)$ is a spline function interpolating the percentile values.

Variance: Approximated numerically.

Probability density function (p.d.f): Approximated numerically using kernel density estimation from generated samples.

Cumulative distribution function (c.d.f): Defined by linear interpolation:

$$F(t) = \begin{cases} p_1/100 & \text{if } t < x_1 \\ p_i/100 + \frac{(t-x_i)(p_{i+1}-p_i)}{100(x_{i+1}-x_i)} & \text{if } x_i \le t < x_{i+1} \\ p_n/100 & \text{if } t \ge x_n \end{cases}$$

Quantile function: Defined by linear interpolation:

$$Q(u) = x_i + \frac{(100u - p_i)(x_{i+1} - x_i)}{p_{i+1} - p_i}$$

for $p_i/100 \le u \le p_{i+1}/100$.

Examples

```
dist <- dist_normal()
percentiles <- seq(0.01, 0.99, by = 0.01)
x <- vapply(percentiles, quantile, double(1L), x = dist)
dist_percentile(list(x), list(percentiles*100))
```

dist_poisson *The Poisson Distribution*

Description

[Stable]

Poisson distributions are frequently used to model counts. The Poisson distribution is commonly used to model the number of events occurring in a fixed interval of time or space when these events occur with a known constant mean rate and independently of the time since the last event. Examples include the number of emails received per hour, the number of decay events per second from a radioactive source, or the number of customers arriving at a store per day.

Usage

```
dist_poisson(lambda)
```

Arguments

`lambda`	The rate parameter (mean and variance) of the distribution. Can be any positive number. This represents the expected number of events in the given interval.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_poisson.html

In the following, let X be a Poisson random variable with parameter `lambda` = λ.

Support: $\{0, 1, 2, 3, ...\}$

Mean: λ

Variance: λ

Probability mass function (p.m.f):

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Cumulative distribution function (c.d.f):

$$P(X \le k) = e^{-\lambda} \sum_{i=0}^{\lfloor k \rfloor} \frac{\lambda^i}{i!}$$

Moment generating function (m.g.f):

$$E(e^{tX}) = e^{\lambda(e^t - 1)}$$

Skewness:

$$\gamma_1 = \frac{1}{\sqrt{\lambda}}$$

Excess kurtosis:

$$\gamma_2 = \frac{1}{\lambda}$$

See Also

stats::Poisson

Examples

```
dist <- dist_poisson(lambda = c(1, 4, 10))

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

---

dist_poisson_inverse_gaussian

The Poisson-Inverse Gaussian distribution

Description

[Stable]

The Poisson-Inverse Gaussian distribution is a compound Poisson distribution where the rate parameter follows an Inverse Gaussian distribution. It is useful for modeling overdispersed count data.

Usage

```
dist_poisson_inverse_gaussian(mean, shape)
```

Arguments

`mean, shape` parameters. Must be strictly positive. Infinite values are supported.

Details

We recommend reading this documentation on pkgdown which renders math nicely. `https://pkg.mitchelloharawild.com/distributional/reference/dist_poisson_inverse_gaussian.html`

In the following, let X be a Poisson-Inverse Gaussian random variable with parameters `mean` = μ and `shape` = ϕ.

Support: $\{0, 1, 2, 3, ...\}$

Mean: μ

Variance: $\frac{\mu}{\phi}(\mu^2 + \phi)$

Probability mass function (p.m.f):

$$P(X = x) = \frac{e^{\phi}}{\sqrt{2\pi}} \left(\frac{\phi}{\mu^2}\right)^{x/2} \frac{1}{x!} \int_0^{\infty} u^{x-1/2} \exp\left(-\frac{\phi u}{2} - \frac{\phi}{2\mu^2 u}\right) du$$

for $x = 0, 1, 2, \ldots$

Cumulative distribution function (c.d.f):

$$P(X \le x) = \sum_{k=0}^{\lfloor x \rfloor} P(X = k)$$

The c.d.f does not have a closed form and is approximated numerically.

Moment generating function (m.g.f):

$$E(e^{tX}) = \exp\left\{\phi\left[1 - \sqrt{1 - \frac{2\mu^2}{\phi}(e^t - 1)}\right]\right\}$$

for $t < -\log(1 + \phi/(2\mu^2))$

See Also

actuar::PoissonInverseGaussian, `actuar::dpoisinvgauss()`, `actuar::ppoisinvgauss()`, `actuar::qpoisinvgauss()`, `actuar::rpoisinvgauss()`

Examples

```
dist <- dist_poisson_inverse_gaussian(mean = rep(0.1, 3), shape = c(0.4, 0.8, 1))
dist


mean(dist)
variance(dist)
support(dist)
generate(dist, 10)
```

```
density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

dist_sample *Sampling distribution*

Description

[Stable]

The sampling distribution represents an empirical distribution based on observed samples. It is useful for bootstrapping, representing posterior distributions from Markov Chain Monte Carlo (MCMC) algorithms, or working with any empirical data where the parametric form is unknown. Unlike parametric distributions, the sampling distribution makes no assumptions about the underlying data-generating process and instead uses the sample itself to estimate distributional properties. The distribution can handle both univariate and multivariate samples.

Usage

```
dist_sample(x)
```

Arguments

x	A list of sampled values. For univariate distributions, each element should be a numeric vector. For multivariate distributions, each element should be a matrix where columns represent variables and rows represent observations.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_sample.html

In the following, let X be a random variable with sample $x_1, x_2, \dots, x_n$ of size n.

Support: The observed range of the sample

Mean (univariate):

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Mean (multivariate): Computed independently for each variable.

Variance (univariate):

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Covariance (multivariate): The sample covariance matrix.

Skewness (univariate):

$$g_1 = \frac{\sqrt{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{\left(\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^{3/2}}\left(1 - \frac{1}{n}\right)^{3/2}$$

Probability density function: Approximated numerically using kernel density estimation.

Cumulative distribution function (univariate):

$$F(q) = \frac{1}{n}\sum_{i=1}^{n} I(x_i \le q)$$

where $I(\cdot)$ is the indicator function.

Cumulative distribution function (multivariate):

$$F(\mathbf{q}) = \frac{1}{n}\sum_{i=1}^{n} I(\mathbf{x}_i \le \mathbf{q})$$

where the inequality is applied element-wise.

Quantile function (univariate): The sample quantile, computed using the specified quantile type (see `stats::quantile()`).

Quantile function (multivariate): Marginal quantiles are computed independently for each variable.

Random generation: Bootstrap sampling with replacement from the empirical sample.

See Also

`stats::density()`, `stats::quantile()`, `stats::cov()`

Examples

```
# Univariate numeric samples
dist <- dist_sample(x = list(rnorm(100), rnorm(100, 10)))

dist
mean(dist)
variance(dist)
skewness(dist)
generate(dist, 10)

density(dist, 1)

# Multivariate numeric samples
```

```
dist <- dist_sample(x = list(cbind(rnorm(100), rnorm(100, 10))))
dimnames(dist) <- c("x", "y")

dist
mean(dist)
variance(dist)
generate(dist, 10)
quantile(dist, 0.4) # Returns the marginal quantiles
cdf(dist, matrix(c(0.3,9), nrow = 1))
```

---

dist_studentized_range

The Studentized Range distribution

Description

[Stable]

Tukey's studentized range distribution, used for Tukey's honestly significant differences test in ANOVA.

Usage

```
dist_studentized_range(nmeans, df, nranges)
```

Arguments

nmeans	sample size for range (same for each group).
df	degrees of freedom for s (see below).
nranges	number of *groups* whose **maximum** range is considered.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_studentized_range.html

In the following, let Q be a Studentized Range random variable with parameters nmeans = k (number of groups), df = ν (degrees of freedom), and nranges = n (number of ranges).

Support: R^+, the set of positive real numbers.

Mean: Approximated numerically.

Variance: Approximated numerically.

Probability density function (p.d.f): The density does not have a closed-form expression and is computed numerically.

Cumulative distribution function (c.d.f): The c.d.f does not have a simple closed-form expression. For $n = 1$ (single range), it involves integration over the joint distribution of the sample range and an independent chi-square variable. The general form is computed numerically using algorithms described in the references for stats::ptukey().

Moment generating function (m.g.f): Does not exist in closed form.

See Also

stats::Tukey

Examples

```
dist <- dist_studentized_range(nmeans = c(6, 2), df = c(5, 4), nranges = c(1, 1))

dist

cdf(dist, 4)

quantile(dist, 0.7)
```

---

dist_student_t — *The (non-central) location-scale Student t Distribution*

Description

[Stable]

The Student's T distribution is closely related to the `Normal()` distribution, but has heavier tails. As ν increases to ∞, the Student's T converges to a Normal. The T distribution appears repeatedly throughout classic frequentist hypothesis testing when comparing group means.

Usage

```
dist_student_t(df, mu = 0, sigma = 1, ncp = NULL)
```

Arguments

`df`	degrees of freedom (> 0, maybe non-integer). `df = Inf` is allowed.
`mu`	The location parameter of the distribution. If `ncp == 0` (or `NULL`), this is the median.
`sigma`	The scale parameter of the distribution.
`ncp`	non-centrality parameter δ; currently except for `rt()`, accurate only for `abs(ncp) <= 37.62`. If omitted, use the central t distribution.

Details

We recommend reading this documentation on pkgdown which renders math nicely. `https://pkg.mitchelloharawild.com/distributional/reference/dist_student_t.html`

In the following, let X be a location-scale Student's T random variable with `df` = ν, `mu` = μ, `sigma` = σ, and `ncp` = δ (non-centrality parameter).

If Z follows a standard Student's T distribution (with `df` = ν and `ncp` = δ), then $X = \mu + \sigma Z$.

Support: R, the set of all real numbers

Mean:

For the central distribution (`ncp` = 0 or `NULL`):

$$E(X) = \mu$$

for $\nu > 1$, and undefined otherwise.

For the non-central distribution (`ncp` $\neq 0$):

$$E(X) = \mu + \delta\sqrt{\frac{\nu}{2}}\frac{\Gamma((\nu-1)/2)}{\Gamma(\nu/2)}\sigma$$

for $\nu > 1$, and undefined otherwise.

Variance:

For the central distribution (`ncp` = 0 or `NULL`):

$$\mathrm{Var}(X) = \frac{\nu}{\nu-2}\sigma^2$$

for $\nu > 2$. Undefined if $\nu \leq 1$, infinite when $1 < \nu \leq 2$.

For the non-central distribution (`ncp` $\neq 0$):

$$\mathrm{Var}(X) = \left[\frac{\nu(1+\delta^2)}{\nu-2} - \left(\delta\sqrt{\frac{\nu}{2}}\frac{\Gamma((\nu-1)/2)}{\Gamma(\nu/2)}\right)^2\right]\sigma^2$$

for $\nu > 2$. Undefined if $\nu \leq 1$, infinite when $1 < \nu \leq 2$.

Probability density function (p.d.f):

For the central distribution (`ncp` = 0 or `NULL`), the standard t distribution with `df` = ν has density:

$$f_Z(z) = \frac{\Gamma((\nu+1)/2)}{\sqrt{\pi\nu}\Gamma(\nu/2)}\left(1+\frac{z^2}{\nu}\right)^{-(\nu+1)/2}$$

The location-scale version with `mu` = μ and `sigma` = σ has density:

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

For the non-central distribution (`ncp` $\neq 0$), the density is computed numerically via `stats::dt()`.

Cumulative distribution function (c.d.f):

For the central distribution (`ncp` = 0 or `NULL`), the cumulative distribution function is computed numerically via `stats::pt()`, which uses the relationship to the incomplete beta function:

$$F_\nu(t) = \frac{1}{2}I_x\left(\frac{\nu}{2}, \frac{1}{2}\right)$$

for $t \leq 0$, where $x = \nu/(\nu + t^2)$ and $I_x(a,b)$ is the incomplete beta function (`stats::pbeta()`). For $t \geq 0$:

$$F_\nu(t) = 1 - \frac{1}{2} I_x\left(\frac{\nu}{2}, \frac{1}{2}\right)$$

The location-scale version is: $F(x) = F_\nu((x - \mu)/\sigma)$.

For the non-central distribution (`ncp` $\neq 0$), the cumulative distribution function is computed numerically via `stats::pt()`.

Moment generating function (m.g.f):

Does not exist in closed form. Moments are computed using the formulas for mean and variance above where available.

See Also

stats::TDist

Examples

```
dist <- dist_student_t(df = c(1,2,5), mu = c(0,1,2), sigma = c(1,2,3))

dist
mean(dist)
variance(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

`dist_transformed` *Modify a distribution with a transformation*

Description

[Maturing]

A transformed distribution applies a monotonic transformation to an existing distribution. This is useful for creating derived distributions such as log-normal (exponential transformation of normal), or other custom transformations of base distributions.

The `density()`, `mean()`, and `variance()` methods are approximate as they are based on numerical derivatives.

Usage

```
dist_transformed(dist, transform, inverse)
```

Arguments

`dist`	A univariate distribution vector.
`transform`	A function used to transform the distribution. This transformation should be monotonic over appropriate domain.
`inverse`	The inverse of the `transform` function.

Details

We recommend reading this documentation on pkgdown which renders math nicely. `https://pkg.mitchelloharawild.com/distributional/reference/dist_transformed.html`

Let $Y = g(X)$ where X is the base distribution with transformation function `transform` = g and `inverse` = g^{-1}. The transformation g must be monotonic over the support of X.

Support: $g(S_X)$ where S_X is the support of X

Mean: Approximated numerically using a second-order Taylor expansion:

$$E(Y) \approx g(\mu_X) + \frac{1}{2}g''(\mu_X)\sigma_X^2$$

where μ_X and σ_X^2 are the mean and variance of the base distribution X, and g'' is the second derivative of the transformation. The derivative is computed numerically using `numDeriv::hessian()`.

Variance: Approximated numerically using the delta method:

$$\mathrm{Var}(Y) \approx [g'(\mu_X)]^2\sigma_X^2 + \frac{1}{2}[g''(\mu_X)\sigma_X^2]^2$$

where g' is the first derivative (Jacobian) computed numerically using `numDeriv::jacobian()`.

Probability density function (p.d.f): Using the change of variables formula:

$$f_Y(y) = f_X(g^{-1}(y))\left|\frac{d}{dy}g^{-1}(y)\right|$$

where f_X is the p.d.f. of the base distribution and the Jacobian $|d/dy\, g^{-1}(y)|$ is computed numerically using `numDeriv::jacobian()`.

Cumulative distribution function (c.d.f):

For monotonically increasing g:

$$F_Y(y) = F_X(g^{-1}(y))$$

For monotonically decreasing g:

$$F_Y(y) = 1 - F_X(g^{-1}(y))$$

where F_X is the c.d.f. of the base distribution.

Quantile function: The inverse of the c.d.f.

For monotonically increasing g:

$$Q_Y(p) = g(Q_X(p))$$

For monotonically decreasing g:

$$Q_Y(p) = g(Q_X(1-p))$$

where Q_X is the quantile function of the base distribution.

See Also

numDeriv::jacobian(), numDeriv::hessian()

Examples

```
# Create a log normal distribution
dist <- dist_transformed(dist_normal(0, 0.5), exp, log)
density(dist, 1) # dlnorm(1, 0, 0.5)
cdf(dist, 4) # plnorm(4, 0, 0.5)
quantile(dist, 0.1) # qlnorm(0.1, 0, 0.5)
generate(dist, 10) # rlnorm(10, 0, 0.5)
```

`dist_truncated` *Truncate a distribution*

Description

[Stable]

Note that the samples are generated using inverse transform sampling, and the means and variances are estimated from samples.

Usage

```
dist_truncated(dist, lower = -Inf, upper = Inf)
```

Arguments

`dist`	The distribution(s) to truncate.
`lower, upper`	The range of values to keep from a distribution.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_truncated.html

In the following, let X be a truncated random variable with underlying distribution Y, truncation bounds `lower` = a and `upper` = b, where $F_Y(x)$ is the c.d.f. of Y and $f_Y(x)$ is the p.d.f. of Y.

Support: $[a, b]$

Mean: For the general case, the mean is approximated numerically. For a truncated Normal distribution with underlying mean μ and standard deviation σ, the mean is:

$$E(X) = \mu + \frac{\phi(\alpha) - \phi(\beta)}{\Phi(\beta) - \Phi(\alpha)}\sigma$$

where $\alpha = (a-\mu)/\sigma$, $\beta = (b-\mu)/\sigma$, ϕ is the standard Normal p.d.f., and Φ is the standard Normal c.d.f.

Variance: Approximated numerically for all distributions.

Probability density function (p.d.f):

$$f(x) = \begin{cases} \frac{f_Y(x)}{F_Y(b)-F_Y(a)} & \text{if } a \le x \le b \\ 0 & \text{otherwise} \end{cases}$$

Cumulative distribution function (c.d.f):

$$F(x) = \begin{cases} 0 & \text{if } x < a \\ \frac{F_Y(x)-F_Y(a)}{F_Y(b)-F_Y(a)} & \text{if } a \le x \le b \\ 1 & \text{if } x > b \end{cases}$$

Quantile function:

$$Q(p) = F_Y^{-1}(F_Y(a) + p(F_Y(b) - F_Y(a)))$$

clamped to the interval $[a, b]$.

Examples

```
dist <- dist_truncated(dist_normal(2,1), lower = 0)

dist
mean(dist)
variance(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)

if(requireNamespace("ggdist")) {
library(ggplot2)
ggplot() +
  ggdist::stat_dist_halfeye(
    aes(y = c("Normal", "Truncated"),
        dist = c(dist_normal(2,1), dist_truncated(dist_normal(2,1), lower = 0)))
  )
}
```

`dist_uniform` *The Uniform distribution*

Description

[Stable]

A distribution with constant density on an interval.

Usage

```
dist_uniform(min, max)
```

Arguments

`min, max`	lower and upper limits of the distribution. Must be finite.

Details

We recommend reading this documentation on pkgdown which renders math nicely. `https://pkg.mitchelloharawild.com/distributional/reference/dist_uniform.html`

In the following, let X be a Uniform random variable with parameters `min` = a and `max` = b.

Support: $[a, b]$

Mean: $\frac{a+b}{2}$

Variance: $\frac{(b-a)^2}{12}$

Probability density function (p.d.f):

$$f(x) = \frac{1}{b-a}$$

for $x \in [a, b]$, and $f(x) = 0$ otherwise.

Cumulative distribution function (c.d.f):

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

for $x \in [a, b]$, with $F(x) = 0$ for $x < a$ and $F(x) = 1$ for $x > b$.

Moment generating function (m.g.f):

$$E(e^{tX}) = \frac{e^{tb} - e^{ta}}{t(b-a)}$$

for $t \neq 0$, and $E(e^{tX}) = 1$ for $t = 0$.

Skewness: 0

Excess Kurtosis: $-\frac{6}{5}$

See Also

stats::Uniform

Examples

```
dist <- dist_uniform(min = c(3, -2), max = c(5, 4))

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

---

dist_weibull *The Weibull distribution*

Description

[Stable]

Generalization of the gamma distribution. Often used in survival and time-to-event analyses.

Usage

```
dist_weibull(shape, scale)
```

Arguments

shape, scale	shape and scale parameters, the latter defaulting to 1.

Details

We recommend reading this documentation on pkgdown which renders math nicely. https://pkg.mitchelloharawild.com/distributional/reference/dist_weibull.html

In the following, let X be a Weibull random variable with shape parameter shape = k and scale parameter scale = λ.

Support: $[0, \infty)$

Mean:

$$E(X) = \lambda\Gamma\left(1 + \frac{1}{k}\right)$$

where Γ is the gamma function.

Variance:

$$\text{Var}(X) = \lambda^2\left[\Gamma\left(1 + \frac{2}{k}\right) - \left(\Gamma\left(1 + \frac{1}{k}\right)\right)^2\right]$$

Probability density function (p.d.f):

$$f(x) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}, \quad x \geq 0$$

Cumulative distribution function (c.d.f):

$$F(x) = 1 - e^{-(x/\lambda)^k}, \quad x \geq 0$$

Moment generating function (m.g.f):

$$E(e^{tX}) = \sum_{n=0}^{\infty} \frac{t^n\lambda^n}{n!}\Gamma\left(1 + \frac{n}{k}\right)$$

Skewness:

$$\gamma_1 = \frac{\mu^3 - 3\mu\sigma^2 - \mu^3}{\sigma^3}$$

where $\mu = E(X)$, $\sigma^2 = \text{Var}(X)$, and the third raw moment is

$$\mu^3 = \lambda^3\Gamma\left(1 + \frac{3}{k}\right)$$

Excess Kurtosis:

$$\gamma_2 = \frac{\mu^4 - 4\gamma_1\mu\sigma^3 - 6\mu^2\sigma^2 - \mu^4}{\sigma^4} - 3$$

where the fourth raw moment is

$$\mu^4 = \lambda^4\Gamma\left(1 + \frac{4}{k}\right)$$

See Also

stats::Weibull

Examples

```
dist <- dist_weibull(shape = c(0.5, 1, 1.5, 5), scale = rep(1, 4))

dist
mean(dist)
variance(dist)
skewness(dist)
kurtosis(dist)

generate(dist, 10)

density(dist, 2)
density(dist, 2, log = TRUE)

cdf(dist, 4)

quantile(dist, 0.7)
```

dist_wrap *Create a distribution from p/d/q/r style functions*

Description

[Maturing]

If a distribution is not yet supported, you can vectorise p/d/q/r functions using this function. `dist_wrap()` stores the distributions parameters, and provides wrappers which call the appropriate p/d/q/r functions.

Using this function to wrap a distribution should only be done if the distribution is not yet available in this package. If you need a distribution which isn't in the package yet, consider making a request at https://github.com/mitchelloharawild/distributional/issues.

Usage

```
dist_wrap(dist, ..., package = NULL)
```

Arguments

`dist`	The name of the distribution used in the functions (name that is prefixed by p/d/q/r)
`...`	Named arguments used to parameterise the distribution.
`package`	The package from which the distribution is provided. If NULL, the calling environment's search path is used to find the distribution functions. Alternatively, an arbitrary environment can also be provided here.

Details

The `dist_wrap()` function provides a generic interface to create distribution objects from any set of p/d/q/r style functions. The statistical properties depend on the specific distribution being wrapped.

Examples

```
dist <- dist_wrap("norm", mean = 1:3, sd = c(3, 9, 2))

density(dist, 1) # dnorm()
cdf(dist, 4) # pnorm()
quantile(dist, 0.975) # qnorm()
generate(dist, 10) # rnorm()

library(actuar)
dist <- dist_wrap("invparalogis", package = "actuar", shape = 2, rate = 2)
density(dist, 1) # actuar::dinvparalogis()
cdf(dist, 4) # actuar::pinvparalogis()
quantile(dist, 0.975) # actuar::qinvparalogis()
generate(dist, 10) # actuar::rinvparalogis()
```

`family.distribution` *Extract the name of the distribution family*

Description

[Experimental]

Usage

```
## S3 method for class 'distribution'
family(object, ...)
```

Arguments

`object`	The distribution(s).
`...`	Additional arguments used by methods.

Examples

```
dist <- c(
  dist_normal(1:2),
  dist_poisson(3),
  dist_multinomial(size = c(4, 3),
  prob = list(c(0.3, 0.5, 0.2), c(0.1, 0.5, 0.4)))
  )
family(dist)
```

generate.distribution *Randomly sample values from a distribution*

Description

[Stable]

Generate random samples from probability distributions.

Usage

```
## S3 method for class 'distribution'
generate(x, times, ...)
```

Arguments

`x`	The distribution(s).
`times`	The number of samples.
`...`	Additional arguments used by methods.

has_symmetry *Check if a distribution is symmetric*

Description

[Experimental]

Determines whether a probability distribution is symmetric around its center.

Usage

```
has_symmetry(x, ...)
```

Arguments

`x`	The distribution(s).
`...`	Additional arguments used by methods.

Value

A logical value indicating whether the distribution is symmetric.

Examples

```
# Normal distribution is symmetric
has_symmetry(dist_normal(mu = 0, sigma = 1))
has_symmetry(dist_normal(mu = 5, sigma = 2))

# Beta distribution symmetry depends on parameters
has_symmetry(dist_beta(shape1 = 2, shape2 = 2))  # symmetric
has_symmetry(dist_beta(shape1 = 2, shape2 = 5))  # not symmetric
```

---

hdr *Compute highest density regions*

Description

Used to extract a specified prediction interval at a particular confidence level from a distribution.

Usage

```
hdr(x, ...)
```

Arguments

x	Object to create hilo from.
...	Additional arguments used by methods.

---

hdr.distribution *Highest density regions of probability distributions*

Description

[Maturing]

This function is highly experimental and will change in the future. In particular, improved functionality for object classes and visualisation tools will be added in a future release.

Computes minimally sized probability intervals highest density regions.

Usage

```
## S3 method for class 'distribution'
hdr(x, size = 95, n = 512, ...)
```

Arguments

x	The distribution(s).
size	The size of the interval (between 0 and 100).
n	The resolution used to estimate the distribution's density.
...	Additional arguments used by methods.

hilo — *Compute intervals*

Description

[Stable]

Used to extract a specified prediction interval at a particular confidence level from a distribution.

The numeric lower and upper bounds can be extracted from the interval using `<hilo>$lower` and `<hilo>$upper` as shown in the examples below.

Usage

```
hilo(x, ...)
```

Arguments

x	Object to create hilo from.
...	Additional arguments used by methods.

Examples

```
# 95% interval from a standard normal distribution
interval <- hilo(dist_normal(0, 1), 95)
interval

# Extract the individual quantities with `$lower`, `$upper`, and `$level`
interval$lower
interval$upper
interval$level
```

hilo.distribution — *Probability intervals of a probability distribution*

Description

[Stable]

Returns a `hilo` central probability interval with probability coverage of `size`. By default, the distribution's `quantile()` will be used to compute the lower and upper bound for a centered interval

Usage

```
## S3 method for class 'distribution'
hilo(x, size = 95, ...)
```

Arguments

`x`	The distribution(s).
`size`	The size of the interval (between 0 and 100).
`...`	Additional arguments used by methods.

See Also

`hdr.distribution()`

`is_distribution` *Test if the object is a distribution*

Description

[Stable]

This function returns `TRUE` for distributions and `FALSE` for all other objects.

Usage

```
is_distribution(x)
```

Arguments

`x`	An object.

Value

TRUE if the object inherits from the distribution class.

Examples

```
dist <- dist_normal()
is_distribution(dist)
is_distribution("distributional")
```

is_hdr *Is the object a hdr*

Description

Is the object a hdr

Usage

```
is_hdr(x)
```

Arguments

x	An object.

is_hilo *Is the object a hilo*

Description

Is the object a hilo

Usage

```
is_hilo(x)
```

Arguments

x	An object.

kurtosis *Kurtosis of a probability distribution*

Description

[Stable]

Usage

```
kurtosis(x, ...)

## S3 method for class 'distribution'
kurtosis(x, ...)
```

Arguments

x	The distribution(s).
...	Additional arguments used by methods.

`likelihood` *The (log) likelihood of a sample matching a distribution*

Description

[Stable]

Usage

```
likelihood(x, ...)

## S3 method for class 'distribution'
likelihood(x, sample, ..., log = FALSE)

log_likelihood(x, ...)
```

Arguments

x	The distribution(s).
...	Additional arguments used by methods.
sample	A list of sampled values to compare to distribution(s).
log	If TRUE, the log-likelihood will be computed.

`mean.distribution` *Mean of a probability distribution*

Description

[Stable]

Returns the empirical mean of the probability distribution. If the method does not exist, the mean of a random sample will be returned.

Usage

```
## S3 method for class 'distribution'
mean(x, ...)
```

Arguments

x	The distribution(s).
...	Additional arguments used by methods.

median.distribution *Median of a probability distribution*

Description

[Stable]

Returns the median (50th percentile) of a probability distribution. This is equivalent to `quantile(x, p=0.5)`.

Usage

```
## S3 method for class 'distribution'
median(x, na.rm = FALSE, ...)
```

Arguments

`x`	The distribution(s).
`na.rm`	Unused, included for consistency with the generic function.
`...`	Additional arguments used by methods.

new_dist *Construct distributions*

Description

[Maturing]

Allows extension package developers to define a new distribution class compatible with the distributional package.

Usage

```
new_dist(..., class = NULL, dimnames = NULL)
```

Arguments

`...`	Parameters of the distribution (named).
`class`	The class of the distribution for S3 dispatch.
`dimnames`	The names of the variables in the distribution (optional).

new_hdr *Construct hdr intervals*

Description

Construct hdr intervals

Usage

```
new_hdr(
  lower = list_of(.ptype = double()),
  upper = list_of(.ptype = double()),
  size = double()
)
```

Arguments

lower, upper	A list of numeric vectors specifying the region's lower and upper bounds.
size	A numeric vector specifying the coverage size of the region.

Value

A "hdr" vector

Author(s)

Mitchell O'Hara-Wild

Examples

```
new_hdr(lower = list(1, c(3,6)), upper = list(10, c(5, 8)), size = c(80, 95))
```

new_hilo *Construct hilo intervals*

Description

[Stable]

Class constructor function to help with manually creating hilo interval objects.

Usage

```
new_hilo(lower = double(), upper = double(), size = double())
```

Arguments

lower, upper	A numeric vector of values for lower and upper limits.
size	Size of the interval between [0, 100].

Value

A "hilo" vector

Author(s)

Earo Wang & Mitchell O'Hara-Wild

Examples

```
new_hilo(lower = rnorm(10), upper = rnorm(10) + 5, size = 95)
```

---

new_support_region   *Construct support regions*

Description

Construct support regions

Usage

```
new_support_region(x = numeric(), limits = list(), closed = list())
```

Arguments

x	A list of prototype vectors defining the distribution type.
limits	A list of value limits for the distribution.
closed	A list of logical(2L) indicating whether the limits are closed.

parameters — *Extract the parameters of a distribution*

Description

[Experimental]

Usage

```
parameters(x, ...)

## S3 method for class 'distribution'
parameters(x, ...)
```

Arguments

`x`	The distribution(s).
`...`	Additional arguments used by methods.

Examples

```
dist <- c(
  dist_normal(1:2),
  dist_poisson(3),
  dist_multinomial(size = c(4, 3),
  prob = list(c(0.3, 0.5, 0.2), c(0.1, 0.5, 0.4)))
  )
parameters(dist)
```

quantile.distribution — *Distribution Quantiles*

Description

[Stable]

Computes the quantiles of a distribution.

Usage

```
## S3 method for class 'distribution'
quantile(x, p, ..., log = FALSE)
```

Arguments

x	The distribution(s).
p	The probability of the quantile.
...	Additional arguments passed to methods.
log	If TRUE, probabilities will be given as log probabilities.

---

skewness — *Skewness of a probability distribution*

Description

[Stable]

Usage

```
skewness(x, ...)

## S3 method for class 'distribution'
skewness(x, ...)
```

Arguments

x	The distribution(s).
...	Additional arguments used by methods.

---

support — *Region of support of a distribution*

Description

[Experimental]

Usage

```
support(x, ...)

## S3 method for class 'distribution'
support(x, ...)
```

Arguments

x	The distribution(s).
...	Additional arguments used by methods.

variance *Variance*

Description

[Stable]

A generic function for computing the variance of an object.

Usage

```
variance(x, ...)

## S3 method for class 'numeric'
variance(x, ...)

## S3 method for class 'matrix'
variance(x, ...)

## S3 method for class 'numeric'
covariance(x, ...)
```

Arguments

x	An object.
...	Additional arguments used by methods.

Details

The implementation of `variance()` for numeric variables coerces the input to a vector then uses `stats::var()` to compute the variance. This means that, unlike `stats::var()`, if `variance()` is passed a matrix or a 2-dimensional array, it will still return the variance (`stats::var()` returns the covariance matrix in that case).

See Also

`variance.distribution()`, `covariance()`

variance.distribution *Variance of a probability distribution*

Description

[Stable]

Returns the empirical variance of the probability distribution. If the method does not exist, the variance of a random sample will be returned.

Usage

```
## S3 method for class 'distribution'
variance(x, ...)
```

Arguments

x	The distribution(s).
...	Additional arguments used by methods.

Index