

Package: extras (via r-universe)

October 24, 2026

Title Helper Functions for Bayesian Analyses

Version 0.10.0.9005

Description Functions to 'numericise' 'R' objects (coerce to numeric objects), summarise 'MCMC' (Monte Carlo Markov Chain) samples and calculate deviance residuals as well as 'R' translations of some 'BUGS' (Bayesian Using Gibbs Sampling), 'JAGS' (Just Another Gibbs Sampler), 'STAN' and 'TMB' (Template Model Builder) functions.

License MIT + file LICENSE

URL <https://poissonconsulting.github.io/extras/>,
<https://github.com/poissonconsulting/extras>

BugReports <https://github.com/poissonconsulting/extras/issues>

Depends R (>= 4.0.0)

Imports chk, lifecycle, stats

Suggests covr, extraDistr, ggplot2, glmnet, hms, knitr, memoise, rlang, rmarkdown, scales, sn, testthat (>= 3.0.0), tibble, tidyr, viridis, withr

VignetteBuilder knitr

Config/Needs/website poissonconsulting/poissontemplate

Config/testthat/edition 3

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.0.9000

Repository <https://poissonconsulting.r-universe.dev>

Date/Publication 2026-10-24 07:20:05 UTC

RemoteUrl <https://github.com/poissonconsulting/extras>

RemoteRef HEAD

RemoteSha d8604664fe0c70d61366ce1661e94ee3d26d2dbd

Contents

as_list	5
as_list_unnamed	6
chk_index	7
chk_indices	7
chk_pars	8
dbern	9
dev_bern	10
dev_beta_binom	11
dev_binom	12
dev_gamma	12
dev_gamma_pois	13
dev_gamma_pois_zi	14
dev_lnorm	15
dev_multinom	15
dev_neg_binom	17
dev_norm	17
dev_pois	18
dev_pois_zi	19
dev_skewlnorm	20
dev_skewnorm	21
dev_student	22
direction	23
directional_information	24
dskewlnorm	26
dskewnorm	28
exp10	29
exp2	30
fabs	30
fill_all	31
fill_na	32
ilog	34
ilog10	35
ilog2	35
ilogit	36
inv_logit	37
inv_odds	37
invlogit	38
kurtosis	39
log_lik_bern	39
log_lik_beta	40
log_lik_beta_binom	41
log_lik_binom	42
log_lik_exp	43
log_lik_gamma	43
log_lik_gamma_pois	44
log_lik_gamma_pois_zi	45

log_lik_lnorm	46
log_lik_multinom	47
log_lik_neg_binom	48
log_lik_norm	49
log_lik_pois	50
log_lik_pois_zi	50
log_lik_skewlnorm	51
log_lik_skewnorm	52
log_lik_student	54
log_lik_unif	55
log_odds	55
log_odds_ratio	56
log_odds_ratio2	57
log_odds<-	57
log<-	58
log10<-	59
log2<-	59
logit	60
logit<-	61
lower	62
numericise	62
odds	65
odds_ratio	65
odds_ratio2	66
odds<-	67
par_pattern	67
pextreme	68
phi	68
pow	69
prob_bern	70
prob_beta	70
prob_beta_binom	71
prob_binom	72
prob_exp	73
prob_gamma	73
prob_gamma_pois	74
prob_gamma_pois_zi	75
prob_lnorm	75
prob_neg_binom	76
prob_norm	77
prob_pois	78
prob_pois_zi	78
prob_skewlnorm	79
prob_skewnorm	80
prob_student	81
prob_unif	82
probability_direction	82
proportional_change	84

proportional_change2	84
proportional_difference	85
proportional_difference2	86
pvalue	86
pzeros	88
quant_bern	88
quant_beta	89
quant_binom	90
quant_exp	90
quant_gamma	91
quant_gamma_pois	92
quant_gamma_pois_zi	92
quant_lnorm	93
quant_neg_binom	94
quant_norm	95
quant_pois	95
quant_pois_zi	96
quant_skewlnorm	97
quant_skewnorm	98
quant_student	99
quant_unif	99
ran_bern	100
ran_beta_binom	101
ran_binom	102
ran_gamma	102
ran_gamma_pois	103
ran_gamma_pois_zi	104
ran_lnorm	104
ran_multinom	105
ran_neg_binom	106
ran_norm	107
ran_pois	108
ran_pois_zi	108
ran_skewlnorm	109
ran_skewnorm	110
ran_student	111
res_bern	112
res_beta_binom	112
res_binom	113
res_gamma	114
res_gamma_pois	115
res_gamma_pois_zi	116
res_lnorm	117
res_multinom	117
res_neg_binom	119
res_norm	120
res_pois	120
res_pois_zi	121

res_skewlnorm	122
res_skewnorm	123
res_student	124
sens_beta	125
sens_exp	126
sens_gamma	127
sens_gamma_pois	127
sens_gamma_pois_zi	128
sens_lnorm	129
sens_neg_binom	130
sens_norm	130
sens_pois	131
sens_skewlnorm	132
sens_skewnorm	133
sens_student	134
sextreme	135
skewness	135
step	136
svalue	137
upper	138
variance	139
xtr_ci	140
xtr_ci_eti	141
xtr_ci_hdi	142
xtr_mean	143
xtr_median	143
xtr_rope	144
xtr_sd	145
zeros	146
zscore	146

Index	148
--------------	------------

as_list	<i>As List</i>
---------	----------------

Description

Coerces an object to an list. All attributes are removed except any names.

Usage

```
as_list(x, ...)
```

```
## Default S3 method:
```

```
as_list(x, ...)
```

Arguments

`x` An object.

`...` Other arguments passed to methods.

Value

A list.

Examples

```
as_list(1:3)
as_list(c(x = 1, y = 2))
```

as_list_unnamed	<i>As List</i>
-----------------	----------------

Description

Coerces an object to an list. All attributes are removed except any names. **[Deprecated]**

Usage

```
as_list_unnamed(x, ...)

## Default S3 method:
as_list_unnamed(x, ...)
```

Arguments

`x` An object.

`...` Other arguments passed to methods.

Value

A list.

Examples

```
as_list_unnamed(1:3)
as_list_unnamed(c(x = 1, y = 2))
```

chk_index	<i>Check Index</i>
-----------	--------------------

Description

Checks if an object is a vector of one or more positive integer values.

Usage

```
chk_index(x, x_name = NULL)
```

```
vld_index(x)
```

Arguments

x	An object.
x_name	A string of the name of object x or NULL.

Value

The `chk_` function throws an informative error if the test fails.

The `vld_` function returns a flag indicating whether the test was met.

Functions

- `vld_index()`: Validate Index

Examples

```
x <- c(2L, 1L)
chk_index(x)
y <- c(2L, -1L)
try(chk_index(y))
vld_index(c(-1))
vld_index(c(3L, 1L))
```

chk_indices	<i>Check Indices</i>
-------------	----------------------

Description

Checks if an object is a list of indices ie vectors of one or more positive integer values.

Usage

```
chk_indices(x, x_name = NULL)
```

```
vld_indices(x)
```

Arguments

`x` An object.

`x_name` A string of the name of object `x` or `NULL`.

Value

The `chk_` function throws an informative error if the test fails.

The `vld_` function returns a flag indicating whether the test was met.

Functions

- `vld_indices()`: Validate Indices

Examples

```
x <- list(c(2L, 1L))
chk_indices(x)
y <- c(2L, 1L)
try(chk_indices(y))
vld_indices(c(3L, 1L))
vld_indices(list(c(3L, 1L)))
```

 chk_pars

Check Parameter Names

Description

Checks if valid parameter names.

Usage

```
chk_pars(x, x_name = NULL)
```

```
vld_pars(x)
```

Arguments

`x` An object.

`x_name` A string of the name of object `x` or `NULL`.

Details

The character vector must consist of values that start with an alpha and only include alphanumeric characters and '_' or '.'.

Missing values and duplicates are permitted.

Value

The `chk_` function throws an informative error if the test fails.

The `vld_` function returns a flag indicating whether the test was met.

Functions

- `vld_pars()`: Validate Parameter Names

Examples

```
x <- c("x", "a1._", "X")
chk_pars(x)
y <- c("x[1]", "a1", "a1", "._0")
try(chk_pars(y))
vld_pars(c("x", "a1._", "X"))
vld_pars(c("x[1]", "a1", "a1", "._0"))
```

dbern

Bernoulli Distribution

Description

Bernoulli Distribution

Usage

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

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

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

rbern(n, prob)
```

Arguments

<code>x</code>	A vector of 0s and 1s.
<code>prob</code>	A numeric vector of values between 0 and 1 of the probability of success.
<code>log</code>	A flag specifying whether to return the log-transformed value.
<code>q</code>	A vector of quantiles.

lower.tail	A flag specifying whether to return the lower or upper tail of the distribution.
p	A numeric vector of probabilities.
n	A non-negative whole number of the number of random samples to generate.

Value

An numeric vector of the random samples.

Examples

```
dbern(1, 0.5)
pbern(0.75, 0.5)
qbern(0.1, 0.5)
rbern(1, 0.5)
```

dev_bern

*Bernoulli Deviances***Description**

Bernoulli Deviances

Usage

```
dev_bern(x, prob = 0.5, res = FALSE)
```

Arguments

x	A vector of 0s and 1s.
prob	A numeric vector of values between 0 and 1 of the probability of success.
res	A flag specifying whether to return the deviance residual as opposed to the deviance.

Value

An numeric vector of the corresponding deviances or deviance residuals.

See Also

Other dev_dist: [dev_beta_binom\(\)](#), [dev_binom\(\)](#), [dev_gamma\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_multinom\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [dev_skewlnorm\(\)](#), [dev_skewnorm\(\)](#), [dev_student\(\)](#)

Examples

```
dev_bern(c(TRUE, FALSE), 0.7)
```

dev_beta_binom *Beta-Binomial Deviances*

Description

This parameterization of the beta-binomial distribution uses an expected probability parameter, prob, and a dispersion parameter, theta. The parameters of the underlying beta mixture are $\alpha = (2 * \text{prob}) / \text{theta}$ and $\beta = (2 * (1 - \text{prob})) / \text{theta}$. This parameterization of theta is unconventional, but has useful properties when modelling. When $\text{theta} = 0$, the beta-binomial reverts to the binomial distribution. When $\text{theta} = 1$ and $\text{prob} = 0.5$, the parameters of the beta distribution become $\alpha = 1$ and $\beta = 1$, which correspond to a uniform distribution for the beta-binomial probability parameter.

Usage

```
dev_beta_binom(x, size = 1, prob = 0.5, theta = 0, res = FALSE)
```

Arguments

x	A non-negative whole numeric vector of values.
size	A non-negative whole numeric vector of the number of trials.
prob	A numeric vector of values between 0 and 1 of the probability of success.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
res	A flag specifying whether to return the deviance residual as opposed to the deviance.

Value

An numeric vector of the corresponding deviances or deviance residuals.

See Also

Other dev_dist: [dev_bern\(\)](#), [dev_binom\(\)](#), [dev_gamma\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_multinom\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [dev_skewlnorm\(\)](#), [dev_skewnorm\(\)](#), [dev_student\(\)](#)

Examples

```
dev_beta_binom(c(0, 1, 2), 10, 0.5, 0.1)
```

dev_binom

*Binomial Deviances***Description**

Binomial Deviances

Usage

```
dev_binom(x, size = 1, prob = 0.5, res = FALSE)
```

Arguments

x	A non-negative whole numeric vector of values.
size	A non-negative whole numeric vector of the number of trials.
prob	A numeric vector of values between 0 and 1 of the probability of success.
res	A flag specifying whether to return the deviance residual as opposed to the deviance.

Value

An numeric vector of the corresponding deviances or deviance residuals.

See Also

Other dev_dist: [dev_bern\(\)](#), [dev_beta_binom\(\)](#), [dev_gamma\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_multinom\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [dev_skewlnorm\(\)](#), [dev_skewnorm\(\)](#), [dev_student\(\)](#)

Examples

```
dev_binom(c(0, 1, 2), 2, 0.3)
```

dev_gamma

*Gamma Deviances***Description**

Gamma Deviances

Usage

```
dev_gamma(x, shape = 1, rate = 1, res = FALSE)
```

Arguments

x	A numeric vector of values.
shape	A non-negative numeric vector of shape.
rate	A non-negative numeric vector of rate.
res	A flag specifying whether to return the deviance residual as opposed to the deviance.

Value

An numeric vector of the corresponding deviances or deviance residuals.

See Also

Other dev_dist: [dev_bern\(\)](#), [dev_beta_binom\(\)](#), [dev_binom\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_multinom\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [dev_skewlnorm\(\)](#), [dev_skewnorm\(\)](#), [dev_student\(\)](#)

Examples

```
dev_gamma(c(0, 1, 2), 1, 2)
```

dev_gamma_pois	<i>Gamma-Poisson Deviances</i>
----------------	--------------------------------

Description

Gamma-Poisson Deviances

Usage

```
dev_gamma_pois(x, lambda = 1, theta = 0, res = FALSE)
```

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
res	A flag specifying whether to return the deviance residual as opposed to the deviance.

Value

An numeric vector of the corresponding deviances or deviance residuals.

See Also

Other dev_dist: `dev_bern()`, `dev_beta_binom()`, `dev_binom()`, `dev_gamma()`, `dev_lnorm()`, `dev_multinom()`, `dev_neg_binom()`, `dev_norm()`, `dev_pois()`, `dev_pois_zi()`, `dev_skewlnorm()`, `dev_skewnorm()`, `dev_student()`

Examples

```
dev_gamma_pois(c(1, 3, 4), 3, 2)
```

dev_gamma_pois_zi	<i>Zero-Inflated Gamma-Poisson Deviances</i>
-------------------	--

Description

Zero-Inflated Gamma-Poisson Deviances

Usage

```
dev_gamma_pois_zi(x, lambda = 1, theta = 0, prob = 0, res = FALSE)
```

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
prob	A numeric vector of values between 0 and 1 of the probability of success.
res	A flag specifying whether to return the deviance residual as opposed to the deviance.

Value

An numeric vector of the corresponding deviances or deviance residuals.

Examples

```
dev_gamma_pois_zi(c(1, 3, 4), 3, 2)
```

dev_lnorm	<i>Log-Normal Deviances</i>
-----------	-----------------------------

Description

Log-Normal Deviances

Usage

```
dev_lnorm(x, meanlog = 0, sdlog = 1, res = FALSE)
```

Arguments

x	A numeric vector of values.
meanlog	A numeric vector of the means on the log scale.
sdlog	A non-negative numeric vector of the standard deviations on the log scale.
res	A flag specifying whether to return the deviance residual as opposed to the deviance.

Value

An numeric vector of the corresponding deviances or deviance residuals.

See Also

Other dev_dist: [dev_bern\(\)](#), [dev_beta_binom\(\)](#), [dev_binom\(\)](#), [dev_gamma\(\)](#), [dev_gamma_pois\(\)](#), [dev_multinom\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [dev_skewlnorm\(\)](#), [dev_skewnorm\(\)](#), [dev_student\(\)](#)

Examples

```
dev_lnorm(exp(-2:2))
```

dev_multinom	<i>Multinomial Deviances</i>
--------------	------------------------------

Description

Models the counts across two or more mutually exclusive categories from a fixed number of trials, in *long* format: one row per category per trial, with group identifying which rows belong to the same trial.

Usage

```
dev_multinom(x, size = 1, prob, group, res = FALSE)
```

Arguments

<code>x</code>	A non-negative whole numeric vector of the category counts.
<code>size</code>	A non-negative whole numeric vector of the number of trials.
<code>prob</code>	A numeric vector of the probability of the category. Must sum to 1 across the rows sharing the same group.
<code>group</code>	A vector identifying which rows belong to the same multinomial trial (whose <code>x</code> values sum to <code>size</code> and <code>prob</code> values sum to 1). Every group must have at least 2 rows and the same number of rows as the rest of the data (a fixed set of categories, as in multinomial logistic regression), and must not contain NA.
<code>res</code>	A flag specifying whether to return the deviance residual as opposed to the deviance.

Details

A category's deviance depends only on its own `x` and $\mu = \text{size} * \text{prob}$, not on the rest of its trial, so `group` is used only to validate `size` and `prob` (see `log_lik_multinom()`), not in the calculation itself. `dev_multinom()` is the Poisson-equivalent deviance (see `dev_pois()`): summing it over a trial's rows recovers the trial's exact multinomial deviance.

Value

An numeric vector of the corresponding deviances or deviance residuals.

References

- McCullagh, P., and Nelder, J.A. 1989. Generalized Linear Models. 2nd edition. Chapman and Hall, London.
- Baker, S.G. 1994. The multinomial-Poisson transformation. The Statistician 43(4): 495-504. [doi:10.2307/2348134](https://doi.org/10.2307/2348134).
- Agresti, A. 2013. Categorical Data Analysis. 3rd edition. John Wiley and Sons, Hoboken, New Jersey.

See Also

Other `dev_dist`: `dev_bern()`, `dev_beta_binom()`, `dev_binom()`, `dev_gamma()`, `dev_gamma_pois()`, `dev_lnorm()`, `dev_neg_binom()`, `dev_norm()`, `dev_pois()`, `dev_pois_zi()`, `dev_skewlnorm()`, `dev_skewnorm()`, `dev_student()`

Examples

```
dev_multinom(c(1, 3, 6), size = 10, prob = c(0.2, 0.3, 0.5), group = c(1, 1, 1))
```

dev_neg_binom	<i>Negative Binomial Deviances</i>
---------------	------------------------------------

Description

Negative Binomial Deviances

Usage

```
dev_neg_binom(x, lambda = 1, theta = 0, res = FALSE)
```

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
res	A flag specifying whether to return the deviance residual as opposed to the deviance.

Value

An numeric vector of the corresponding deviances or deviance residuals.

See Also

Other dev_dist: [dev_bern\(\)](#), [dev_beta_binom\(\)](#), [dev_binom\(\)](#), [dev_gamma\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_multinom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [dev_skewlnorm\(\)](#), [dev_skewnorm\(\)](#), [dev_student\(\)](#)

Examples

```
dev_neg_binom(c(1, 2, 5), 2, 3)
```

dev_norm	<i>Normal Deviances</i>
----------	-------------------------

Description

Normal Deviances

Usage

```
dev_norm(x, mean = 0, sd = 1, res = FALSE)
```

Arguments

x	A numeric vector of values.
mean	A numeric vector of the means.
sd	A non-negative numeric vector of the standard deviations.
res	A flag specifying whether to return the deviance residual as opposed to the deviance.

Value

An numeric vector of the corresponding deviances or deviance residuals.

See Also

Other dev_dist: [dev_bern\(\)](#), [dev_beta_binom\(\)](#), [dev_binom\(\)](#), [dev_gamma\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_multinom\(\)](#), [dev_neg_binom\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [dev_skewlnorm\(\)](#), [dev_skewnorm\(\)](#), [dev_student\(\)](#)

Examples

```
dev_norm(c(-2:2))
```

dev_pois

Poisson Deviances

Description

Poisson Deviances

Usage

```
dev_pois(x, lambda, res = FALSE)
```

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
res	A flag specifying whether to return the deviance residual as opposed to the deviance.

Value

An numeric vector of the corresponding deviances or deviance residuals.

See Also

Other dev_dist: [dev_bern\(\)](#), [dev_beta_binom\(\)](#), [dev_binom\(\)](#), [dev_gamma\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_multinom\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois_zi\(\)](#), [dev_skewlnorm\(\)](#), [dev_skewnorm\(\)](#), [dev_student\(\)](#)

Examples

```
dev_pois(c(1, 3, 4), 3)
```

dev_pois_zi	<i>Zero-Inflated Poisson Deviances</i>
-------------	--

Description

Zero-Inflated Poisson Deviances

Usage

```
dev_pois_zi(x, lambda, prob = 0, res = FALSE)
```

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
prob	A numeric vector of values between 0 and 1 of the probability of success.
res	A flag specifying whether to return the deviance residual as opposed to the deviance.

Value

An numeric vector of the corresponding deviances or deviance residuals.

See Also

Other dev_dist: [dev_bern\(\)](#), [dev_beta_binom\(\)](#), [dev_binom\(\)](#), [dev_gamma\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_multinom\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_skewlnorm\(\)](#), [dev_skewnorm\(\)](#), [dev_student\(\)](#)

Examples

```
dev_pois_zi(c(1, 3, 4), 3)
```

dev_skewlnorm	<i>Skew-Lognormal Deviances</i>
---------------	---------------------------------

Description

Skew-Lognormal Deviances

Usage

```
dev_skewlnorm(
  x,
  locationlog = 0,
  scalelog = 1,
  shapelog = 0,
  res = FALSE,
  ...,
  meanlog,
  sdlog,
  shape
)
```

Arguments

x	A numeric vector of values.
locationlog	A numeric vector of location parameters of $\log(x)$.
scalelog	A non-negative numeric vector of scale parameters of $\log(x)$.
shapelog	A numeric vector of shape parameters of $\log(x)$. Negative values result in leftward skew, while positive values result in rightward skew.
res	A flag specifying whether to return the deviance residual as opposed to the deviance.
...	Other arguments passed to methods.
meanlog	A numeric vector of the means on the log scale.
sdlog	A non-negative numeric vector of the standard deviations on the log scale.
shape	A non-negative numeric vector of shape.

Value

An numeric vector of the corresponding deviances or deviance residuals.

See Also

Other dev_dist: [dev_bern\(\)](#), [dev_beta_binom\(\)](#), [dev_binom\(\)](#), [dev_gamma\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_multinom\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [dev_skewnorm\(\)](#), [dev_student\(\)](#)

Examples

```
dev_skewlnorm(exp(-2:2))
dev_skewlnorm(exp(-2:2), 0, 1, 5)
dev_skewlnorm(exp(-2:2), 0, 1, 5, res = TRUE)
```

dev_skewnorm	<i>Skew Normal Deviances</i>
--------------	------------------------------

Description

Skew Normal Deviances

Usage

```
dev_skewnorm(x, location = 0, scale = 1, shape = 0, res = FALSE, ..., mean, sd)
```

Arguments

x	A numeric vector of values.
location	A numeric vector of the location parameter.
scale	A non-negative numeric vector of the scale parameter.
shape	A non-negative numeric vector of shape.
res	A flag specifying whether to return the deviance residual as opposed to the deviance.
...	Other arguments passed to methods.
mean	A numeric vector of the means.
sd	A non-negative numeric vector of the standard deviations.

Value

An numeric vector of the corresponding deviances or deviance residuals.

See Also

Other dev_dist: [dev_bern\(\)](#), [dev_beta_binom\(\)](#), [dev_binom\(\)](#), [dev_gamma\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_multinom\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [dev_skewlnorm\(\)](#), [dev_student\(\)](#)

Examples

```
dev_skewnorm(c(-2:2))
dev_skewnorm(-2:2, 0, 1, 5)
dev_skewnorm(-2:2, 0, 1, 5, res = TRUE)
```

dev_student	<i>Student's t Deviances</i>
-------------	------------------------------

Description

Student's t Deviances

Usage

```
dev_student(x, mean = 0, sd = 1, theta = 0, res = FALSE)
```

Arguments

x	A numeric vector of values.
mean	A numeric vector of the means.
sd	A non-negative numeric vector of the standard deviations.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
res	A flag specifying whether to return the deviance residual as opposed to the deviance.

Value

An numeric vector of the corresponding deviances or deviance residuals.

See Also

Other dev_dist: [dev_bern\(\)](#), [dev_beta_binom\(\)](#), [dev_binom\(\)](#), [dev_gamma\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_multinom\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [dev_skewlnorm\(\)](#), [dev_skewnorm\(\)](#)

Examples

```
dev_student(c(1, 3.5, 4), 3)
```

direction	<i>Direction of a distribution</i>
-----------	------------------------------------

Description

The direction of a distribution is the side (left/right) that the distribution's center falls on, relative to a threshold. The center can be calculated using a user-specified function, including the median (default), mean, geometric mean, mode, or any other custom function. By convention, values below the threshold fall to the left, while values above the threshold fall to the right. Center estimates equal to the threshold are assumed to fall to the right.

Usage

```
direction(x, estimate = xtr_median, threshold = 0, na_rm = FALSE)
```

Arguments

x	A numeric vector of MCMC values or any other numeric vector of samples.
estimate	A function for estimating the center of the distribution. Defaults to <code>xtr_median()</code> , but can also be <code>xtr_mean()</code> or any custom function that returns a number (a non-missing numeric vector of length 1). If <code>na_rm</code> is true, NA values are dropped before calling the function.
threshold	A number of the threshold value.
na_rm	A flag specifying whether to remove missing values.

Value

A string indicating if at least half of the observations are above the threshold ("right") or not ("left"), or NA_character_ in the case of missing values when `na_rm = FALSE`.

See Also

Other summary: `directional_information()`, `kurtosis()`, `lower()`, `probability_direction()`, `pvalue()`, `pzeros()`, `skewness()`, `svalue()`, `upper()`, `variance()`, `xtr_mean()`, `xtr_median()`, `xtr_sd()`, `zeros()`, `zscore()`

Examples

```
direction(c(1, 2, 3))
direction(c(-1))
direction(c(0, 0, 0))
direction(c(-100, 1, 1))
direction(c(-100, 1, 1), mean)
direction(c(100, 0.01, 0.01), function(.x) exp(mean(log(.x))))
```

directional_information

Directional information

Description

The occurrence of an event (or lack thereof) transmits more or less information depending on the event's probability.

Usage

```
directional_information(
  x,
  ...,
  side = "median",
  threshold = 0,
  threshold_split = "proportional",
  skeptical = TRUE,
  na_rm = FALSE
)

p2info(p, n = Inf)
```

Arguments

x	A numeric vector of MCMC values.
...	Unused.
side	A string indicating whether to calculate the directional information relative to the left side ("left"; $x < \text{threshold}$), or the right side ("right"; $x > \text{threshold}$). Positive information suggests greater evidence for the specified side. Defaults to "median", which uses the side of the median of x via direction() .
threshold	A number of the threshold value.
threshold_split	A string indicating how to deal with threshold values: <ul style="list-style-type: none"> • "left" to include them on the left side, • "right" to include them on the right side, • "equal" to split them equally between the left and side, • "proportional" (default) to split them between the left and right sides proportionally to the values of x on the left and right sides, • "exclude" to drop the values of x equal to threshold (identical to using "proportional").
skeptical	A flag specifying whether or not to add one sample to the empty side of the threshold when 100% of samples are on one side. Avoids zero p-values and infinite s-values, and also imposes stronger bounds on directional information than $[-n, n]$, which assume the MCMC samples are independent and representative.

na_rm	A flag specifying whether to remove missing values.
p	A numeric vector of probabilities of direction.
n	A numeric vector of the number of posterior samples used to estimate each value of p. Used to limit the information to be within the interval $[-n, n]$.

Details

Quantifies the information about direction in a posterior distribution based on the directional probability. This function calculates such information using the difference in the probability of direction (see `probability_direction()`), after converting each probability to bits (also see `svalue()`).

When `skeptical = TRUE` (default), one sample is added to the empty side, giving bounds of $\pm \log_2(n)$ rather than $\pm n$, to mimic the behaviour of `pvalue()` and `svalue()`. When `skeptical = FALSE`, information is instead clamped to $[-n, n]$, which assumes the MCMC samples are independent and representative.

Value

A number indicating the directional information in bits. If x has NA values but `na_rm` is FALSE, returns `NA_real`.

Functions

- `directional_information()`: Calculate the directional information from a posterior distribution.
- `p2info()`: Calculate the information from a vector of probabilities.

References

Kery, M., and Schaub, M. 2011. Bayesian population analysis using WinBUGS: a hierarchical perspective. Academic Press, Boston. Available from <https://www.vogelwarte.ch/en/research/population-biology/book-bpa/>.

See Also

Other summary: `direction()`, `kurtosis()`, `lower()`, `probability_direction()`, `pvalue()`, `pzeros()`, `skewness()`, `svalue()`, `upper()`, `variance()`, `xtr_mean()`, `xtr_median()`, `xtr_sd()`, `zeros()`, `zscore()`

Examples

```
directional_information(0)
directional_information(1) # one coin flip of information
directional_information(c(1, 1)) # two coin flips
directional_information(c(1, 1, -1)) # x[2] and x[3] cancel out
directional_information(c(1, 1, -1, -1)) # both sides cancel out
directional_information(rnorm(1e3, mean = 0))
directional_information(rnorm(1e3, mean = 1))
directional_information(rnorm(1e3, mean = 10)) # all coin flips are positive
directional_information(rnorm(1e3, mean = -10)) # all coin flips are negative
```

```
directional_information(rnorm(1e3, mean = 1e3)) # only quantiles matter
directional_information(rnorm(1e6, mean = 1e3)) # more `x` implies more info
directional_information(rep(1, 1000)) # skeptical = TRUE (default) gives log2(n)
directional_information(rep(1, 1000), skeptical = FALSE) # skeptical = FALSE gives n

p2info(seq(0, 1, by = 0.1))
p2info(seq(0, 1, by = 0.1), n = 10) # limit information to be in [-10, 10]
```

dskewlnorm

Skew-Lognormal Distribution

Description

The skew-lognormal distribution of a random variable whose natural logarithm follows a [Skew-Normal](#) distribution with location `locationlog`, scale `scalelog` and `shapelog`. It reduces to the Log-Normal distribution when `shapelog = 0`.

Usage

```
dskewlnorm(
  x,
  locationlog = 0,
  scalelog = 1,
  shapelog = 0,
  log = FALSE,
  ...,
  meanlog,
  sdlog,
  shape
)
```

```
pskewlnorm(
  q,
  locationlog = 0,
  scalelog = 1,
  shapelog = 0,
  ...,
  meanlog,
  sdlog,
  shape
)
```

```
qskewlnorm(
  p,
  locationlog = 0,
  scalelog = 1,
  shapelog = 0,
```

```

    ...,
    meanlog,
    sdlog,
    shape
)

rskewlnorm(
  n = 1,
  locationlog = 0,
  scalelog = 1,
  shapelog = 0,
  ...,
  meanlog,
  sdlog,
  shape
)

```

Arguments

x	A numeric vector of values.
locationlog	A numeric vector of location parameters of $\log(x)$.
scalelog	A non-negative numeric vector of scale parameters of $\log(x)$.
shapelog	A numeric vector of shape parameters of $\log(x)$. Negative values result in leftward skew, while positive values result in rightward skew.
log	A flag specifying whether to return the log-transformed value.
...	Unused.
meanlog	[Deprecated] A numeric vector of location parameters of $\log(x)$. Described as "a numeric vector of the means on the log scale" prior to v. 0.10.1. Will be removed in a future version.
sdlog	[Deprecated] A non-negative numeric vector of scale parameters of $\log(x)$. Described as "a non-negative numeric vector of the standard deviations on the log scale" prior to v. 0.10.1. Will be removed in a future version.
shape	[Deprecated] A numeric vector of shape parameters of $\log(x)$. Will be removed in a future version.
q	A vector of quantiles.
p	A numeric vector of probabilities.
n	A non-negative whole number of the number of random samples to generate.

Value

dskewlnorm gives the density, pskewlnorm gives the distribution function, qskewlnorm gives the quantile function, and rskewlnorm generates random deviates. pskewlnorm and qskewlnorm use the lower tail probability.

Examples

```
dskewlnorm(x = 1:5, locationlog = 0, scalelog = 1, shapelog = 0.1)
dskewlnorm(x = 1:5, locationlog = 0, scalelog = 1, shapelog = -1)
qskewlnorm(p = c(0.1, 0.4), locationlog = 0, scalelog = 1, shapelog = 0.1)
qskewlnorm(p = c(0.1, 0.4), locationlog = 0, scalelog = 1, shapelog = -1)
pskewlnorm(q = 1:5, locationlog = 0, scalelog = 1, shapelog = 0.1)
pskewlnorm(q = 1:5, locationlog = 0, scalelog = 1, shapelog = -1)
rskewlnorm(n = 3, locationlog = 0, scalelog = 1, shapelog = 0.1)
rskewlnorm(n = 3, locationlog = 0, scalelog = 1, shapelog = -1)
```

dskewnorm

Skew-Normal Distribution

Description

Skew-Normal Distribution

Usage

```
dskewnorm(x, location = 0, scale = 1, shape = 0, log = FALSE, ..., mean, sd)

pskewnorm(q, location = 0, scale = 1, shape = 0, ..., mean, sd)

qskewnorm(p, location = 0, scale = 1, shape = 0, ..., mean, sd)

rskewnorm(n = 1, location = 0, scale = 1, shape = 0, ..., mean, sd)
```

Arguments

x	A numeric vector of values.
location	A numeric vector of the location parameter.
scale	A non-negative numeric vector of the scale parameter.
shape	A numeric vector of shape. Negative values result in leftward skew, while positive values result in rightward skew.
log	A flag specifying whether to return the log-transformed value.
...	Unused.
mean	[Deprecated] A numeric vector of the location parameter. Described as "a numeric vector of the means" prior to v. 0.10.1. Will be removed in a future version.
sd	[Deprecated] A non-negative numeric vector of the scale parameter. Described as "a non-negative numeric vector of the standard deviations" prior to v. 0.10.1. Will be removed in a future version.
q	A vector of quantiles.
p	A numeric vector of probabilities.
n	A non-negative whole number of the number of random samples to generate.

Value

dskewnorm gives the density, pskewnorm gives the distribution function, qskewnorm gives the quantile function, and rskewnorm generates random deviates. pskewnorm and qskewnorm use the lower tail probability.

Examples

```
dskewnorm(x = -2:2, location = 0, scale = 1, shape = 0.1)
dskewnorm(x = -2:2, location = 0, scale = 1, shape = -1)
qskewnorm(p = c(0.1, 0.4), location = 0, scale = 1, shape = 0.1)
qskewnorm(p = c(0.1, 0.4), location = 0, scale = 1, shape = -1)
pskewnorm(q = -2:2, location = 0, scale = 1, shape = 0.1)
pskewnorm(q = -2:2, location = 0, scale = 1, shape = -1)
rskewnorm(n = 3, location = 0, scale = 1, shape = 0.1)
rskewnorm(n = 3, location = 0, scale = 1, shape = -1)
```

exp10

*Exponential Transformation of Base 10***Description**

Returns the transformation of 10^x .

Usage

```
exp10(x)
```

Arguments

x An numeric atomic object.

Value

A numeric atomic object with the value of 10^x .

See Also

Other translations: [exp2\(\)](#), [fabs\(\)](#), [ilog\(\)](#), [ilog10\(\)](#), [ilog2\(\)](#), [ilogit\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [log10<-\(\)](#), [log2<-\(\)](#), [log<-\(\)](#), [logit\(\)](#), [logit<-\(\)](#), [phi\(\)](#), [pow\(\)](#), [step\(\)](#)

Examples

```
x <- c(5, 10.5)
exp10(x)
```

exp2

Exponential Transformation of Base 2

Description

Returns the transformation of 2^x .

Usage

```
exp2(x)
```

Arguments

x An numeric atomic object.

Value

A numeric atomic object with the value of 2^x .

See Also

Other translations: [exp10\(\)](#), [fabs\(\)](#), [ilog\(\)](#), [ilog10\(\)](#), [ilog2\(\)](#), [ilogit\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [log10<-\(\)](#), [log2<-\(\)](#), [log<-\(\)](#), [logit\(\)](#), [logit<-\(\)](#), [phi\(\)](#), [pow\(\)](#), [step\(\)](#)

Examples

```
x <- c(5, 10.5)
exp2(x)
```

fabs

Absolute

Description

Computes the absolute value of x. Used in TMB as replacement for `abs()` which is seemingly ambiguous.

Usage

```
fabs(x)
```

Arguments

x An existing R object.

Details

A wrapper on [abs\(\)](#).

Value

A numeric vector of the corresponding absolute values.

See Also

Other translations: [exp10\(\)](#), [exp2\(\)](#), [ilog\(\)](#), [ilog10\(\)](#), [ilog2\(\)](#), [ilogit\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [log10<-\(\)](#), [log2<-\(\)](#), [log<-\(\)](#), [logit\(\)](#), [logit<-\(\)](#), [phi\(\)](#), [pow\(\)](#), [step\(\)](#)

Examples

```
fabs(c(0, -1, 2))
```

fill_all

Fill All Values

Description

Fills all of an object's (missing and non-missing) values while preserving the object's dimensionality and class.

Usage

```
fill_all(x, value, ...)

## S3 method for class 'logical'
fill_all(x, value = FALSE, nas = TRUE, ...)

## S3 method for class 'integer'
fill_all(x, value = 0L, nas = TRUE, ...)

## S3 method for class 'numeric'
fill_all(x, value = 0, nas = TRUE, ...)

## S3 method for class 'character'
fill_all(x, value = "0", nas = TRUE, ...)
```

Arguments

x	An object.
value	A scalar of the value to replace values with.
...	Other arguments passed to methods.
nas	A flag specifying whether to also fill missing values.

Details

It should only be defined for objects with values of consistent class ie not standard data.frames.

Value

The modified object.

Methods (by class)

- `fill_all(logical)`: Fill All for logical Objects
- `fill_all(integer)`: Fill All for integer Objects
- `fill_all(numeric)`: Fill All for numeric Objects
- `fill_all(character)`: Fill All for character Objects

See Also

Other fill: [fill_na\(\)](#)

Examples

```
# logical
fill_all(c(TRUE, NA, FALSE))
fill_all(c(TRUE, NA, FALSE, nas = FALSE))
fill_all(c(TRUE, NA, FALSE, value = NA))

# integer
fill_all(matrix(1:4, nrow = 2), value = -1)

# numeric
fill_all(c(1, 4, NA), value = TRUE)
fill_all(c(1, 4, NA), value = TRUE, nas = FALSE)

# character
fill_all(c("some", "words"), value = TRUE)
```

fill_na

Fill Missing Values

Description

Fills all of an object's missing values while preserving the object's dimensionality and class.

Usage

```
fill_na(x, value, ...)

## S3 method for class 'logical'
fill_na(x, value = FALSE, ...)

## S3 method for class 'integer'
fill_na(x, value = 0L, ...)

## S3 method for class 'numeric'
fill_na(x, value = 0, ...)

## S3 method for class 'character'
fill_na(x, value = "", ...)
```

Arguments

x	An object.
value	A scalar of the value to replace values with.
...	Other arguments passed to methods.

Details

It should only be defined for objects with values of consistent class ie not standard data.frames.

Value

The modified object.

Methods (by class)

- `fill_na(logical)`: Fill Missing Values for logical Objects
- `fill_na(integer)`: Fill Missing Values for integer Objects
- `fill_na(numeric)`: Fill Missing Values for numeric Objects
- `fill_na(character)`: Fill Missing Values for character Objects

See Also

Other fill: [fill_all\(\)](#)

Examples

```
# logical
fill_na(c(TRUE, NA))

# integer
fill_na(c(1L, NA), 0)
```

```
# numeric
fill_na(c(1, NA), Inf)

# character
fill_na(c("text", NA))
fill_na(matrix(c("text", NA)), value = Inf)
```

ilog*Inverse Log Transformation*

Description

Inverse log transforms a numeric atomic object.

Usage

```
ilog(x)
```

Arguments

x An object.

Details

A wrapper on [exp\(value\)](#).

Value

A numeric atomic object.

See Also

Other translations: [exp10\(\)](#), [exp2\(\)](#), [fabs\(\)](#), [ilog10\(\)](#), [ilog2\(\)](#), [ilogit\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [log10<-\(\)](#), [log2<-\(\)](#), [log<-\(\)](#), [logit\(\)](#), [logit<-\(\)](#), [phi\(\)](#), [pow\(\)](#), [step\(\)](#)

Examples

```
x <- 1
ilog(x)
```

ilog10	<i>Inverse Log Base 10 Transformation</i>
--------	---

Description

Inverse log transforms a numeric atomic object with base 10.

Usage

```
ilog10(x)
```

Arguments

x	An object.
---	------------

Details

A wrapper on [exp10\(value\)](#).

Value

A numeric atomic object.

See Also

Other translations: [exp10\(\)](#), [exp2\(\)](#), [fabs\(\)](#), [ilog\(\)](#), [ilog2\(\)](#), [ilogit\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [log10<-\(\)](#), [log2<-\(\)](#), [log<-\(\)](#), [logit\(\)](#), [logit<-\(\)](#), [phi\(\)](#), [pow\(\)](#), [step\(\)](#)

Examples

```
x <- c(2, 4.5)
ilog10(x)
```

ilog2	<i>Inverse Log Base 2 Transformation</i>
-------	--

Description

Inverse log transforms a numeric atomic object with base 2.

Usage

```
ilog2(x)
```

Arguments

x	An object.
---	------------

Details

A wrapper on `exp2(value)`.

Value

A numeric atomic object.

See Also

Other translations: `exp10()`, `exp2()`, `fabs()`, `ilog()`, `ilog10()`, `ilogit()`, `inv_logit()`, `invlogit()`, `log10<-()`, `log2<-()`, `log<-()`, `logit()`, `logit<-()`, `phi()`, `pow()`, `step()`

Examples

```
x <- c(2, 4.5)
ilog2(x)
```

ilogit

Inverse Logistic Transformation

Description

Inverse logistically transforms a numeric atomic object.

Usage

```
ilogit(x)
```

Arguments

`x` A numeric atomic object.

Details

A wrapper on `stats::plogis()`.

Value

A numeric atomic object.

See Also

Other translations: `exp10()`, `exp2()`, `fabs()`, `ilog()`, `ilog10()`, `ilog2()`, `inv_logit()`, `invlogit()`, `log10<-()`, `log2<-()`, `log<-()`, `logit()`, `logit<-()`, `phi()`, `pow()`, `step()`

Examples

```
ilogit(c(-1, 0, 5))
```

inv_logit	<i>Inverse Logistic Transformation</i>
-----------	--

Description

Inverse logistically transforms a numeric atomic object.

Usage

```
inv_logit(x)
```

Arguments

x A numeric atomic object.

Details

A wrapper on `stats::plogis()`.

Value

A numeric atomic object.

See Also

Other translations: `exp10()`, `exp2()`, `fabs()`, `ilog()`, `ilog10()`, `ilog2()`, `ilogit()`, `invlogit()`, `log10<-()`, `log2<-()`, `log<-()`, `logit()`, `logit<-()`, `phi()`, `pow()`, `step()`

Examples

```
inv_logit(c(-1, 0, 5))
```

inv_odds	<i>Inverse Odds</i>
----------	---------------------

Description

Calculates the probabilities for odds.

Usage

```
inv_odds(x)
```

Arguments

x A numeric object (vector, matrix or array) of odds.

Value

A numeric object of the the probabilities for each odd.

See Also

Other odds: [log_odds\(\)](#), [log_odds<-\(\)](#), [log_odds_ratio\(\)](#), [odds\(\)](#), [odds<-\(\)](#), [odds_ratio\(\)](#)

Examples

```
inv_odds(c(0, 1, 9, 9999))
```

 invlogit

Inverse Logistic Transformation

Description

Inverse logistically transforms a numeric atomic object.

Usage

```
invlogit(x)
```

Arguments

x A numeric atomic object.

Details

A wrapper on [stats::plogis\(\)](#).

Value

A numeric atomic object.

See Also

Other translations: [exp10\(\)](#), [exp2\(\)](#), [fabs\(\)](#), [ilog\(\)](#), [ilog10\(\)](#), [ilog2\(\)](#), [ilogit\(\)](#), [inv_logit\(\)](#), [log10<-\(\)](#), [log2<-\(\)](#), [log<-\(\)](#), [logit\(\)](#), [logit<-\(\)](#), [phi\(\)](#), [pow\(\)](#), [step\(\)](#)

Examples

```
invlogit(c(-1, 0, 5))
```

kurtosis	<i>Kurtosis</i>
----------	-----------------

Description

Kurtosis

Usage

```
kurtosis(x, na_rm = FALSE)
```

Arguments

x	A numeric object of MCMC values.
na_rm	A flag specifying whether to remove missing values.

Value

A number.

See Also

Other summary: [direction\(\)](#), [directional_information\(\)](#), [lower\(\)](#), [probability_direction\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr_mean\(\)](#), [xtr_median\(\)](#), [xtr_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

Examples

```
kurtosis(1:10)
```

log_lik_bern	<i>Bernoulli Log-Likelihood</i>
--------------	---------------------------------

Description

Bernoulli Log-Likelihood

Usage

```
log_lik_bern(x, prob = 0.5)
```

Arguments

x	A vector of 0s and 1s.
prob	A numeric vector of values between 0 and 1 of the probability of success.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_beta\(\)](#), [log_lik_beta_binom\(\)](#), [log_lik_binom\(\)](#), [log_lik_exp\(\)](#), [log_lik_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_multinom\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewlnorm\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#), [log_lik_unif\(\)](#)

Examples

```
log_lik_bern(c(TRUE, FALSE), 0.7)
```

log_lik_beta	<i>Beta Log-Likelihood</i>
--------------	----------------------------

Description

Beta Log-Likelihood

Usage

```
log_lik_beta(x, alpha = 1, beta = 1, tlower = 0, tupper = 1)
```

Arguments

- x A vector of values between 0s and 1s.
- alpha The first shape parameter of the beta distribution.
- beta The second shape parameter of the beta distribution.
- tlower A numeric vector of the lower truncation point.
- tupper A numeric vector of the upper truncation point.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta_binom\(\)](#), [log_lik_binom\(\)](#), [log_lik_exp\(\)](#), [log_lik_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_multinom\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewlnorm\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#), [log_lik_unif\(\)](#)

Examples

```
log_lik_beta(c(0, 0.5, 0.7, 1), 0.7)
```

log_lik_beta_binom	<i>Beta-Binomial Log-Likelihood</i>
--------------------	-------------------------------------

Description

This parameterization of the beta-binomial distribution uses an expected probability parameter, prob, and a dispersion parameter, theta. The parameters of the underlying beta mixture are $\alpha = (2 * \text{prob}) / \text{theta}$ and $\beta = (2 * (1 - \text{prob})) / \text{theta}$. This parameterization of theta is unconventional, but has useful properties when modelling. When $\text{theta} = 0$, the beta-binomial reverts to the binomial distribution. When $\text{theta} = 1$ and $\text{prob} = 0.5$, the parameters of the beta distribution become $\alpha = 1$ and $\beta = 1$, which correspond to a uniform distribution for the beta-binomial probability parameter.

Usage

```
log_lik_beta_binom(
  x,
  size = 1,
  prob = 0.5,
  theta = 0,
  tlower = 0,
  tupper = Inf,
  memoize = FALSE
)
```

Arguments

x	A non-negative whole numeric vector of values.
size	A non-negative whole numeric vector of the number of trials.
prob	A numeric vector of values between 0 and 1 of the probability of success.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
tlower	A numeric vector of the lower truncation point.
tupper	A numeric vector of the upper truncation point.
memoize	Whether or not to memoize the function.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta\(\)](#), [log_lik_binom\(\)](#), [log_lik_exp\(\)](#), [log_lik_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_multinom\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewlnorm\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#), [log_lik_unif\(\)](#)

Examples

```
log_lik_beta_binom(c(0, 1, 2), 3, 0.5, 0)
```

log_lik_binom

Binomial Log-Likelihood

Description

Binomial Log-Likelihood

Usage

```
log_lik_binom(x, size = 1, prob = 0.5, tlower = 0, tupper = Inf)
```

Arguments

x	A non-negative whole numeric vector of values.
size	A non-negative whole numeric vector of the number of trials.
prob	A numeric vector of values between 0 and 1 of the probability of success.
tlower	A numeric vector of the lower truncation point.
tupper	A numeric vector of the upper truncation point.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta\(\)](#), [log_lik_beta_binom\(\)](#), [log_lik_exp\(\)](#), [log_lik_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_multinom\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewlnorm\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#), [log_lik_unif\(\)](#)

Examples

```
log_lik_binom(c(0, 1, 2), 2, 0.3)
```

log_lik_exp	<i>Exponential Log-Likelihood</i>
-------------	-----------------------------------

Description

The exponential distribution is parameterized in terms of the rate.

Usage

```
log_lik_exp(x, rate = 1, tlower = 0, tupper = Inf)
```

Arguments

x	A non-negative numeric vector of values.
rate	A non-negative numeric vector of rate.
tlower	A numeric vector of the lower truncation point.
tupper	A numeric vector of the upper truncation point.

Value

A numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta\(\)](#), [log_lik_beta_binom\(\)](#), [log_lik_binom\(\)](#), [log_lik_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_multinom\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewlnorm\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#), [log_lik_unif\(\)](#)

Examples

```
log_lik_exp(c(0, 1, 2), 2)
```

log_lik_gamma	<i>Gamma Log-Likelihood</i>
---------------	-----------------------------

Description

Gamma Log-Likelihood

Usage

```
log_lik_gamma(x, shape = 1, rate = 1, tlower = 0, tupper = Inf)
```

Arguments

x	A numeric vector of values.
shape	A non-negative numeric vector of shape.
rate	A non-negative numeric vector of rate.
tlower	A numeric vector of the lower truncation point.
tupper	A numeric vector of the upper truncation point.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta\(\)](#), [log_lik_beta_binom\(\)](#), [log_lik_binom\(\)](#), [log_lik_exp\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_multinom\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewlnorm\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#), [log_lik_unif\(\)](#)

Examples

```
log_lik_gamma(c(0, 1, 2), 1, 2)
```

log_lik_gamma_pois	<i>Gamma-Poisson Log-Likelihood</i>
--------------------	-------------------------------------

Description

Gamma-Poisson Log-Likelihood

Usage

```
log_lik_gamma_pois(x, lambda = 1, theta = 0, tlower = 0, tupper = Inf)
```

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
tlower	A numeric vector of the lower truncation point.
tupper	A numeric vector of the upper truncation point.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta\(\)](#), [log_lik_beta_binom\(\)](#), [log_lik_binom\(\)](#), [log_lik_exp\(\)](#), [log_lik_gamma\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_multinom\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewlnorm\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#), [log_lik_unif\(\)](#)

Examples

```
log_lik_gamma_pois(c(0, 1, 2), 1, 1)
```

log_lik_gamma_pois_zi *Zero-Inflated Gamma-Poisson Log-Likelihood*

Description

Zero-Inflated Gamma-Poisson Log-Likelihood

Usage

```
log_lik_gamma_pois_zi(
  x,
  lambda = 1,
  theta = 0,
  prob = 0,
  tlower = 0,
  tupper = Inf
)
```

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
prob	A numeric vector of values between 0 and 1 of the probability of success.
tlower	A numeric vector of the lower truncation point.
tupper	A numeric vector of the upper truncation point.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta\(\)](#), [log_lik_beta_binom\(\)](#), [log_lik_binom\(\)](#), [log_lik_exp\(\)](#), [log_lik_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_multinom\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewlnorm\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#), [log_lik_unif\(\)](#)

Examples

```
log_lik_gamma_pois_zi(c(1, 3, 4), 3, 1, prob = 0.5)
```

log_lik_lnorm	<i>Log-Normal Log-Likelihood</i>
---------------	----------------------------------

Description

Log-Normal Log-Likelihood

Usage

```
log_lik_lnorm(x, meanlog = 0, sdlog = 1, tlower = 0, tupper = Inf)
```

Arguments

- x A numeric vector of values.
- meanlog A numeric vector of the means on the log scale.
- sdlog A non-negative numeric vector of the standard deviations on the log scale.
- tlower A numeric vector of the lower truncation point.
- tupper A numeric vector of the upper truncation point.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta\(\)](#), [log_lik_beta_binom\(\)](#), [log_lik_binom\(\)](#), [log_lik_exp\(\)](#), [log_lik_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_multinom\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewlnorm\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#), [log_lik_unif\(\)](#)

Examples

```
log_lik_lnorm(10, 0, 2)
```

log_lik_multinom	<i>Multinomial Log-Likelihood</i>
------------------	-----------------------------------

Description

Models the counts across two or more mutually exclusive categories from a fixed number of trials, in *long* format: one row per category per trial, with group identifying which rows belong to the same trial. All rows sharing a group must have the same size, and their prob values must sum to 1.

Usage

```
log_lik_multinom(x, size = 1, prob, group)
```

Arguments

x	A non-negative whole numeric vector of the category counts.
size	A non-negative whole numeric vector of the number of trials.
prob	A numeric vector of the probability of the category. Must sum to 1 across the rows sharing the same group. NA in size or prob for any row of a trial makes the log-likelihood NA for every row of that trial, since a trial's categories are scored jointly.
group	A vector identifying which rows belong to the same multinomial trial (whose x values sum to size and prob values sum to 1). Every group must have at least 2 rows and the same number of rows as the rest of the data (a fixed set of categories, as in multinomial logistic regression), and must not contain NA.

Details

A trial's log-likelihood doesn't split evenly across its rows, since the multinomial coefficient belongs to the whole trial. `log_lik_multinom()` uses the multinomial-as-independent-Poissons identity: each row's value is the Poisson log-likelihood of x given $\mu = \text{size} * \text{prob}$, minus an even share of the trial's normalizing constant, so summing over a group recovers the trial's exact multinomial log-likelihood.

The even split of the normalizing constant is a bookkeeping convention: only the sum over a group is exact, so an individual row's value is not a valid pointwise log-likelihood. Cross-validation and information criteria should therefore treat the trial, not the row, as the leave-one-out unit, which is achieved by summing the values within each group to get the trial's exact log-likelihood.

Value

An numeric vector of the corresponding log-likelihoods, one value per row of x .

References

- Baker, S.G. 1994. The multinomial-Poisson transformation. *The Statistician* 43(4): 495-504. doi:10.2307/2348134.
- McCullagh, P., and Nelder, J.A. 1989. *Generalized Linear Models*. 2nd edition. Chapman and Hall, London.
- Forster, J.J. 2010. Bayesian inference for Poisson and multinomial log-linear models. *Statistical Methodology* 7(3): 210-224. doi:10.1016/j.stamet.2009.12.004.
- Vehtari, A., Gelman, A., and Gabry, J. 2017. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing* 27(5): 1413-1432. doi:10.1007/s11222-01696963.

See Also

Other log_lik_dist: `log_lik_bern()`, `log_lik_beta()`, `log_lik_beta_binom()`, `log_lik_binom()`, `log_lik_exp()`, `log_lik_gamma()`, `log_lik_gamma_pois()`, `log_lik_gamma_pois_zi()`, `log_lik_lnorm()`, `log_lik_neg_binom()`, `log_lik_norm()`, `log_lik_pois()`, `log_lik_pois_zi()`, `log_lik_skewlnorm()`, `log_lik_skewnorm()`, `log_lik_student()`, `log_lik_unif()`

Examples

```
log_lik_multinom(c(1, 3, 6), size = 10, prob = c(0.2, 0.3, 0.5), group = c(1, 1, 1))
```

log_lik_neg_binom	<i>Negative Binomial Log-Likelihood</i>
-------------------	---

Description

Negative Binomial Log-Likelihood

Usage

```
log_lik_neg_binom(x, lambda = 1, theta = 0, tlower = 0, tupper = Inf)
```

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
tlower	A numeric vector of the lower truncation point.
tupper	A numeric vector of the upper truncation point.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta\(\)](#), [log_lik_beta_binom\(\)](#), [log_lik_binom\(\)](#), [log_lik_exp\(\)](#), [log_lik_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_multinom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewlnorm\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#), [log_lik_unif\(\)](#)

Examples

```
log_lik_neg_binom(c(0, 1, 2), 2, 1)
```

log_lik_norm	<i>Normal Log-Likelihood</i>
--------------	------------------------------

Description

Normal Log-Likelihood

Usage

```
log_lik_norm(x, mean = 0, sd = 1, tlower = -Inf, tupper = Inf)
```

Arguments

x	A numeric vector of values.
mean	A numeric vector of the means.
sd	A non-negative numeric vector of the standard deviations.
tlower	A numeric vector of the lower truncation point.
tupper	A numeric vector of the upper truncation point.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta\(\)](#), [log_lik_beta_binom\(\)](#), [log_lik_binom\(\)](#), [log_lik_exp\(\)](#), [log_lik_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_multinom\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewlnorm\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#), [log_lik_unif\(\)](#)

Examples

```
log_lik_norm(c(-2:2))
```

log_lik_pois	<i>Poisson Log-Likelihood</i>
--------------	-------------------------------

Description

Poisson Log-Likelihood

Usage

```
log_lik_pois(x, lambda = 1, tlower = 0, tupper = Inf)
```

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
tlower	A numeric vector of the lower truncation point.
tupper	A numeric vector of the upper truncation point.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta\(\)](#), [log_lik_beta_binom\(\)](#), [log_lik_binom\(\)](#), [log_lik_exp\(\)](#), [log_lik_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_multinom\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewlnorm\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#), [log_lik_unif\(\)](#)

Examples

```
log_lik_pois(c(1, 3, 4), 3)
```

log_lik_pois_zi	<i>Zero-Inflated Poisson Log-Likelihood</i>
-----------------	---

Description

Zero-Inflated Poisson Log-Likelihood

Usage

```
log_lik_pois_zi(x, lambda = 1, prob = 0, tlower = 0, tupper = Inf)
```

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
prob	A numeric vector of values between 0 and 1 of the probability of success.
tlower	A numeric vector of the lower truncation point.
tupper	A numeric vector of the upper truncation point.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta\(\)](#), [log_lik_beta_binom\(\)](#), [log_lik_binom\(\)](#), [log_lik_exp\(\)](#), [log_lik_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_multinom\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_skewlnorm\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#), [log_lik_unif\(\)](#)

Examples

```
log_lik_pois_zi(c(1, 3, 4), 3, prob = 0.5)
```

log_lik_skewlnorm	<i>Skew-Lognormal Log-Likelihood</i>
-------------------	--------------------------------------

Description

Skew-Lognormal Log-Likelihood

Usage

```
log_lik_skewlnorm(
  x,
  locationlog = 0,
  scalelog = 1,
  shapelog = 0,
  tlower = 0,
  tupper = Inf,
  ...,
  meanlog,
  sdlog,
  shape
)
```

Arguments

x	A numeric vector of values.
locationlog	A numeric vector of location parameters of $\log(x)$.
scalelog	A non-negative numeric vector of scale parameters of $\log(x)$.
shapelog	A numeric vector of shape parameters of $\log(x)$. Negative values result in leftward skew, while positive values result in rightward skew.
tlower	A numeric vector of the lower truncation point.
tupper	A numeric vector of the upper truncation point.
...	Unused.
meanlog	[Deprecated] A numeric vector of location parameters of $\log(x)$. Described as "a numeric vector of the means on the log scale" prior to v. 0.10.1. Will be removed in a future version.
sdlog	[Deprecated] A non-negative numeric vector of scale parameters of $\log(x)$. Described as "a non-negative numeric vector of the standard deviations on the log scale" prior to v. 0.10.1. Will be removed in a future version.
shape	[Deprecated] A numeric vector of shape parameters of $\log(x)$. Will be removed in a future version.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta\(\)](#), [log_lik_beta_binom\(\)](#), [log_lik_binom\(\)](#), [log_lik_exp\(\)](#), [log_lik_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_multinom\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#), [log_lik_unif\(\)](#)

Examples

```
log_lik_skewlnorm(1:5)
log_lik_skewlnorm(1:5, shapelog = -2)
log_lik_skewlnorm(1:5, shapelog = 2)
```

log_lik_skewnorm	<i>Skew Normal Log-Likelihood</i>
------------------	-----------------------------------

Description

Skew Normal Log-Likelihood

Usage

```
log_lik_skewnorm(
  x,
  location = 0,
  scale = 1,
  shape = 0,
  tlower = -Inf,
  tupper = Inf,
  ...,
  mean,
  sd
)
```

Arguments

x	A numeric vector of values.
location	A numeric vector of the location parameter.
scale	A non-negative numeric vector of the scale parameter.
shape	A numeric vector of shape. Negative values result in leftward skew, while positive values result in rightward skew.
tlower	A numeric vector of the lower truncation point.
tupper	A numeric vector of the upper truncation point.
...	Unused.
mean	[Deprecated] A numeric vector of the location parameter. Described as "a numeric vector of the means" prior to v. 0.10.1. Will be removed in a future version.
sd	[Deprecated] A non-negative numeric vector of the scale parameter. Described as "a non-negative numeric vector of the standard deviations" prior to v. 0.10.1. Will be removed in a future version.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta\(\)](#), [log_lik_beta_binom\(\)](#), [log_lik_binom\(\)](#), [log_lik_exp\(\)](#), [log_lik_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_multinom\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewlnorm\(\)](#), [log_lik_student\(\)](#), [log_lik_unif\(\)](#)

Examples

```
log_lik_skewnorm(c(-2:2))
log_lik_skewnorm(c(-2:2), shape = -2)
log_lik_skewnorm(c(-2:2), shape = 2)
```

log_lik_student	<i>Student's t Log-Likelihood</i>
-----------------	-----------------------------------

Description

Student's t Log-Likelihood

Usage

```
log_lik_student(x, mean = 0, sd = 1, theta = 0, tlower = -Inf, tupper = Inf)
```

Arguments

x	A numeric vector of values.
mean	A numeric vector of the means.
sd	A non-negative numeric vector of the standard deviations.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
tlower	A numeric vector of the lower truncation point.
tupper	A numeric vector of the upper truncation point.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta\(\)](#), [log_lik_beta_binom\(\)](#), [log_lik_binom\(\)](#), [log_lik_exp\(\)](#), [log_lik_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_multinom\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewlnorm\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_unif\(\)](#)

Examples

```
log_lik_student(c(1, 3.5, 4), mean = 1, sd = 2, theta = 1 / 3)
```

log_lik_unif	<i>Uniform Log-Likelihood</i>
--------------	-------------------------------

Description

Uniform Log-Likelihood

Usage

```
log_lik_unif(x, min = 0, max = 1)
```

Arguments

x	A numeric vector of values.
min	A numeric vector of the minimums.
max	A numeric vector of the maximums.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta\(\)](#), [log_lik_beta_binom\(\)](#), [log_lik_binom\(\)](#), [log_lik_exp\(\)](#), [log_lik_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_multinom\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewlnorm\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#)

Examples

```
log_lik_unif(c(0, 0.7, 1))
```

log_odds	<i>Log Odds</i>
----------	-----------------

Description

Calculates the log odds for probabilities.

Usage

```
log_odds(x)
```

Arguments

x	A numeric object (vector, matrix or array) of probabilities.
---	--

Value

A numeric object of the the log odds for each probability.

See Also

Other odds: [inv_odds\(\)](#), [log_odds<-\(\(\)\)](#), [log_odds_ratio\(\)](#), [odds\(\)](#), [odds<-\(\(\)\)](#), [odds_ratio\(\)](#)

Examples

```
log_odds(c(0, 0.5, 0.9, 1))
```

log_odds_ratio	<i>Log-Odds Ratio</i>
----------------	-----------------------

Description

Calculates the log odds ratio for two probabilities.

Usage

```
log_odds_ratio(x, x2)
```

Arguments

x	A numeric object (vector, matrix or array) of probabilities.
x2	A second numeric object of probabilities.

Value

A numeric object of the log odds ratios.

See Also

Other odds: [inv_odds\(\)](#), [log_odds\(\)](#), [log_odds<-\(\(\)\)](#), [odds\(\)](#), [odds<-\(\(\)\)](#), [odds_ratio\(\)](#)

Examples

```
log_odds_ratio(0.5, 0.75)
```

log_odds_ratio2	<i>Log Odds Ratio2</i>
-----------------	------------------------

Description

Calculates the log odds ratio for a vector of two probabilities.

Usage

```
log_odds_ratio2(x)
```

Arguments

x A numeric vector of length 2.

Value

A number.

See Also

Other odds fun2: [odds_ratio2\(\)](#)

Examples

```
log_odds_ratio2(c(0.5, 0.9))
log_odds_ratio2(c(0.9, 0.5))
```

log_odds<-	<i>Inverse Log Odds Transformation</i>
------------	--

Description

Replaces an object with the inverse log odds of value.

Usage

```
log_odds(x) <- value
```

Arguments

x An existing R object.
value A numeric atomic object.

Value

Called for the side effect of updating x.

See Also

Other odds: [inv_odds\(\)](#), [log_odds\(\)](#), [log_odds_ratio\(\)](#), [odds\(\)](#), [odds<-\(\)](#), [odds_ratio\(\)](#)

Examples

```
x <- NULL
log_odds(x) <- 0.5
x
```

log<-

Log Transformation

Description

Replaces a object with the exponent of value.

Usage

```
log(x) <- value
```

Arguments

x	An object.
value	A numeric atomic object.

Details

A wrapper on [exp\(value\)](#).

Value

Called for the side effect of updating x.

See Also

Other translations: [exp10\(\)](#), [exp2\(\)](#), [fabs\(\)](#), [ilog\(\)](#), [ilog10\(\)](#), [ilog2\(\)](#), [ilogit\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [log10<-\(\)](#), [log2<-\(\)](#), [logit\(\)](#), [logit<-\(\)](#), [phi\(\)](#), [pow\(\)](#), [step\(\)](#)

Examples

```
x <- NULL
log(x) <- 0.5
x
```

`log10<-`*Log Base 10 Transformation*

Description

Replaces a object with the base 10 exponent of value.

Usage

```
log10(x) <- value
```

Arguments

x	An object.
value	A numeric atomic object.

Details

A wrapper on [exp10](#)(value).

Value

Called for the side effect of updating x.

See Also

Other translations: [exp10\(\)](#), [exp2\(\)](#), [fabs\(\)](#), [ilog\(\)](#), [ilog10\(\)](#), [ilog2\(\)](#), [ilogit\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [log2<-\(\)](#), [log<-\(\)](#), [logit\(\)](#), [logit<-\(\)](#), [phi\(\)](#), [pow\(\)](#), [step\(\)](#)

Examples

```
x <- NULL
log10(x) <- c(0.5, 5)
x
```

`log2<-`*Log Base 2 Transformation*

Description

Replaces a object with the base 2 exponent of value.

Usage

```
log2(x) <- value
```

Arguments

`x` An object.
`value` A numeric atomic object.

Details

A wrapper on `exp2(value)`.

Value

Called for the side effect of updating `x`.

See Also

Other translations: `exp10()`, `exp2()`, `fabs()`, `ilog()`, `ilog10()`, `ilog2()`, `ilogit()`, `inv_logit()`, `invlogit()`, `log10<-()`, `log<-()`, `logit()`, `logit<-()`, `phi()`, `pow()`, `step()`

Examples

```
x <- NULL
log2(x) <- c(0.5, 5)
x
```

logit

Logistic Transformation

Description

Logistic transforms a numeric atomic object.

Usage

```
logit(x)
```

Arguments

`x` A numeric atomic object.

Details

A wrapper on `stats::qlogis()`.

Value

The logistically transformed numeric atomic object.

See Also

Other translations: [exp10\(\)](#), [exp2\(\)](#), [fabs\(\)](#), [ilog\(\)](#), [ilog10\(\)](#), [ilog2\(\)](#), [ilogit\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [log10<-\(\)](#), [log2<-\(\)](#), [log<-\(\)](#), [logit<-\(\)](#), [phi\(\)](#), [pow\(\)](#), [step\(\)](#)

Examples

```
logit(c(0.25, 0.5, 0.75))
```

logit<-

Logistic Transformation

Description

Logistic Transformation

Usage

```
logit(x) <- value
```

Arguments

x	An existing object.
value	A numeric atomic object of the value to inverse logistically transform.

Details

A wrapper on `stats::plogis(value)`.

Value

Called for the side effect of updating x.

See Also

Other translations: [exp10\(\)](#), [exp2\(\)](#), [fabs\(\)](#), [ilog\(\)](#), [ilog10\(\)](#), [ilog2\(\)](#), [ilogit\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [log10<-\(\)](#), [log2<-\(\)](#), [log<-\(\)](#), [logit\(\)](#), [phi\(\)](#), [pow\(\)](#), [step\(\)](#)

Examples

```
x <- 1
logit(x) <- 0.5
x
```

lower	<i>Lower Credible Limit</i>
-------	-----------------------------

Description

Calculates the quantile-based lower credible limit.

Usage

```
lower(x, conf_level = 0.95, na_rm = FALSE)
```

Arguments

x	A numeric vector of MCMC values.
conf_level	A numeric scalar between 0 and 1 specifying the confidence level.
na_rm	A flag specifying whether to remove missing values.

Details

By default it returns the 95% credible limit which corresponds to the 2.5% quantile.

Value

A number.

See Also

Other summary: [direction\(\)](#), [directional_information\(\)](#), [kurtosis\(\)](#), [probability_direction\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr_mean\(\)](#), [xtr_median\(\)](#), [xtr_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

Examples

```
lower(as.numeric(0:100))
```

numericise	<i>Numericise (or Numericize)</i>
------------	-----------------------------------

Description

Coerce an R object to a numeric atomic object.

Usage

```
numericise(x, ...)  
  
numericize(x, ...)  
  
## S3 method for class 'logical'  
numericise(x, ...)  
  
## S3 method for class 'integer'  
numericise(x, ...)  
  
## S3 method for class 'double'  
numericise(x, ...)  
  
## S3 method for class 'factor'  
numericise(x, ...)  
  
## S3 method for class 'Date'  
numericise(x, ...)  
  
## S3 method for class 'POSIXct'  
numericise(x, ...)  
  
## S3 method for class 'hms'  
numericise(x, ...)  
  
## S3 method for class 'matrix'  
numericise(x, ...)  
  
## S3 method for class 'array'  
numericise(x, ...)  
  
## S3 method for class 'data.frame'  
numericise(x, ...)
```

Arguments

x	An object.
...	Other arguments passed to methods.

Details

numericize() is an alias for numericise. If you want to implement a method for a class "foo", implement numericise.foo().

Value

A numeric atomic object.

Methods (by class)

- `numericise(logical)`: Numericise a logical Object
- `numericise(integer)`: Numericise an integer Object
- `numericise(double)`: Numericise an double Object
- `numericise(factor)`: Numericise a factor
- `numericise(Date)`: Numericise a Date vector
- `numericise(POSIXct)`: Numericise a POSIXct vector
- `numericise(hms)`: Numericise a hms vector
- `numericise(matrix)`: Numericise a matrix
- `numericise(array)`: Numericise an array
- `numericise(data.frame)`: Numericise a data.frame

Examples

```
# logical
numericise(TRUE)
numericise(matrix(c(TRUE, FALSE), nrow = 2))

# integer
numericise(2L)

# double
numericise(c(1, 3))

# factor
numericise(factor(c("c", "a"))))

# Date
numericise(as.Date("1972-01-01"))

# POSIXct
numericise(as.POSIXct("1972-01-01", tz = "UTC"))

# hms
numericise(hms::as_hms("00:01:03"))

# matrix
numericise(matrix(TRUE))

# array
numericise(array(TRUE))

# data.frame
numericise(data.frame(
  logical = c(TRUE, FALSE, NA),
  integer = 1:3,
```

```
numeric = c(4, 10, NA),  
factor = as.factor(c("c", "A", "green"))  
)
```

odds	<i>Odds</i>
------	-------------

Description

Calculates the odds for probabilities.

Usage

```
odds(x)
```

Arguments

x A numeric object (vector, matrix or array) of probabilities.

Value

A numeric object of the the odds for each probability.

See Also

Other odds: [inv_odds\(\)](#), [log_odds\(\)](#), [log_odds<-\(\(\)\)](#), [log_odds_ratio\(\)](#), [odds<-\(\(\)\)](#), [odds_ratio\(\)](#)

Examples

```
odds(c(0, 0.5, 0.9, 1))
```

odds_ratio	<i>Odds Ratio</i>
------------	-------------------

Description

Calculates the odds ratio for two probabilities.

Usage

```
odds_ratio(x, x2)
```

Arguments

x A numeric object (vector, matrix or array) of probabilities.

x2 A second numeric object of probabilities.

Value

A numeric object of the odds ratios.

See Also

Other odds: [inv_odds\(\)](#), [log_odds\(\)](#), [log_odds<-\(\(\)\)](#), [log_odds_ratio\(\)](#), [odds\(\)](#), [odds<-\(\(\)\)](#)

Examples

```
odds_ratio(0.5, 0.75)
```

odds_ratio2	<i>Odds Ratio2</i>
-------------	--------------------

Description

Calculates the odds ratio for a vector of two probabilities.

Usage

```
odds_ratio2(x)
```

Arguments

x A numeric vector of length 2.

Value

A number.

See Also

Other odds fun2: [log_odds_ratio2\(\)](#)

Examples

```
odds_ratio2(c(0.5, 0.9))
odds_ratio2(c(0.9, 0.5))
```

odds<-	<i>Inverse Odds Transformation</i>
--------	------------------------------------

Description

Replaces an object with the inverse odds of value.

Usage

```
odds(x) <- value
```

Arguments

x	An existing R object.
value	A numeric atomic object.

Value

Called for the side effect of updating x.

See Also

Other odds: [inv_odds\(\)](#), [log_odds\(\)](#), [log_odds<-\(\)](#), [log_odds_ratio\(\)](#), [odds\(\)](#), [odds_ratio\(\)](#)

Examples

```
x <- NULL
odds(x) <- 0.5
x
```

par_pattern	<i>Parameter Pattern</i>
-------------	--------------------------

Description

Parameter Pattern

Usage

```
par_pattern()
```

Value

A string of the regular expression for a parameter name.

Examples

```
par_pattern()
```

pextreme

Extreme Probability

Description

Calculates the probability that a cumulative distribution function probability is at least that extreme.
[Deprecated]

Usage

```
pextreme(x)
```

Arguments

x A numeric vector of values between 0 and 1.

Value

A numeric vector of values between 0 and 1.

See Also

Other residuals: [sextreme\(\)](#)

Examples

```
pextreme(seq(0, 1, by = 0.1))
```

phi

Phi

Description

The standard normal cumulative density function.

Usage

```
phi(x)
```

Arguments

x A numeric atomic object.

Details

A wrapper on [stats::pnorm\(\)](#).

Value

A numeric atomic object.

See Also

Other translations: [exp10\(\)](#), [exp2\(\)](#), [fabs\(\)](#), [ilog\(\)](#), [ilog10\(\)](#), [ilog2\(\)](#), [ilogit\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [log10<-\(\)](#), [log2<-\(\)](#), [log<-\(\)](#), [logit\(\)](#), [logit<-\(\)](#), [pow\(\)](#), [step\(\)](#)

Examples

```
phi(0:2)
```

pow	<i>Power</i>
-----	--------------

Description

R equivalent to the power function.

Usage

```
pow(x, n)
```

Arguments

x	A numeric atomic object of the base.
n	A numeric atomic object of the exponent.

Details

Wrapper on x^n .

Value

A numeric atomic object of x raised to n .

See Also

Other translations: [exp10\(\)](#), [exp2\(\)](#), [fabs\(\)](#), [ilog\(\)](#), [ilog10\(\)](#), [ilog2\(\)](#), [ilogit\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [log10<-\(\)](#), [log2<-\(\)](#), [log<-\(\)](#), [logit\(\)](#), [logit<-\(\)](#), [phi\(\)](#), [step\(\)](#)

Examples

```
pow(10, 2)
```

prob_bern	<i>Bernoulli Cumulative Distribution Function</i>
-----------	---

Description

Bernoulli Cumulative Distribution Function

Usage

```
prob_bern(x, prob = 0.5)
```

Arguments

x	A numeric vector of quantiles.
prob	A numeric vector of values between 0 and 1 of the probability of success.

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_beta\(\)](#), [prob_beta_binom\(\)](#), [prob_binom\(\)](#), [prob_exp\(\)](#), [prob_gamma\(\)](#), [prob_gamma_pois\(\)](#), [prob_gamma_pois_zi\(\)](#), [prob_lnorm\(\)](#), [prob_neg_binom\(\)](#), [prob_norm\(\)](#), [prob_pois\(\)](#), [prob_pois_zi\(\)](#), [prob_skewlnorm\(\)](#), [prob_skewnorm\(\)](#), [prob_student\(\)](#), [prob_unif\(\)](#)

Examples

```
prob_bern(c(TRUE, FALSE), 0.7)
```

prob_beta	<i>Beta Cumulative Distribution Function</i>
-----------	--

Description

Beta Cumulative Distribution Function

Usage

```
prob_beta(x, alpha = 1, beta = 1)
```

Arguments

x	A numeric vector of quantiles.
alpha	The first shape parameter of the beta distribution.
beta	The second shape parameter of the beta distribution.

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_bern\(\)](#), [prob_beta_binom\(\)](#), [prob_binom\(\)](#), [prob_exp\(\)](#), [prob_gamma\(\)](#), [prob_gamma_pois\(\)](#), [prob_gamma_pois_zi\(\)](#), [prob_lnorm\(\)](#), [prob_neg_binom\(\)](#), [prob_norm\(\)](#), [prob_pois\(\)](#), [prob_pois_zi\(\)](#), [prob_skewlnorm\(\)](#), [prob_skewnorm\(\)](#), [prob_student\(\)](#), [prob_unif\(\)](#)

Examples

```
prob_beta(c(0, 0.5, 1), 2, 3)
```

 prob_beta_binom

Beta-Binomial Cumulative Distribution Function

Description

This parameterization of the beta-binomial distribution uses an expected probability parameter, prob, and a dispersion parameter, theta. The parameters of the underlying beta mixture are $\alpha = (2 * \text{prob}) / \text{theta}$ and $\beta = (2 * (1 - \text{prob})) / \text{theta}$. This parameterization of theta is unconventional, but has useful properties when modelling. When $\text{theta} = 0$, the beta-binomial reverts to the binomial distribution. When $\text{theta} = 1$ and $\text{prob} = 0.5$, the parameters of the beta distribution become $\alpha = 1$ and $\beta = 1$, which correspond to a uniform distribution for the beta-binomial probability parameter.

Usage

```
prob_beta_binom(x, size = 1, prob = 0.5, theta = 0)
```

Arguments

x	A numeric vector of quantiles.
size	A non-negative whole numeric vector of the number of trials.
prob	A numeric vector of values between 0 and 1 of the probability of success.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_bern\(\)](#), [prob_beta\(\)](#), [prob_binom\(\)](#), [prob_exp\(\)](#), [prob_gamma\(\)](#), [prob_gamma_pois\(\)](#), [prob_gamma_pois_zi\(\)](#), [prob_lnorm\(\)](#), [prob_neg_binom\(\)](#), [prob_norm\(\)](#), [prob_pois\(\)](#), [prob_pois_zi\(\)](#), [prob_skewlnorm\(\)](#), [prob_skewnorm\(\)](#), [prob_student\(\)](#), [prob_unif\(\)](#)

Examples

```
prob_beta_binom(c(0, 1, 2), 3, 0.5, 0)
```

prob_binom

Binomial Cumulative Distribution Function

Description

Binomial Cumulative Distribution Function

Usage

```
prob_binom(x, size = 1, prob = 0.5)
```

Arguments

x	A numeric vector of quantiles.
size	A non-negative whole numeric vector of the number of trials.
prob	A numeric vector of values between 0 and 1 of the probability of success.

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_bern\(\)](#), [prob_beta\(\)](#), [prob_beta_binom\(\)](#), [prob_exp\(\)](#), [prob_gamma\(\)](#), [prob_gamma_pois\(\)](#), [prob_gamma_pois_zi\(\)](#), [prob_lnorm\(\)](#), [prob_neg_binom\(\)](#), [prob_norm\(\)](#), [prob_pois\(\)](#), [prob_pois_zi\(\)](#), [prob_skewlnorm\(\)](#), [prob_skewnorm\(\)](#), [prob_student\(\)](#), [prob_unif\(\)](#)

Examples

```
prob_binom(c(0, 1, 2), 2, 0.3)
```

prob_exp	<i>Exponential Cumulative Distribution Function</i>
----------	---

Description

Exponential Cumulative Distribution Function

Usage

```
prob_exp(x, rate = 1)
```

Arguments

x	A numeric vector of quantiles.
rate	A non-negative numeric vector of rate.

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_bern\(\)](#), [prob_beta\(\)](#), [prob_beta_binom\(\)](#), [prob_binom\(\)](#), [prob_gamma\(\)](#), [prob_gamma_pois\(\)](#), [prob_gamma_pois_zi\(\)](#), [prob_lnorm\(\)](#), [prob_neg_binom\(\)](#), [prob_norm\(\)](#), [prob_pois\(\)](#), [prob_pois_zi\(\)](#), [prob_skewlnorm\(\)](#), [prob_skewnorm\(\)](#), [prob_student\(\)](#), [prob_unif\(\)](#)

Examples

```
prob_exp(c(0, 1, 2), 2)
```

prob_gamma	<i>Gamma Cumulative Distribution Function</i>
------------	---

Description

Gamma Cumulative Distribution Function

Usage

```
prob_gamma(x, shape = 1, rate = 1)
```

Arguments

x	A numeric vector of quantiles.
shape	A non-negative numeric vector of shape.
rate	A non-negative numeric vector of rate.

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_bern\(\)](#), [prob_beta\(\)](#), [prob_beta_binom\(\)](#), [prob_binom\(\)](#), [prob_exp\(\)](#), [prob_gamma_pois\(\)](#), [prob_gamma_pois_zi\(\)](#), [prob_lnorm\(\)](#), [prob_neg_binom\(\)](#), [prob_norm\(\)](#), [prob_pois\(\)](#), [prob_pois_zi\(\)](#), [prob_skewlnorm\(\)](#), [prob_skewnorm\(\)](#), [prob_student\(\)](#), [prob_unif\(\)](#)

Examples

```
prob_gamma(c(0, 1, 2), 1, 2)
```

prob_gamma_pois	<i>Gamma-Poisson Cumulative Distribution Function</i>
-----------------	---

Description

Gamma-Poisson Cumulative Distribution Function

Usage

```
prob_gamma_pois(x, lambda = 1, theta = 0)
```

Arguments

- x A numeric vector of quantiles.
- lambda A non-negative numeric vector of means.
- theta A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_bern\(\)](#), [prob_beta\(\)](#), [prob_beta_binom\(\)](#), [prob_binom\(\)](#), [prob_exp\(\)](#), [prob_gamma\(\)](#), [prob_gamma_pois_zi\(\)](#), [prob_lnorm\(\)](#), [prob_neg_binom\(\)](#), [prob_norm\(\)](#), [prob_pois\(\)](#), [prob_pois_zi\(\)](#), [prob_skewlnorm\(\)](#), [prob_skewnorm\(\)](#), [prob_student\(\)](#), [prob_unif\(\)](#)

Examples

```
prob_gamma_pois(c(0, 1, 2), 1, 1)
```

prob_gamma_pois_zi	<i>Zero-Inflated Gamma-Poisson Cumulative Distribution Function</i>
--------------------	---

Description

Zero-Inflated Gamma-Poisson Cumulative Distribution Function

Usage

```
prob_gamma_pois_zi(x, lambda = 1, theta = 0, prob = 0)
```

Arguments

x	A numeric vector of quantiles.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
prob	A numeric vector of values between 0 and 1 of the probability of success.

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_bern\(\)](#), [prob_beta\(\)](#), [prob_beta_binom\(\)](#), [prob_binom\(\)](#), [prob_exp\(\)](#), [prob_gamma\(\)](#), [prob_gamma_pois\(\)](#), [prob_lnorm\(\)](#), [prob_neg_binom\(\)](#), [prob_norm\(\)](#), [prob_pois\(\)](#), [prob_pois_zi\(\)](#), [prob_skewlnorm\(\)](#), [prob_skewnorm\(\)](#), [prob_student\(\)](#), [prob_unif\(\)](#)

Examples

```
prob_gamma_pois_zi(c(1, 3, 4), 3, 1, prob = 0.5)
```

prob_lnorm	<i>Log-Normal Cumulative Distribution Function</i>
------------	--

Description

Log-Normal Cumulative Distribution Function

Usage

```
prob_lnorm(x, meanlog = 0, sdlog = 1)
```

Arguments

x	A numeric vector of quantiles.
meanlog	A numeric vector of the means on the log scale.
sdlog	A non-negative numeric vector of the standard deviations on the log scale.

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_bern\(\)](#), [prob_beta\(\)](#), [prob_beta_binom\(\)](#), [prob_binom\(\)](#), [prob_exp\(\)](#), [prob_gamma\(\)](#), [prob_gamma_pois\(\)](#), [prob_gamma_pois_zi\(\)](#), [prob_neg_binom\(\)](#), [prob_norm\(\)](#), [prob_pois\(\)](#), [prob_pois_zi\(\)](#), [prob_skewlnorm\(\)](#), [prob_skewnorm\(\)](#), [prob_student\(\)](#), [prob_unif\(\)](#)

Examples

```
prob_lnorm(10, 0, 2)
```

prob_neg_binom	<i>Negative Binomial Cumulative Distribution Function</i>
----------------	---

Description

Negative Binomial Cumulative Distribution Function

Usage

```
prob_neg_binom(x, lambda = 1, theta = 0)
```

Arguments

x	A numeric vector of quantiles.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_bern\(\)](#), [prob_beta\(\)](#), [prob_beta_binom\(\)](#), [prob_binom\(\)](#), [prob_exp\(\)](#), [prob_gamma\(\)](#), [prob_gamma_pois\(\)](#), [prob_gamma_pois_zi\(\)](#), [prob_lnorm\(\)](#), [prob_norm\(\)](#), [prob_pois\(\)](#), [prob_pois_zi\(\)](#), [prob_skewlnorm\(\)](#), [prob_skewnorm\(\)](#), [prob_student\(\)](#), [prob_unif\(\)](#)

Examples

```
prob_neg_binom(c(0, 1, 2), 2, 1)
```

prob_norm

Normal Cumulative Distribution Function

Description

Normal Cumulative Distribution Function

Usage

```
prob_norm(x, mean = 0, sd = 1)
```

Arguments

x	A numeric vector of quantiles.
mean	A numeric vector of the means.
sd	A non-negative numeric vector of the standard deviations.

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_bern\(\)](#), [prob_beta\(\)](#), [prob_beta_binom\(\)](#), [prob_binom\(\)](#), [prob_exp\(\)](#), [prob_gamma\(\)](#), [prob_gamma_pois\(\)](#), [prob_gamma_pois_zi\(\)](#), [prob_lnorm\(\)](#), [prob_neg_binom\(\)](#), [prob_pois\(\)](#), [prob_pois_zi\(\)](#), [prob_skewlnorm\(\)](#), [prob_skewnorm\(\)](#), [prob_student\(\)](#), [prob_unif\(\)](#)

Examples

```
prob_norm(c(-2:2))
```

prob_pois	<i>Poisson Cumulative Distribution Function</i>
-----------	---

Description

Poisson Cumulative Distribution Function

Usage

```
prob_pois(x, lambda = 1)
```

Arguments

x	A numeric vector of quantiles.
lambda	A non-negative numeric vector of means.

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_bern\(\)](#), [prob_beta\(\)](#), [prob_beta_binom\(\)](#), [prob_binom\(\)](#), [prob_exp\(\)](#), [prob_gamma\(\)](#), [prob_gamma_pois\(\)](#), [prob_gamma_pois_zi\(\)](#), [prob_lnorm\(\)](#), [prob_neg_binom\(\)](#), [prob_norm\(\)](#), [prob_pois_zi\(\)](#), [prob_skewlnorm\(\)](#), [prob_skewnorm\(\)](#), [prob_student\(\)](#), [prob_unif\(\)](#)

Examples

```
prob_pois(c(1, 3, 4), 3)
```

prob_pois_zi	<i>Zero-Inflated Poisson Cumulative Distribution Function</i>
--------------	---

Description

Zero-Inflated Poisson Cumulative Distribution Function

Usage

```
prob_pois_zi(x, lambda = 1, prob = 0)
```

Arguments

x	A numeric vector of quantiles.
lambda	A non-negative numeric vector of means.
prob	A numeric vector of values between 0 and 1 of the probability of success.

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_bern\(\)](#), [prob_beta\(\)](#), [prob_beta_binom\(\)](#), [prob_binom\(\)](#), [prob_exp\(\)](#), [prob_gamma\(\)](#), [prob_gamma_pois\(\)](#), [prob_gamma_pois_zi\(\)](#), [prob_lnorm\(\)](#), [prob_neg_binom\(\)](#), [prob_norm\(\)](#), [prob_pois\(\)](#), [prob_skewlnorm\(\)](#), [prob_skewnorm\(\)](#), [prob_student\(\)](#), [prob_unif\(\)](#)

Examples

```
prob_pois_zi(c(1, 3, 4), 3, prob = 0.5)
```

prob_skewlnorm	<i>Skew-Lognormal Cumulative Distribution Function</i>
----------------	--

Description

Skew-Lognormal Cumulative Distribution Function

Usage

```
prob_skewlnorm(
  x,
  locationlog = 0,
  scalelog = 1,
  shapelog = 0,
  ...,
  meanlog,
  sdlog,
  shape
)
```

Arguments

x	A numeric vector of quantiles.
locationlog	A numeric vector of location parameters of $\log(x)$.
scalelog	A non-negative numeric vector of scale parameters of $\log(x)$.
shapelog	A numeric vector of shape parameters of $\log(x)$. Negative values result in leftward skew, while positive values result in rightward skew.
...	Unused.
meanlog	[Deprecated] A numeric vector of location parameters of $\log(x)$. Described as "a numeric vector of the means on the log scale" prior to v. 0.10.1. Will be removed in a future version.

sdlog	[Deprecated] A non-negative numeric vector of scale parameters of log(x). Described as "a non-negative numeric vector of the standard deviations on the log scale" prior to v. 0.10.1. Will be removed in a future version.
shape	[Deprecated] A numeric vector of shape parameters of log(x). Will be removed in a future version.

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_bern\(\)](#), [prob_beta\(\)](#), [prob_beta_binom\(\)](#), [prob_binom\(\)](#), [prob_exp\(\)](#), [prob_gamma\(\)](#), [prob_gamma_pois\(\)](#), [prob_gamma_pois_zi\(\)](#), [prob_lnorm\(\)](#), [prob_neg_binom\(\)](#), [prob_norm\(\)](#), [prob_pois\(\)](#), [prob_pois_zi\(\)](#), [prob_skewnorm\(\)](#), [prob_student\(\)](#), [prob_unif\(\)](#)

Examples

```
prob_skewlnorm(1:5)
prob_skewlnorm(1:5, shapelog = -2)
prob_skewlnorm(1:5, shapelog = 2)
```

prob_skewnorm	<i>Skew Normal Cumulative Distribution Function</i>
---------------	---

Description

Skew Normal Cumulative Distribution Function

Usage

```
prob_skewnorm(x, location = 0, scale = 1, shape = 0, ..., mean, sd)
```

Arguments

x	A numeric vector of quantiles.
location	A numeric vector of the location parameter.
scale	A non-negative numeric vector of the scale parameter.
shape	A numeric vector of shape. Negative values result in leftward skew, while positive values result in rightward skew.
...	Unused.
mean	[Deprecated] A numeric vector of the location parameter. Described as "a numeric vector of the means" prior to to v. 0.10.1. Will be removed in a future version.
sd	[Deprecated] A non-negative numeric vector of the scale parameter. Described as "a non-negative numeric vector of the standard deviations" prior to v. 0.10.1. Will be removed in a future version.

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_bern\(\)](#), [prob_beta\(\)](#), [prob_beta_binom\(\)](#), [prob_binom\(\)](#), [prob_exp\(\)](#), [prob_gamma\(\)](#), [prob_gamma_pois\(\)](#), [prob_gamma_pois_zi\(\)](#), [prob_lnorm\(\)](#), [prob_neg_binom\(\)](#), [prob_norm\(\)](#), [prob_pois\(\)](#), [prob_pois_zi\(\)](#), [prob_skewlnorm\(\)](#), [prob_student\(\)](#), [prob_unif\(\)](#)

Examples

```
prob_skewnorm(c(-2:2))
prob_skewnorm(c(-2:2), shape = -2)
prob_skewnorm(c(-2:2), shape = 2)
```

prob_student	<i>Student's t Cumulative Distribution Function</i>
--------------	---

Description

Student's t Cumulative Distribution Function

Usage

```
prob_student(x, mean = 0, sd = 1, theta = 0)
```

Arguments

x	A numeric vector of quantiles.
mean	A numeric vector of the means.
sd	A non-negative numeric vector of the standard deviations.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_bern\(\)](#), [prob_beta\(\)](#), [prob_beta_binom\(\)](#), [prob_binom\(\)](#), [prob_exp\(\)](#), [prob_gamma\(\)](#), [prob_gamma_pois\(\)](#), [prob_gamma_pois_zi\(\)](#), [prob_lnorm\(\)](#), [prob_neg_binom\(\)](#), [prob_norm\(\)](#), [prob_pois\(\)](#), [prob_pois_zi\(\)](#), [prob_skewlnorm\(\)](#), [prob_skewnorm\(\)](#), [prob_unif\(\)](#)

Examples

```
prob_student(c(1, 3.5, 4), mean = 1, sd = 2, theta = 1 / 3)
```

prob_unif	<i>Uniform Cumulative Distribution Function</i>
-----------	---

Description

Uniform Cumulative Distribution Function

Usage

```
prob_unif(x, min = 0, max = 1)
```

Arguments

x	A numeric vector of quantiles.
min	A numeric vector of the minimums.
max	A numeric vector of the maximums.

Value

An numeric vector of the corresponding probabilities.

See Also

Other prob_dist: [prob_bern\(\)](#), [prob_beta\(\)](#), [prob_beta_binom\(\)](#), [prob_binom\(\)](#), [prob_exp\(\)](#), [prob_gamma\(\)](#), [prob_gamma_pois\(\)](#), [prob_gamma_pois_zi\(\)](#), [prob_lnorm\(\)](#), [prob_neg_binom\(\)](#), [prob_norm\(\)](#), [prob_pois\(\)](#), [prob_pois_zi\(\)](#), [prob_skewlnorm\(\)](#), [prob_skewnorm\(\)](#), [prob_student\(\)](#)

Examples

```
prob_unif(c(0, 0.5, 1))
```

probability_direction	<i>Probability of Direction</i>
-----------------------	---------------------------------

Description

The probability of direction (PD) is the proportion of the (posterior) distribution above (right) or below (left) a threshold.

Usage

```
probability_direction(x, side = "median", threshold = 0, na_rm = FALSE)
```

Arguments

x	A numeric vector of MCMC values.
side	A character vector of length 1 indicating whether to calculate the directional probability for the left tail ("left"; $x < \text{threshold}$), or the right tail ("right"; $x > \text{threshold}$). Defaults to "median", which uses the side of the median of x via <code>direction()</code> .
threshold	A number of the threshold value, which is excluded from the interval for the probability.
na_rm	A flag specifying whether to remove missing values.

Details

By default, the direction is based on the side of the median value, but it can be specified to measure support for specific hypotheses. A right-side PD of 0.9 indicates that the interval spanning from the threshold to infinity has a coverage of 90%. Can be used as a measure of certainty in the direction of the estimate (e.g., positive or negative when using a threshold of 0). **NOTE:** probability estimates of 0 or 1 are corrected towards 0.5 by adding or subtracting $1 / (\text{length}(x) + 1)$, where x is a vector of MCMC samples. Ideally, x should be large enough as to make the correction negligible.

Value

A number between 0 and 1. If x has NA values but na_rm is FALSE, returns NA_real.

References

Makowski, D., Ben-Shachar, M.S., Chen, S.H.A., and Lüdtke, D. 2019. Indices of Effect Existence and Significance in the Bayesian Framework. *Front. Psychol.* 10: 2767. doi:10.3389/fpsyg.2019.02767.

See Also

Other summary: `direction()`, `directional_information()`, `kurtosis()`, `lower()`, `pvalue()`, `pzeros()`, `skewness()`, `svalue()`, `upper()`, `variance()`, `xtr_mean()`, `xtr_median()`, `xtr_sd()`, `zeros()`, `zscore()`

Examples

```
x <- rnorm(1e6, qnorm(0.05, lower.tail = TRUE))
probability_direction(x, side = "left")
probability_direction(x, side = "right") # = 1 - probability_direction(x, side = "left")
probability_direction(c(0, 0, 1), side = "right") # returns P(X > 0) = 1/3 instead of P(X >= 0) = 1
probability_direction(c(1, 1), side = "right") # p = 1 - 1/(n+1)
```

proportional_change	<i>Proportional Change</i>
---------------------	----------------------------

Description

Calculates the proportional change for two sets of numbers.

Usage

```
proportional_change(x, x2)
```

Arguments

x	A numeric object (vector, matrix or array) of non-negative numbers.
x2	A second numeric object of non-negative numbers.

Value

A numeric object of the proportional change.

See Also

Other proportional: [proportional_difference\(\)](#)

Examples

```
proportional_change(1, 2)
proportional_change(2, 1)
```

proportional_change2	<i>Proportional Change2</i>
----------------------	-----------------------------

Description

Calculates the proportional change for a vector of two non-negative numbers.

Usage

```
proportional_change2(x)
```

Arguments

x	A numeric vector of length 2.
---	-------------------------------

Value

A number.

See Also

Other proportional fun2: [proportional_difference2\(\)](#)

Examples

```
proportional_change2(c(1, 2))  
proportional_change2(c(2, 1))
```

`proportional_difference`

Proportional Difference

Description

Calculates the proportional difference for two sets of numbers.

Usage

```
proportional_difference(x, x2)
```

Arguments

<code>x</code>	A numeric object (vector, matrix or array) of non-negative numbers.
<code>x2</code>	A second numeric object of non-negative numbers.

Value

A numeric object of the proportional change.

See Also

Other proportional: [proportional_change\(\)](#)

Examples

```
proportional_difference(1, 2)  
proportional_difference(2, 1)
```

proportional_difference2

Proportional Difference2

Description

Calculates the proportional difference for a vector of two non-negative numbers.

Usage

```
proportional_difference2(x)
```

Arguments

x A numeric vector of length 2.

Value

A number.

See Also

Other proportional fun2: [proportional_change2\(\)](#)

Examples

```
proportional_difference2(c(1, 2))
proportional_difference2(c(2, 1))
```

pvalue

Bayesian P-Value

Description

A Bayesian p-value (p) is here defined in terms of the quantile-based (1-p) * 100% credible interval (CRI) that just includes a threshold (Kery and Schaub 2011).

Usage

```
pvalue(x, ..., side = "both", threshold = 0, skeptical = TRUE, na_rm = FALSE)
```

Arguments

x	A numeric vector of MCMC values.
...	Unused.
side	A character vector of length 1 indicating whether to calculate p-values for the left tail ("left"), right tail ("right"), or two-sided ("both"; default).
threshold	A number of the threshold value.
skeptical	A flag specifying whether or not to add one sample to the empty side of the threshold when 100% of samples are on one side. Avoids zero p-values and infinite s-values, and also imposes stronger bounds on directional information than $[-n, n]$, which assume the MCMC samples are independent and representative.
na_rm	A flag specifying whether to remove missing values.

Details

A p-value of 0.05 indicates that the 95% CRI just includes the threshold value.

Note that the function contains the sample-size correction $p_c = p * n / (n + 1)$ to avoid p-values of 0. The function can still return p-values of 1.

When `skeptical = TRUE` (default), a floor of $1/(n + 1)$ is applied to avoid p-values of 0 when all samples are on one side of the threshold. When `skeptical = FALSE`, p-values of 0 are allowed.

To use as a measure of certainty in the direction of the estimate (i.e., positive or negative), see [probability_direction\(\)](#).

For p-values converted to bits, see [svalue\(\)](#).

To convert MCMC objects to information, see [directional_information\(\)](#).

Value

A number between 0 and 1. If x has NA values but `na_rm` is FALSE, returns NA_real.

References

Kery, M., and Schaub, M. 2011. Bayesian population analysis using WinBUGS: a hierarchical perspective. Academic Press, Boston. Available from <https://www.vogelwarte.ch/en/research/population-biology/book-bpa/>.

See Also

Other summary: [direction\(\)](#), [directional_information\(\)](#), [kurtosis\(\)](#), [lower\(\)](#), [probability_direction\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr_mean\(\)](#), [xtr_median\(\)](#), [xtr_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

Examples

```
x <- rnorm(1e6, qnorm(0.05, lower.tail = TRUE))
pvalue(x) # should be 0.05 * 2
pvalue(x, side = "left") # should be 0.95
pvalue(x, side = "right") # should be 0.05
```

```
pvalue(rep(1, 10)) # skeptical = TRUE (default) avoids p = 0
pvalue(rep(1, 10), skeptical = FALSE) # skeptical = FALSE allows p = 0
```

pzeros

Proportion of Zeros

Description

The proportion of zeros in an numeric object.

Usage

```
pzeros(x, na_rm = FALSE)
```

Arguments

x A numeric object of MCMC values.
na_rm A flag specifying whether to remove missing values.

Value

A number between 0 and 1.

See Also

Other summary: [direction\(\)](#), [directional_information\(\)](#), [kurtosis\(\)](#), [lower\(\)](#), [probability_direction\(\)](#), [pvalue\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr_mean\(\)](#), [xtr_median\(\)](#), [xtr_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

Examples

```
pzeros(c(0:2))
```

quant_bern

Bernoulli Quantile Function

Description

Bernoulli Quantile Function

Usage

```
quant_bern(x, prob = 0.5)
```

Arguments

x	A numeric vector of probabilities.
prob	A numeric vector of values between 0 and 1 of the probability of success.

Value

An numeric vector of the corresponding quantiles.

See Also

Other quant_dist: [quant_beta\(\)](#), [quant_binom\(\)](#), [quant_exp\(\)](#), [quant_gamma\(\)](#), [quant_gamma_pois\(\)](#), [quant_gamma_pois_zi\(\)](#), [quant_lnorm\(\)](#), [quant_neg_binom\(\)](#), [quant_norm\(\)](#), [quant_pois\(\)](#), [quant_pois_zi\(\)](#), [quant_skewlnorm\(\)](#), [quant_skewnorm\(\)](#), [quant_student\(\)](#), [quant_unif\(\)](#)

Examples

```
quant_bern(c(0.3, 0.8), 0.7)
```

quant_beta	<i>Beta Quantile Function</i>
------------	-------------------------------

Description

Beta Quantile Function

Usage

```
quant_beta(x, alpha = 1, beta = 1)
```

Arguments

x	A numeric vector of probabilities.
alpha	The first shape parameter of the beta distribution.
beta	The second shape parameter of the beta distribution.

Value

An numeric vector of the corresponding quantiles.

See Also

Other quant_dist: [quant_bern\(\)](#), [quant_binom\(\)](#), [quant_exp\(\)](#), [quant_gamma\(\)](#), [quant_gamma_pois\(\)](#), [quant_gamma_pois_zi\(\)](#), [quant_lnorm\(\)](#), [quant_neg_binom\(\)](#), [quant_norm\(\)](#), [quant_pois\(\)](#), [quant_pois_zi\(\)](#), [quant_skewlnorm\(\)](#), [quant_skewnorm\(\)](#), [quant_student\(\)](#), [quant_unif\(\)](#)

Examples

```
quant_beta(c(0.1, 0.4, 0.6), 2, 3)
```

quant_binom	<i>Binomial Quantile Function</i>
-------------	-----------------------------------

Description

Binomial Quantile Function

Usage

```
quant_binom(x, size = 1, prob = 0.5)
```

Arguments

x	A numeric vector of probabilities.
size	A non-negative whole numeric vector of the number of trials.
prob	A numeric vector of values between 0 and 1 of the probability of success.

Value

An numeric vector of the corresponding quantiles.

See Also

Other quant_dist: [quant_bern\(\)](#), [quant_beta\(\)](#), [quant_exp\(\)](#), [quant_gamma\(\)](#), [quant_gamma_pois\(\)](#), [quant_gamma_pois_zi\(\)](#), [quant_lnorm\(\)](#), [quant_neg_binom\(\)](#), [quant_norm\(\)](#), [quant_pois\(\)](#), [quant_pois_zi\(\)](#), [quant_skewlnorm\(\)](#), [quant_skewnorm\(\)](#), [quant_student\(\)](#), [quant_unif\(\)](#)

Examples

```
quant_binom(c(0.1, 0.4, 0.6), 2, 0.3)
```

quant_exp	<i>Exponential Quantile Function</i>
-----------	--------------------------------------

Description

Exponential Quantile Function

Usage

```
quant_exp(x, rate = 1)
```

Arguments

x	A numeric vector of probabilities.
rate	A non-negative numeric vector of rate.

Value

An numeric vector of the corresponding quantiles.

See Also

Other quant_dist: [quant_bern\(\)](#), [quant_beta\(\)](#), [quant_binom\(\)](#), [quant_gamma\(\)](#), [quant_gamma_pois\(\)](#), [quant_gamma_pois_zi\(\)](#), [quant_lnorm\(\)](#), [quant_neg_binom\(\)](#), [quant_norm\(\)](#), [quant_pois\(\)](#), [quant_pois_zi\(\)](#), [quant_skewlnorm\(\)](#), [quant_skewnorm\(\)](#), [quant_student\(\)](#), [quant_unif\(\)](#)

Examples

```
quant_exp(c(0.1, 0.4, 0.6), 2)
```

quant_gamma	<i>Gamma Quantile Function</i>
-------------	--------------------------------

Description

Gamma Quantile Function

Usage

```
quant_gamma(x, shape = 1, rate = 1)
```

Arguments

- x A numeric vector of probabilities.
- shape A non-negative numeric vector of shape.
- rate A non-negative numeric vector of rate.

Value

An numeric vector of the corresponding quantiles.

See Also

Other quant_dist: [quant_bern\(\)](#), [quant_beta\(\)](#), [quant_binom\(\)](#), [quant_exp\(\)](#), [quant_gamma_pois\(\)](#), [quant_gamma_pois_zi\(\)](#), [quant_lnorm\(\)](#), [quant_neg_binom\(\)](#), [quant_norm\(\)](#), [quant_pois\(\)](#), [quant_pois_zi\(\)](#), [quant_skewlnorm\(\)](#), [quant_skewnorm\(\)](#), [quant_student\(\)](#), [quant_unif\(\)](#)

Examples

```
quant_gamma(c(0.1, 0.4, 0.6), 1, 2)
```

quant_gamma_pois	<i>Gamma-Poisson Quantile Function</i>
------------------	--

Description

Gamma-Poisson Quantile Function

Usage

```
quant_gamma_pois(x, lambda = 1, theta = 0)
```

Arguments

x	A numeric vector of probabilities.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).

Value

An numeric vector of the corresponding quantiles.

See Also

Other quant_dist: [quant_bern\(\)](#), [quant_beta\(\)](#), [quant_binom\(\)](#), [quant_exp\(\)](#), [quant_gamma\(\)](#), [quant_gamma_pois_zi\(\)](#), [quant_lnorm\(\)](#), [quant_neg_binom\(\)](#), [quant_norm\(\)](#), [quant_pois\(\)](#), [quant_pois_zi\(\)](#), [quant_skewlnorm\(\)](#), [quant_skewnorm\(\)](#), [quant_student\(\)](#), [quant_unif\(\)](#)

Examples

```
quant_gamma_pois(c(0.1, 0.4, 0.6), 1, 1)
```

quant_gamma_pois_zi	<i>Zero-Inflated Gamma-Poisson Quantile Function</i>
---------------------	--

Description

Zero-Inflated Gamma-Poisson Quantile Function

Usage

```
quant_gamma_pois_zi(x, lambda = 1, theta = 0, prob = 0)
```

Arguments

x	A numeric vector of probabilities.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
prob	A numeric vector of values between 0 and 1 of the probability of success.

Value

An numeric vector of the corresponding quantiles.

See Also

Other quant_dist: [quant_bern\(\)](#), [quant_beta\(\)](#), [quant_binom\(\)](#), [quant_exp\(\)](#), [quant_gamma\(\)](#), [quant_gamma_pois\(\)](#), [quant_lnorm\(\)](#), [quant_neg_binom\(\)](#), [quant_norm\(\)](#), [quant_pois\(\)](#), [quant_pois_zi\(\)](#), [quant_skewlnorm\(\)](#), [quant_skewnorm\(\)](#), [quant_student\(\)](#), [quant_unif\(\)](#)

Examples

```
quant_gamma_pois_zi(c(0.1, 0.4, 0.6), 3, 1, prob = 0.5)
```

quant_lnorm	<i>Log-Normal Quantile Function</i>
-------------	-------------------------------------

Description

Log-Normal Quantile Function

Usage

```
quant_lnorm(x, meanlog = 0, sdlog = 1)
```

Arguments

x	A numeric vector of probabilities.
meanlog	A numeric vector of the means on the log scale.
sdlog	A non-negative numeric vector of the standard deviations on the log scale.

Value

An numeric vector of the corresponding quantiles.

See Also

Other quant_dist: [quant_bern\(\)](#), [quant_beta\(\)](#), [quant_binom\(\)](#), [quant_exp\(\)](#), [quant_gamma\(\)](#), [quant_gamma_pois\(\)](#), [quant_gamma_pois_zi\(\)](#), [quant_neg_binom\(\)](#), [quant_norm\(\)](#), [quant_pois\(\)](#), [quant_pois_zi\(\)](#), [quant_skewlnorm\(\)](#), [quant_skewnorm\(\)](#), [quant_student\(\)](#), [quant_unif\(\)](#)

Examples

```
quant_lnorm(c(0.1, 0.4, 0.6), 0, 2)
```

quant_neg_binom	<i>Negative Binomial Quantile Function</i>
-----------------	--

Description

Negative Binomial Quantile Function

Usage

```
quant_neg_binom(x, lambda = 1, theta = 0)
```

Arguments

x	A numeric vector of probabilities.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).

Value

An numeric vector of the corresponding quantiles.

See Also

Other quant_dist: [quant_bern\(\)](#), [quant_beta\(\)](#), [quant_binom\(\)](#), [quant_exp\(\)](#), [quant_gamma\(\)](#), [quant_gamma_pois\(\)](#), [quant_gamma_pois_zi\(\)](#), [quant_lnorm\(\)](#), [quant_norm\(\)](#), [quant_pois\(\)](#), [quant_pois_zi\(\)](#), [quant_skewlnorm\(\)](#), [quant_skewnorm\(\)](#), [quant_student\(\)](#), [quant_unif\(\)](#)

Examples

```
quant_neg_binom(c(0.1, 0.4, 0.6), 2, 1)
```

quant_norm	<i>Normal Quantile Function</i>
------------	---------------------------------

Description

Normal Quantile Function

Usage

```
quant_norm(x, mean = 0, sd = 1)
```

Arguments

x	A numeric vector of probabilities.
mean	A numeric vector of the means.
sd	A non-negative numeric vector of the standard deviations.

Value

An numeric vector of the corresponding quantiles.

See Also

Other quant_dist: [quant_bern\(\)](#), [quant_beta\(\)](#), [quant_binom\(\)](#), [quant_exp\(\)](#), [quant_gamma\(\)](#), [quant_gamma_pois\(\)](#), [quant_gamma_pois_zi\(\)](#), [quant_lnorm\(\)](#), [quant_neg_binom\(\)](#), [quant_pois\(\)](#), [quant_pois_zi\(\)](#), [quant_skewlnorm\(\)](#), [quant_skewnorm\(\)](#), [quant_student\(\)](#), [quant_unif\(\)](#)

Examples

```
quant_norm(c(0.1, 0.4, 0.6))
```

quant_pois	<i>Poisson Quantile Function</i>
------------	----------------------------------

Description

Poisson Quantile Function

Usage

```
quant_pois(x, lambda = 1)
```

Arguments

x	A numeric vector of probabilities.
lambda	A non-negative numeric vector of means.

Value

An numeric vector of the corresponding quantiles.

See Also

Other quant_dist: [quant_bern\(\)](#), [quant_beta\(\)](#), [quant_binom\(\)](#), [quant_exp\(\)](#), [quant_gamma\(\)](#), [quant_gamma_pois\(\)](#), [quant_gamma_pois_zi\(\)](#), [quant_lnorm\(\)](#), [quant_neg_binom\(\)](#), [quant_norm\(\)](#), [quant_pois_zi\(\)](#), [quant_skewlnorm\(\)](#), [quant_skewnorm\(\)](#), [quant_student\(\)](#), [quant_unif\(\)](#)

Examples

```
quant_pois(c(0.1, 0.4, 0.6), 3)
```

quant_pois_zi	<i>Zero-Inflated Poisson Quantile Function</i>
---------------	--

Description

Zero-Inflated Poisson Quantile Function

Usage

```
quant_pois_zi(x, lambda = 1, prob = 0)
```

Arguments

x	A numeric vector of probabilities.
lambda	A non-negative numeric vector of means.
prob	A numeric vector of values between 0 and 1 of the probability of success.

Value

An numeric vector of the corresponding quantiles.

See Also

Other quant_dist: [quant_bern\(\)](#), [quant_beta\(\)](#), [quant_binom\(\)](#), [quant_exp\(\)](#), [quant_gamma\(\)](#), [quant_gamma_pois\(\)](#), [quant_gamma_pois_zi\(\)](#), [quant_lnorm\(\)](#), [quant_neg_binom\(\)](#), [quant_norm\(\)](#), [quant_pois\(\)](#), [quant_skewlnorm\(\)](#), [quant_skewnorm\(\)](#), [quant_student\(\)](#), [quant_unif\(\)](#)

Examples

```
quant_pois_zi(c(0.1, 0.4, 0.6), 3, prob = 0.5)
```

quant_skewlnorm	<i>Skew-Lognormal Quantile Function</i>
-----------------	---

Description

Skew-Lognormal Quantile Function

Usage

```
quant_skewlnorm(
  x,
  locationlog = 0,
  scalelog = 1,
  shapelog = 0,
  ...,
  meanlog,
  sdlog,
  shape
)
```

Arguments

x	A numeric vector of probabilities.
locationlog	A numeric vector of location parameters of $\log(x)$.
scalelog	A non-negative numeric vector of scale parameters of $\log(x)$.
shapelog	A numeric vector of shape parameters of $\log(x)$. Negative values result in leftward skew, while positive values result in rightward skew.
...	Unused.
meanlog	[Deprecated] A numeric vector of location parameters of $\log(x)$. Described as "a numeric vector of the means on the log scale" prior to v. 0.10.1. Will be removed in a future version.
sdlog	[Deprecated] A non-negative numeric vector of scale parameters of $\log(x)$. Described as "a non-negative numeric vector of the standard deviations on the log scale" prior to v. 0.10.1. Will be removed in a future version.
shape	[Deprecated] A numeric vector of shape parameters of $\log(x)$. Will be removed in a future version.

Value

An numeric vector of the corresponding quantiles.

See Also

Other quant_dist: [quant_bern\(\)](#), [quant_beta\(\)](#), [quant_binom\(\)](#), [quant_exp\(\)](#), [quant_gamma\(\)](#), [quant_gamma_pois\(\)](#), [quant_gamma_pois_zi\(\)](#), [quant_lnorm\(\)](#), [quant_neg_binom\(\)](#), [quant_norm\(\)](#), [quant_pois\(\)](#), [quant_pois_zi\(\)](#), [quant_skewnorm\(\)](#), [quant_student\(\)](#), [quant_unif\(\)](#)

Examples

```
quant_skewlnorm(c(0.1, 0.4, 0.6))
quant_skewlnorm(c(0.1, 0.4, 0.6), shapelog = -2)
quant_skewlnorm(c(0.1, 0.4, 0.6), shapelog = 2)
```

quant_skewnrm	<i>Skew Normal Quantile Function</i>
---------------	--------------------------------------

Description

Skew Normal Quantile Function

Usage

```
quant_skewnrm(x, location = 0, scale = 1, shape = 0, ..., mean, sd)
```

Arguments

x	A numeric vector of probabilities.
location	A numeric vector of the location parameter.
scale	A non-negative numeric vector of the scale parameter.
shape	A numeric vector of shape. Negative values result in leftward skew, while positive values result in rightward skew.
...	Unused.
mean	[Deprecated] A numeric vector of the location parameter. Described as "a numeric vector of the means" prior to v. 0.10.1. Will be removed in a future version.
sd	[Deprecated] A non-negative numeric vector of the scale parameter. Described as "a non-negative numeric vector of the standard deviations" prior to v. 0.10.1. Will be removed in a future version.

Value

An numeric vector of the corresponding quantiles.

See Also

Other quant_dist: [quant_bern\(\)](#), [quant_beta\(\)](#), [quant_binom\(\)](#), [quant_exp\(\)](#), [quant_gamma\(\)](#), [quant_gamma_pois\(\)](#), [quant_gamma_pois_zi\(\)](#), [quant_lnorm\(\)](#), [quant_neg_binom\(\)](#), [quant_norm\(\)](#), [quant_pois\(\)](#), [quant_pois_zi\(\)](#), [quant_skewlnorm\(\)](#), [quant_student\(\)](#), [quant_unif\(\)](#)

Examples

```
quant_skewnrm(c(0.1, 0.4, 0.6))
quant_skewnrm(c(0.1, 0.4, 0.6), shape = -2)
quant_skewnrm(c(0.1, 0.4, 0.6), shape = 2)
```

quant_student	<i>Student's t Quantile Function</i>
---------------	--------------------------------------

Description

Student's t Quantile Function

Usage

```
quant_student(x, mean = 0, sd = 1, theta = 0)
```

Arguments

x	A numeric vector of probabilities.
mean	A numeric vector of the means.
sd	A non-negative numeric vector of the standard deviations.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).

Value

An numeric vector of the corresponding quantiles.

See Also

Other quant_dist: [quant_bern\(\)](#), [quant_beta\(\)](#), [quant_binom\(\)](#), [quant_exp\(\)](#), [quant_gamma\(\)](#), [quant_gamma_pois\(\)](#), [quant_gamma_pois_zi\(\)](#), [quant_lnorm\(\)](#), [quant_neg_binom\(\)](#), [quant_norm\(\)](#), [quant_pois\(\)](#), [quant_pois_zi\(\)](#), [quant_skewlnorm\(\)](#), [quant_skewnorm\(\)](#), [quant_unif\(\)](#)

Examples

```
quant_student(c(0.1, 0.4, 0.6), mean = 1, sd = 2, theta = 1 / 3)
```

quant_unif	<i>Uniform Quantile Function</i>
------------	----------------------------------

Description

Uniform Quantile Function

Usage

```
quant_unif(x, min = 0, max = 1)
```

Arguments

x	A numeric vector of probabilities.
min	A numeric vector of the minimums.
max	A numeric vector of the maximums.

Value

An numeric vector of the corresponding quantiles.

See Also

Other quant_dist: [quant_bern\(\)](#), [quant_beta\(\)](#), [quant_binom\(\)](#), [quant_exp\(\)](#), [quant_gamma\(\)](#), [quant_gamma_pois\(\)](#), [quant_gamma_pois_zi\(\)](#), [quant_lnorm\(\)](#), [quant_neg_binom\(\)](#), [quant_norm\(\)](#), [quant_pois\(\)](#), [quant_pois_zi\(\)](#), [quant_skewlnorm\(\)](#), [quant_skewnorm\(\)](#), [quant_student\(\)](#)

Examples

```
quant_unif(c(0.1, 0.4, 0.6))
```

ran_bern

Bernoulli Random Samples

Description

Bernoulli Random Samples

Usage

```
ran_bern(n = 1, prob = 0.5)
```

Arguments

n	A non-negative whole number of the number of random samples to generate.
prob	A numeric vector of values between 0 and 1 of the probability of success.

Value

A numeric vector of the random samples.

See Also

Other ran_dist: [ran_beta_binom\(\)](#), [ran_binom\(\)](#), [ran_gamma\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma_pois_zi\(\)](#), [ran_lnorm\(\)](#), [ran_multinom\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewlnorm\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

```
ran_bern(10)
```

ran_beta_binom	<i>Beta-Binomial Random Samples</i>
----------------	-------------------------------------

Description

This parameterization of the beta-binomial distribution uses an expected probability parameter, prob, and a dispersion parameter, theta. The parameters of the underlying beta mixture are $\alpha = (2 * \text{prob}) / \text{theta}$ and $\beta = (2 * (1 - \text{prob})) / \text{theta}$. This parameterization of theta is unconventional, but has useful properties when modelling. When $\text{theta} = 0$, the beta-binomial reverts to the binomial distribution. When $\text{theta} = 1$ and $\text{prob} = 0.5$, the parameters of the beta distribution become $\alpha = 1$ and $\beta = 1$, which correspond to a uniform distribution for the beta-binomial probability parameter.

Usage

```
ran_beta_binom(n = 1, size = 1, prob = 0.5, theta = 0)
```

Arguments

n	A non-negative whole number of the number of random samples to generate.
size	A non-negative whole numeric vector of the number of trials.
prob	A numeric vector of values between 0 and 1 of the probability of success.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).

Value

A numeric vector of the random samples.

See Also

Other ran_dist: [ran_bern\(\)](#), [ran_binom\(\)](#), [ran_gamma\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma_pois_zi\(\)](#), [ran_lnorm\(\)](#), [ran_multinom\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewlnorm\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

```
ran_beta_binom(10, 1, 0.5, 0)
```

ran_binom	<i>Binomial Random Samples</i>
-----------	--------------------------------

Description

Binomial Random Samples

Usage

```
ran_binom(n = 1, size = 1, prob = 0.5)
```

Arguments

n	A non-negative whole number of the number of random samples to generate.
size	A non-negative whole numeric vector of the number of trials.
prob	A numeric vector of values between 0 and 1 of the probability of success.

Value

A numeric vector of the random samples.

See Also

Other ran_dist: [ran_bern\(\)](#), [ran_beta_binom\(\)](#), [ran_gamma\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma_pois_zi\(\)](#), [ran_lnorm\(\)](#), [ran_multinom\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewlnorm\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

```
ran_binom(10)
```

ran_gamma	<i>Gamma Random Samples</i>
-----------	-----------------------------

Description

Gamma Random Samples

Usage

```
ran_gamma(n = 1, shape = 1, rate = 1)
```

Arguments

n	A non-negative whole number of the number of random samples to generate.
shape	A non-negative numeric vector of shape.
rate	A non-negative numeric vector of rate.

Value

A numeric vector of the random samples.

See Also

Other `ran_dist`: `ran_bern()`, `ran_beta_binom()`, `ran_binom()`, `ran_gamma_pois()`, `ran_gamma_pois_zi()`, `ran_lnorm()`, `ran_multinom()`, `ran_neg_binom()`, `ran_norm()`, `ran_pois()`, `ran_pois_zi()`, `ran_skewlnorm()`, `ran_skewnorm()`, `ran_student()`

Examples

```
ran_gamma(10)
```

ran_gamma_pois	<i>Gamma-Poisson Random Samples</i>
----------------	-------------------------------------

Description

Gamma-Poisson Random Samples

Usage

```
ran_gamma_pois(n = 1, lambda = 1, theta = 0)
```

Arguments

<code>n</code>	A non-negative whole number of the number of random samples to generate.
<code>lambda</code>	A non-negative numeric vector of means.
<code>theta</code>	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).

Value

A numeric vector of the random samples.

See Also

Other `ran_dist`: `ran_bern()`, `ran_beta_binom()`, `ran_binom()`, `ran_gamma()`, `ran_gamma_pois_zi()`, `ran_lnorm()`, `ran_multinom()`, `ran_neg_binom()`, `ran_norm()`, `ran_pois()`, `ran_pois_zi()`, `ran_skewlnorm()`, `ran_skewnorm()`, `ran_student()`

Examples

```
ran_gamma_pois(10, theta = 1)
```

ran_gamma_pois_zi	<i>Zero-Inflated Gamma-Poisson Random Samples</i>
-------------------	---

Description

Zero-Inflated Gamma-Poisson Random Samples

Usage

```
ran_gamma_pois_zi(n = 1, lambda = 1, theta = 0, prob = 0)
```

Arguments

n	A non-negative whole number of the number of random samples to generate.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
prob	A numeric vector of values between 0 and 1 of the probability of success.

Value

A numeric vector of the random samples.

See Also

Other `ran_dist`: [ran_bern\(\)](#), [ran_beta_binom\(\)](#), [ran_binom\(\)](#), [ran_gamma\(\)](#), [ran_gamma_pois\(\)](#), [ran_lnorm\(\)](#), [ran_multinom\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewlnorm\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

```
ran_gamma_pois_zi(10, lambda = 3, theta = 1, prob = 0.5)
```

ran_lnorm	<i>Log-Normal Random Samples</i>
-----------	----------------------------------

Description

Log-Normal Random Samples

Usage

```
ran_lnorm(n = 1, meanlog = 0, sdlog = 1)
```

Arguments

n	A non-negative whole number of the number of random samples to generate.
meanlog	A numeric vector of the means on the log scale.
sdlog	A non-negative numeric vector of the standard deviations on the log scale.

Value

A numeric vector of the random samples.

See Also

Other `ran_dist`: [ran_bern\(\)](#), [ran_beta_binom\(\)](#), [ran_binom\(\)](#), [ran_gamma\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma_pois_zi\(\)](#), [ran_multinom\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewlnorm\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

```
ran_lnorm(10)
```

ran_multinom	<i>Multinomial Random Samples</i>
--------------	-----------------------------------

Description

Models the counts across two or more mutually exclusive categories from a fixed number of trials, in *long* format: one value per category per trial, with group identifying which rows belong to the same trial. All rows sharing a group must have the same size, and their prob values must sum to 1.

Usage

```
ran_multinom(size = 1, prob, group)
```

Arguments

size	A non-negative whole numeric vector of the number of trials.
prob	A numeric vector of the probability of the category. Must sum to 1 across the rows sharing the same group. NA in size or prob for any row of a trial makes the sample NA for every row of that trial, since a trial's categories are drawn jointly.
group	A vector identifying which rows belong to the same multinomial trial (whose x values sum to size and prob values sum to 1). Every group must have at least 2 rows and the same number of rows as the rest of the data (a fixed set of categories, as in multinomial logistic regression), and must not contain NA.

Details

Unlike the other `ran_*()` functions, `ran_multinom()` has no `n` argument: the number of samples is fully determined by `length(prob)` (equivalently `length(group)`), since a trial's categories can't be generated independently of one another.

Value

An integer vector of the random samples, one per row of `prob`.

References

Johnson, N.L., Kotz, S., and Balakrishnan, N. 1997. Discrete Multivariate Distributions. John Wiley and Sons, New York.

Gelman, A., Meng, X.-L., and Stern, H. 1996. Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica* 6(4): 733-807.

See Also

Other `ran_dist`: [ran_bern\(\)](#), [ran_beta_binom\(\)](#), [ran_binom\(\)](#), [ran_gamma\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma_pois_zi\(\)](#), [ran_lnorm\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewlnorm\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

```
ran_multinom(size = 10, prob = c(0.2, 0.3, 0.5), group = c(1, 1, 1))
```

ran_neg_binom	<i>Negative Binomial Random Samples</i>
---------------	---

Description

Identical to Gamma-Poisson Random Samples.

Usage

```
ran_neg_binom(n = 1, lambda = 1, theta = 0)
```

Arguments

<code>n</code>	A non-negative whole number of the number of random samples to generate.
<code>lambda</code>	A non-negative numeric vector of means.
<code>theta</code>	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).

Value

A numeric vector of the random samples.

See Also

Other ran_dist: [ran_bern\(\)](#), [ran_beta_binom\(\)](#), [ran_binom\(\)](#), [ran_gamma\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma_pois_zi\(\)](#), [ran_lnorm\(\)](#), [ran_multinom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewlnorm\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

```
ran_neg_binom(10, theta = 1)
```

ran_norm

Normal Random Samples

Description

Normal Random Samples

Usage

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

Arguments

n	A non-negative whole number of the number of random samples to generate.
mean	A numeric vector of the means.
sd	A non-negative numeric vector of the standard deviations.

Value

A numeric vector of the random samples.

See Also

Other ran_dist: [ran_bern\(\)](#), [ran_beta_binom\(\)](#), [ran_binom\(\)](#), [ran_gamma\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma_pois_zi\(\)](#), [ran_lnorm\(\)](#), [ran_multinom\(\)](#), [ran_neg_binom\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewlnorm\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

```
ran_norm(10)
```

ran_pois	<i>Poisson Random Samples</i>
----------	-------------------------------

Description

Poisson Random Samples

Usage

```
ran_pois(n = 1, lambda = 1)
```

Arguments

n	A non-negative whole number of the number of random samples to generate.
lambda	A non-negative numeric vector of means.

Value

A numeric vector of the random samples.

See Also

Other ran_dist: [ran_bern\(\)](#), [ran_beta_binom\(\)](#), [ran_binom\(\)](#), [ran_gamma\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma_pois_zi\(\)](#), [ran_lnorm\(\)](#), [ran_multinom\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois_zi\(\)](#), [ran_skewlnorm\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

```
ran_pois(10)
```

ran_pois_zi	<i>Zero-Inflated Poisson Random Samples</i>
-------------	---

Description

Zero-Inflated Poisson Random Samples

Usage

```
ran_pois_zi(n = 1, lambda = 1, prob = 0)
```

Arguments

n	A non-negative whole number of the number of random samples to generate.
lambda	A non-negative numeric vector of means.
prob	A numeric vector of values between 0 and 1 of the probability of success.

Value

A numeric vector of the random samples.

See Also

Other ran_dist: [ran_bern\(\)](#), [ran_beta_binom\(\)](#), [ran_binom\(\)](#), [ran_gamma\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma_pois_zi\(\)](#), [ran_lnorm\(\)](#), [ran_multinom\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_skewlnorm\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

```
ran_pois_zi(10, prob = 0.5)
```

ran_skewlnorm	<i>Skew-Lognormal Random Samples</i>
---------------	--------------------------------------

Description

Skew-Lognormal Random Samples

Usage

```
ran_skewlnorm(  
  n = 1,  
  locationlog = 0,  
  scalelog = 1,  
  shapelog = 0,  
  ...,  
  meanlog,  
  sdlog,  
  shape  
)
```

Arguments

- | | |
|-------------|---|
| n | A non-negative whole number of the number of random samples to generate. |
| locationlog | A numeric vector of location parameters of log(x). |
| scalelog | A non-negative numeric vector of scale parameters of log(x). |
| shapelog | A numeric vector of shape parameters of log(x). Negative values result in leftward skew, while positive values result in rightward skew. |
| ... | Unused. |
| meanlog | [Deprecated] A numeric vector of location parameters of log(x). Described as "a numeric vector of the means on the log scale" prior to v. 0.10.1. Will be removed in a future version. |

sdlog	[Deprecated] A non-negative numeric vector of scale parameters of $\log(x)$. Described as "a non-negative numeric vector of the standard deviations on the log scale" prior to v. 0.10.1. Will be removed in a future version.
shape	[Deprecated] A numeric vector of shape parameters of $\log(x)$. Will be removed in a future version.

Value

A numeric vector of the random samples.

See Also

Other `ran_dist`: `ran_bern()`, `ran_beta_binom()`, `ran_binom()`, `ran_gamma()`, `ran_gamma_pois()`, `ran_gamma_pois_zi()`, `ran_lnorm()`, `ran_multinom()`, `ran_neg_binom()`, `ran_norm()`, `ran_pois()`, `ran_pois_zi()`, `ran_skewnorm()`, `ran_student()`

Examples

```
ran_skewlnorm(10, shapelog = -1)
ran_skewlnorm(10, shapelog = 0)
ran_skewlnorm(10, shapelog = 1)
```

ran_skewnorm	<i>Skew Normal Random Samples</i>
--------------	-----------------------------------

Description

Skew Normal Random Samples

Usage

```
ran_skewnorm(n = 1, location = 0, scale = 1, shape = 0, ..., mean, sd)
```

Arguments

n	A non-negative whole number of the number of random samples to generate.
location	A numeric vector of the location parameter.
scale	A non-negative numeric vector of the scale parameter.
shape	A numeric vector of shape. Negative values result in leftward skew, while positive values result in rightward skew.
...	Unused.
mean	[Deprecated] A numeric vector of the location parameter. Described as "a numeric vector of the means" prior to v. 0.10.1. Will be removed in a future version.
sd	[Deprecated] A non-negative numeric vector of the scale parameter. Described as "a non-negative numeric vector of the standard deviations" prior to v. 0.10.1. Will be removed in a future version.

Value

A numeric vector of the random samples.

See Also

Other ran_dist: [ran_bern\(\)](#), [ran_beta_binom\(\)](#), [ran_binom\(\)](#), [ran_gamma\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma_pois_zi\(\)](#), [ran_lnorm\(\)](#), [ran_multinom\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewlnorm\(\)](#), [ran_student\(\)](#)

Examples

```
ran_skewnorm(10, shape = -1)
ran_skewnorm(10, shape = 0)
ran_skewnorm(10, shape = 1)
```

ran_student	<i>Student's t Random Samples</i>
-------------	-----------------------------------

Description

Student's t Random Samples

Usage

```
ran_student(n = 1, mean = 0, sd = 1, theta = 0)
```

Arguments

n	A non-negative whole number of the number of random samples to generate.
mean	A numeric vector of the means.
sd	A non-negative numeric vector of the standard deviations.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).

Value

A numeric vector of the random samples.

See Also

Other ran_dist: [ran_bern\(\)](#), [ran_beta_binom\(\)](#), [ran_binom\(\)](#), [ran_gamma\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma_pois_zi\(\)](#), [ran_lnorm\(\)](#), [ran_multinom\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewlnorm\(\)](#), [ran_skewnorm\(\)](#)

Examples

```
ran_student(10, theta = 1 / 2)
```

res_bern	<i>Bernoulli Residuals</i>
----------	----------------------------

Description

Bernoulli Residuals

Usage

```
res_bern(x, prob = 0.5, type = "dev", simulate = FALSE)
```

Arguments

x	A vector of 0s and 1s.
prob	A numeric vector of values between 0 and 1 of the probability of success.
type	A string of the residual type. 'raw' for raw residuals 'dev' for deviance residuals and 'data' for the data.
simulate	A flag specifying whether to simulate residuals.

Value

An numeric vector of the corresponding residuals.

See Also

Other res_dist: [res_beta_binom\(\)](#), [res_binom\(\)](#), [res_gamma\(\)](#), [res_gamma_pois\(\)](#), [res_gamma_pois_zi\(\)](#), [res_lnorm\(\)](#), [res_multinom\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [res_skewlnorm\(\)](#), [res_skewnorm\(\)](#), [res_student\(\)](#)

Examples

```
res_bern(c(TRUE, FALSE), 0.7)
```

res_beta_binom	<i>Beta-Binomial Residuals</i>
----------------	--------------------------------

Description

This parameterization of the beta-binomial distribution uses an expected probability parameter, prob, and a dispersion parameter, theta. The parameters of the underlying beta mixture are $\alpha = (2 * \text{prob}) / \text{theta}$ and $\beta = (2 * (1 - \text{prob})) / \text{theta}$. This parameterization of theta is unconventional, but has useful properties when modelling. When $\text{theta} = 0$, the beta-binomial reverts to the binomial distribution. When $\text{theta} = 1$ and $\text{prob} = 0.5$, the parameters of the beta distribution become $\alpha = 1$ and $\beta = 1$, which correspond to a uniform distribution for the beta-binomial probability parameter.

Usage

```
res_beta_binom(
  x,
  size = 1,
  prob = 0.5,
  theta = 0,
  type = "dev",
  simulate = FALSE
)
```

Arguments

x	A non-negative whole numeric vector of values.
size	A non-negative whole numeric vector of the number of trials.
prob	A numeric vector of values between 0 and 1 of the probability of success.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
type	A string of the residual type. 'raw' for raw residuals 'dev' for deviance residuals and 'data' for the data.
simulate	A flag specifying whether to simulate residuals.

Value

An numeric vector of the corresponding residuals.

See Also

Other res_dist: [res_bern\(\)](#), [res_binom\(\)](#), [res_gamma\(\)](#), [res_gamma_pois\(\)](#), [res_gamma_pois_zi\(\)](#), [res_lnorm\(\)](#), [res_multinom\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [res_skewlnorm\(\)](#), [res_skewnorm\(\)](#), [res_student\(\)](#)

Examples

```
res_beta_binom(c(0, 1, 2), 4, 0.5, 0.1)
```

res_binom

Binomial Residuals

Description

Binomial Residuals

Usage

```
res_binom(x, size = 1, prob = 0.5, type = "dev", simulate = FALSE)
```

Arguments

x	A non-negative whole numeric vector of values.
size	A non-negative whole numeric vector of the number of trials.
prob	A numeric vector of values between 0 and 1 of the probability of success.
type	A string of the residual type. 'raw' for raw residuals 'dev' for deviance residuals and 'data' for the data.
simulate	A flag specifying whether to simulate residuals.

Value

An numeric vector of the corresponding residuals.

See Also

Other res_dist: [res_bern\(\)](#), [res_beta_binom\(\)](#), [res_gamma\(\)](#), [res_gamma_pois\(\)](#), [res_gamma_pois_zi\(\)](#), [res_lnorm\(\)](#), [res_multinom\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [res_skewlnorm\(\)](#), [res_skewnorm\(\)](#), [res_student\(\)](#)

Examples

```
res_binom(c(0, 1, 2), 2, 0.3)
```

res_gamma	<i>Gamma Residuals</i>
-----------	------------------------

Description

Gamma Residuals

Usage

```
res_gamma(x, shape = 1, rate = 1, type = "dev", simulate = FALSE)
```

Arguments

x	A numeric vector of values.
shape	A non-negative numeric vector of shape.
rate	A non-negative numeric vector of rate.
type	A string of the residual type. 'raw' for raw residuals 'dev' for deviance residuals and 'data' for the data.
simulate	A flag specifying whether to simulate residuals.

Value

An numeric vector of the corresponding residuals.

See Also

Other res_dist: [res_bern\(\)](#), [res_beta_binom\(\)](#), [res_binom\(\)](#), [res_gamma_pois\(\)](#), [res_gamma_pois_zi\(\)](#), [res_lnorm\(\)](#), [res_multinom\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [res_skewlnorm\(\)](#), [res_skewnorm\(\)](#), [res_student\(\)](#)

Examples

```
res_gamma(c(0, 1, 2), 1, 2)
```

res_gamma_pois	<i>Gamma-Poisson Residuals</i>
----------------	--------------------------------

Description

Gamma-Poisson Residuals

Usage

```
res_gamma_pois(x, lambda = 1, theta = 0, type = "dev", simulate = FALSE)
```

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
type	A string of the residual type. 'raw' for raw residuals 'dev' for deviance residuals and 'data' for the data.
simulate	A flag specifying whether to simulate residuals.

Value

An numeric vector of the corresponding residuals.

See Also

Other res_dist: [res_bern\(\)](#), [res_beta_binom\(\)](#), [res_binom\(\)](#), [res_gamma\(\)](#), [res_gamma_pois_zi\(\)](#), [res_lnorm\(\)](#), [res_multinom\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [res_skewlnorm\(\)](#), [res_skewnorm\(\)](#), [res_student\(\)](#)

Examples

```
res_gamma_pois(c(0, 1, 2), 1, 1)
```

res_gamma_pois_zi	<i>Zero-Inflated Gamma-Poisson Residuals</i>
-------------------	--

Description

Zero-Inflated Gamma-Poisson Residuals

Usage

```
res_gamma_pois_zi(
  x,
  lambda = 1,
  theta = 0,
  prob = 0,
  type = "dev",
  simulate = FALSE
)
```

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
prob	A numeric vector of values between 0 and 1 of the probability of zero-inflation.
type	A string of the residual type. 'raw' for raw residuals 'dev' for deviance residuals and 'data' for the data.
simulate	A flag specifying whether to simulate residuals.

Value

An numeric vector of the corresponding residuals.

See Also

Other res_dist: [res_bern\(\)](#), [res_beta_binom\(\)](#), [res_binom\(\)](#), [res_gamma\(\)](#), [res_gamma_pois\(\)](#), [res_lnorm\(\)](#), [res_multinom\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [res_skewlnorm\(\)](#), [res_skewnorm\(\)](#), [res_student\(\)](#)

Examples

```
res_gamma_pois_zi(c(0, 1, 2), 1, 1, 0.5)
```

res_lnorm	<i>Log-Normal Residuals</i>
-----------	-----------------------------

Description

Log-Normal Residuals

Usage

```
res_lnorm(x, meanlog = 0, sdlog = 1, type = "dev", simulate = FALSE)
```

Arguments

x	A numeric vector of values.
meanlog	A numeric vector of the means on the log scale.
sdlog	A non-negative numeric vector of the standard deviations on the log scale.
type	A string of the residual type. 'raw' for raw residuals 'dev' for deviance residuals and 'data' for the data.
simulate	A flag specifying whether to simulate residuals.

Value

An numeric vector of the corresponding residuals.

See Also

Other res_dist: [res_bern\(\)](#), [res_beta_binom\(\)](#), [res_binom\(\)](#), [res_gamma\(\)](#), [res_gamma_pois\(\)](#), [res_gamma_pois_zi\(\)](#), [res_multinom\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [res_skewlnorm\(\)](#), [res_skewnorm\(\)](#), [res_student\(\)](#)

Examples

```
res_lnorm(10)
```

res_multinom	<i>Multinomial Residuals</i>
--------------	------------------------------

Description

Models the counts across two or more mutually exclusive categories from a fixed number of trials, in *long* format: one row per category per trial, with group identifying which rows belong to the same trial (see [log_lik_multinom\(\)](#) for details). `res_multinom()` returns one residual per row, not one per trial, since a trial's categories aren't independent and so have no single meaningful residual as a whole; the classic per-trial deviance statistic can be recovered by summing the squared `type = "dev"` residuals within a group.

Usage

```
res_multinom(x, size = 1, prob, group, type = "dev", simulate = FALSE)
```

Arguments

<code>x</code>	A non-negative whole numeric vector of the category counts.
<code>size</code>	A non-negative whole numeric vector of the number of trials.
<code>prob</code>	A numeric vector of the probability of the category. Must sum to 1 across the rows sharing the same group.
<code>group</code>	A vector identifying which rows belong to the same multinomial trial (whose <code>x</code> values sum to <code>size</code> and <code>prob</code> values sum to 1). Every group must have at least 2 rows and the same number of rows as the rest of the data (a fixed set of categories, as in multinomial logistic regression), and must not contain NA.
<code>type</code>	A string of the residual type. 'raw' for raw residuals 'dev' for deviance residuals and 'data' for the data.
<code>simulate</code>	A flag specifying whether to simulate residuals.

Details

`group` is validated (same size, `prob` summing to 1, no singleton or short groups, no NA) regardless of `simulate`, but is only otherwise used when `simulate = TRUE`, to draw a joint, correlation-preserving replicate per trial (via `ran_multinom()`) rather than simulating each category independently, which requires `res_multinom()` to see every row of a group in the same call.

Value

An numeric vector of the corresponding residuals.

References

- Haberman, S.J. 1973. The analysis of residuals in cross-classified tables. *Biometrics* 29(1): 205-220. doi:[10.2307/2529686](https://doi.org/10.2307/2529686).
- Pierce, D.A., and Schafer, D.W. 1986. Residuals in generalized linear models. *Journal of the American Statistical Association* 81(396): 977-986. doi:[10.1080/01621459.1986.10478361](https://doi.org/10.1080/01621459.1986.10478361).
- Gelman, A., Meng, X.-L., and Stern, H. 1996. Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica* 6(4): 733-807.

See Also

Other `res_dist`: `res_bern()`, `res_beta_binom()`, `res_binom()`, `res_gamma()`, `res_gamma_pois()`, `res_gamma_pois_zi()`, `res_lnorm()`, `res_neg_binom()`, `res_norm()`, `res_pois()`, `res_pois_zi()`, `res_skewlnorm()`, `res_skewnorm()`, `res_student()`

Examples

```
res_multinom(c(1, 3, 6), size = 10, prob = c(0.2, 0.3, 0.5), group = c(1, 1, 1))
```

res_neg_binom	<i>Negative Binomial Residuals</i>
---------------	------------------------------------

Description

Negative Binomial Residuals

Usage

```
res_neg_binom(x, lambda = 1, theta = 0, type = "dev", simulate = FALSE)
```

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
type	A string of the residual type. 'raw' for raw residuals 'dev' for deviance residuals and 'data' for the data.
simulate	A flag specifying whether to simulate residuals.

Value

An numeric vector of the corresponding residuals.

See Also

Other res_dist: [res_bern\(\)](#), [res_beta_binom\(\)](#), [res_binom\(\)](#), [res_gamma\(\)](#), [res_gamma_pois\(\)](#), [res_gamma_pois_zi\(\)](#), [res_lnorm\(\)](#), [res_multinom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [res_skewlnorm\(\)](#), [res_skewnorm\(\)](#), [res_student\(\)](#)

Examples

```
res_neg_binom(c(0, 1, 5), 2, 3)
```

res_norm	<i>Normal Residuals</i>
----------	-------------------------

Description

Normal Residuals

Usage

```
res_norm(x, mean = 0, sd = 1, type = "dev", simulate = FALSE)
```

Arguments

- x A numeric vector of values.
- mean A numeric vector of the means.
- sd A non-negative numeric vector of the standard deviations.
- type A string of the residual type. 'raw' for raw residuals 'dev' for deviance residuals and 'data' for the data.
- simulate A flag specifying whether to simulate residuals.

Value

An numeric vector of the corresponding residuals.

See Also

Other res_dist: [res_bern\(\)](#), [res_beta_binom\(\)](#), [res_binom\(\)](#), [res_gamma\(\)](#), [res_gamma_pois\(\)](#), [res_gamma_pois_zi\(\)](#), [res_lnorm\(\)](#), [res_multinom\(\)](#), [res_neg_binom\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [res_skewlnorm\(\)](#), [res_skewnorm\(\)](#), [res_student\(\)](#)

Examples

```
res_norm(c(-2:2))
```

res_pois	<i>Poisson Residuals</i>
----------	--------------------------

Description

Poisson Residuals

Usage

```
res_pois(x, lambda = 1, type = "dev", simulate = FALSE)
```

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
type	A string of the residual type. 'raw' for raw residuals 'dev' for deviance residuals and 'data' for the data.
simulate	A flag specifying whether to simulate residuals.

Value

An numeric vector of the corresponding residuals.

See Also

Other res_dist: [res_bern\(\)](#), [res_beta_binom\(\)](#), [res_binom\(\)](#), [res_gamma\(\)](#), [res_gamma_pois\(\)](#), [res_gamma_pois_zi\(\)](#), [res_lnorm\(\)](#), [res_multinom\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois_zi\(\)](#), [res_skewlnorm\(\)](#), [res_skewnorm\(\)](#), [res_student\(\)](#)

Examples

```
res_pois(c(1, 3, 4), 3)
```

res_pois_zi	<i>Zero-Inflated Poisson Residuals</i>
-------------	--

Description

Zero-Inflated Poisson Residuals

Usage

```
res_pois_zi(x, lambda = 1, prob = 0, type = "dev", simulate = FALSE)
```

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
prob	A numeric vector of values between 0 and 1 of the probability of zero-inflation.
type	A string of the residual type. 'raw' for raw residuals 'dev' for deviance residuals and 'data' for the data.
simulate	A flag specifying whether to simulate residuals.

Value

An numeric vector of the corresponding residuals.

See Also

Other res_dist: [res_bern\(\)](#), [res_beta_binom\(\)](#), [res_binom\(\)](#), [res_gamma\(\)](#), [res_gamma_pois\(\)](#), [res_gamma_pois_zi\(\)](#), [res_lnorm\(\)](#), [res_multinom\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_skewlnorm\(\)](#), [res_skewnorm\(\)](#), [res_student\(\)](#)

Examples

```
res_pois_zi(c(1, 3, 4), 6, 0.5, type = "raw")
```

res_skewlnorm	<i>Skew-Lognormal Residuals</i>
---------------	---------------------------------

Description

Skew-Lognormal Residuals

Usage

```
res_skewlnorm(  
  x,  
  locationlog = 0,  
  scalelog = 1,  
  shapelog = 0,  
  type = "dev",  
  simulate = FALSE,  
  ...,  
  meanlog,  
  sdlog,  
  shape  
)
```

Arguments

x	A numeric vector of values.
locationlog	A numeric vector of location parameters of log(x).
scalelog	A non-negative numeric vector of scale parameters of log(x).
shapelog	A numeric vector of shape parameters of log(x). Negative values result in leftward skew, while positive values result in rightward skew.
type	A string of the residual type. 'raw' for raw residuals 'dev' for deviance residuals and 'data' for the data.
simulate	A flag specifying whether to simulate residuals.
...	Unused.
meanlog	[Deprecated] A numeric vector of location parameters of log(x). Described as "a numeric vector of the means on the log scale" prior to v. 0.10.1. Will be removed in a future version.

sdlog	[Deprecated] A non-negative numeric vector of scale parameters of $\log(x)$. Described as "a non-negative numeric vector of the standard deviations on the log scale" prior to v. 0.10.1. Will be removed in a future version.
shape	[Deprecated] A numeric vector of shape parameters of $\log(x)$. Will be removed in a future version.

Value

An numeric vector of the corresponding residuals.

See Also

Other res_dist: [res_bern\(\)](#), [res_beta_binom\(\)](#), [res_binom\(\)](#), [res_gamma\(\)](#), [res_gamma_pois\(\)](#), [res_gamma_pois_zi\(\)](#), [res_lnorm\(\)](#), [res_multinom\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [res_skewnorm\(\)](#), [res_student\(\)](#)

Examples

```
res_skewlnorm(exp(-2:2))
```

res_skewnorm	<i>Skew Normal Residuals</i>
--------------	------------------------------

Description

Skew Normal Residuals

Usage

```
res_skewnorm(
  x,
  location = 0,
  scale = 1,
  shape = 0,
  type = "dev",
  simulate = FALSE,
  ...,
  mean,
  sd
)
```

Arguments

x	A numeric vector of values.
location	A numeric vector of the location parameter.
scale	A non-negative numeric vector of the scale parameter.
shape	A numeric vector of shape. Negative values result in leftward skew, while positive values result in rightward skew.
type	A string of the residual type. 'raw' for raw residuals 'dev' for deviance residuals and 'data' for the data.
simulate	A flag specifying whether to simulate residuals.
...	Unused.
mean	[Deprecated] A numeric vector of the location parameter. Described as "a numeric vector of the means" prior to v. 0.10.1. Will be removed in a future version.
sd	[Deprecated] A non-negative numeric vector of the scale parameter. Described as "a non-negative numeric vector of the standard deviations" prior to v. 0.10.1. Will be removed in a future version.

Value

An numeric vector of the corresponding residuals.

See Also

Other res_dist: [res_bern\(\)](#), [res_beta_binom\(\)](#), [res_binom\(\)](#), [res_gamma\(\)](#), [res_gamma_pois\(\)](#), [res_gamma_pois_zi\(\)](#), [res_lnorm\(\)](#), [res_multinom\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [res_skewlnorm\(\)](#), [res_student\(\)](#)

Examples

```
res_skewnorm(c(-2:2))
```

res_student	<i>Student's t Residuals</i>
-------------	------------------------------

Description

Student's t Residuals

Usage

```
res_student(x, mean = 0, sd = 1, theta = 0, type = "dev", simulate = FALSE)
```

Arguments

x	A non-negative whole numeric vector of values.
mean	A numeric vector of the means.
sd	A non-negative numeric vector of the standard deviations.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
type	A string of the residual type. 'raw' for raw residuals 'dev' for deviance residuals and 'data' for the data.
simulate	A flag specifying whether to simulate residuals.

Value

An numeric vector of the corresponding residuals.

See Also

Other res_dist: [res_bern\(\)](#), [res_beta_binom\(\)](#), [res_binom\(\)](#), [res_gamma\(\)](#), [res_gamma_pois\(\)](#), [res_gamma_pois_zi\(\)](#), [res_lnorm\(\)](#), [res_multinom\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [res_skewlnorm\(\)](#), [res_skewnorm\(\)](#)

Examples

```
res_student(c(1, 3.5, 4), mean = 6, sd = 0.5, theta = 1 / 3, type = "raw")
```

sens_beta

Adjust Beta Distribution Parameters for Sensitivity Analyses

Description

Expands ($sd_mult > 1$) or reduces ($sd_mult < 1$) the standard deviation of the Beta distribution. The Beta distribution has a maximum variance of $mean(x) * (1 - mean(x))$, where $mean(x) = \alpha / (\alpha + \beta)$. If the inputs produce a desired variance that is greater than the maximum possible variance, or provides alpha and/or beta parameters that are < 1 and thus push more probability weight towards extreme probability values, this function returns $\alpha = 1$ and $\beta = 1$ (the uniform distribution).

Usage

```
sens_beta(alpha, beta, sd_mult = 2)
```

Arguments

alpha	The first shape parameter of the beta distribution.
beta	The second shape parameter of the beta distribution.
sd_mult	A non-negative multiplier on the standard deviation of the distribution.

Value

A named list of the adjusted distribution’s parameters.

See Also

Other sens_dist: [sens_exp\(\)](#), [sens_gamma\(\)](#), [sens_gamma_pois\(\)](#), [sens_gamma_pois_zi\(\)](#), [sens_lnorm\(\)](#), [sens_neg_binom\(\)](#), [sens_norm\(\)](#), [sens_pois\(\)](#), [sens_skewlnorm\(\)](#), [sens_skewnorm\(\)](#), [sens_student\(\)](#)

Examples

```
sens_beta(10, 10, 2)
sens_beta(10, 10, 0.8)
```

sens_exp	<i>Adjust Exponential Distribution Parameters for Sensitivity Analyses</i>
----------	--

Description

Expands ($sd_mult > 1$) or reduces ($sd_mult < 1$) the standard deviation of the exponential distribution. Due to the parameterization of this distribution, adjusting the standard deviation necessarily changes the mean value.

Usage

```
sens_exp(rate, sd_mult = 2)
```

Arguments

- rate A non-negative numeric vector of rate.
- sd_mult A non-negative multiplier on the standard deviation of the distribution.

Value

A named list of the adjusted distribution’s parameters.

See Also

Other sens_dist: [sens_beta\(\)](#), [sens_gamma\(\)](#), [sens_gamma_pois\(\)](#), [sens_gamma_pois_zi\(\)](#), [sens_lnorm\(\)](#), [sens_neg_binom\(\)](#), [sens_norm\(\)](#), [sens_pois\(\)](#), [sens_skewlnorm\(\)](#), [sens_skewnorm\(\)](#), [sens_student\(\)](#)

Examples

```
sens_exp(10, 2)
sens_exp(10, 0.8)
```

sens_gamma	<i>Adjust Gamma Distribution Parameters for Sensitivity Analyses</i>
------------	--

Description

Expands (`sd_mult > 1`) or reduces (`sd_mult < 1`) the standard deviation of the Gamma distribution.

Usage

```
sens_gamma(shape, rate, sd_mult = 2)
```

Arguments

- shape A non-negative numeric vector of shape.
- rate A non-negative numeric vector of rate.
- sd_mult A non-negative multiplier on the standard deviation of the distribution.

Value

A named list of the adjusted distribution's parameters.

See Also

Other `sens_dist`: [sens_beta\(\)](#), [sens_exp\(\)](#), [sens_gamma_pois\(\)](#), [sens_gamma_pois_zi\(\)](#), [sens_lnorm\(\)](#), [sens_neg_binom\(\)](#), [sens_norm\(\)](#), [sens_pois\(\)](#), [sens_skewlnorm\(\)](#), [sens_skewnorm\(\)](#), [sens_student\(\)](#)

Examples

```
sens_gamma(10, 2, 2)
sens_gamma(10, 2, 0.2)
```

sens_gamma_pois	<i>Adjust Gamma-Poisson Distribution Parameters for Sensitivity Analyses</i>
-----------------	--

Description

Expands (`sd_mult > 1`) the standard deviation of the Negative Binomial distribution. This function does not currently have the option to reduce the standard deviation.

Usage

```
sens_gamma_pois(lambda, theta, sd_mult = 2)
```

Arguments

lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
sd_mult	A non-negative multiplier on the standard deviation of the distribution.

Value

A named list of the adjusted distribution's parameters.

See Also

Other sens_dist: [sens_beta\(\)](#), [sens_exp\(\)](#), [sens_gamma\(\)](#), [sens_gamma_pois_zi\(\)](#), [sens_lnorm\(\)](#), [sens_neg_binom\(\)](#), [sens_norm\(\)](#), [sens_pois\(\)](#), [sens_skewlnorm\(\)](#), [sens_skewnorm\(\)](#), [sens_student\(\)](#)

Examples

```
sens_gamma_pois(10, 0.1, 2)
```

sens_gamma_pois_zi	<i>Adjust Zero-Inflated Gamma-Poisson Distribution Parameters for Sensitivity Analyses</i>
--------------------	--

Description

Expands ($\text{sd_mult} > 1$) or reduces ($\text{sd_mult} < 1$) the standard deviation of the Zero-Inflated Gamma-Poisson distribution.

Usage

```
sens_gamma_pois_zi(lambda, theta, prob, sd_mult = 2)
```

Arguments

lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
prob	A numeric vector of values between 0 and 1 of the probability of success.
sd_mult	A non-negative multiplier on the standard deviation of the distribution.

Value

A named list of the adjusted distribution's parameters.

See Also

Other sens_dist: [sens_beta\(\)](#), [sens_exp\(\)](#), [sens_gamma\(\)](#), [sens_gamma_pois\(\)](#), [sens_lnorm\(\)](#), [sens_neg_binom\(\)](#), [sens_norm\(\)](#), [sens_pois\(\)](#), [sens_skewlnorm\(\)](#), [sens_skewnorm\(\)](#), [sens_student\(\)](#)

Examples

```
sens_gamma_pois_zi(10, 0.1, 0.3, 2)
```

sens_lnorm	<i>Adjust Log-Normal Distribution Parameters for Sensitivity Analysis</i>
------------	---

Description

Expands ($\text{sd_mult} > 1$) or reduces ($\text{sd_mult} < 1$) the standard deviation of the Log-Normal distribution. With high values of sdlog (i.e., > 9), and $\text{sd_mult} > 1$, the mean of the adjusted distribution can be expected to have a mean value that is very different from the original mean, however, the proportional difference in these values should not be very different.

Usage

```
sens_lnorm(meanlog, sdlog, sd_mult = 2)
```

Arguments

meanlog	A numeric vector of the means on the log scale.
sdlog	A non-negative numeric vector of the standard deviations on the log scale.
sd_mult	A non-negative multiplier on the standard deviation of the distribution.

Value

A named list of the adjusted distribution's parameters.

See Also

Other sens_dist: [sens_beta\(\)](#), [sens_exp\(\)](#), [sens_gamma\(\)](#), [sens_gamma_pois\(\)](#), [sens_gamma_pois_zi\(\)](#), [sens_neg_binom\(\)](#), [sens_norm\(\)](#), [sens_pois\(\)](#), [sens_skewlnorm\(\)](#), [sens_skewnorm\(\)](#), [sens_student\(\)](#)

Examples

```
sens_lnorm(0, 1, 2)
sens_lnorm(0, 1, 0.8)
```

sens_neg_binom	<i>Adjust Negative Binomial Distribution Parameters for Sensitivity Analyses</i>
----------------	--

Description

Expands ($\text{sd_mult} > 1$) the standard deviation of the Negative Binomial distribution. This function does not currently have the option to reduce the standard deviation.

Usage

```
sens_neg_binom(lambda, theta, sd_mult = 2)
```

Arguments

lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
sd_mult	A non-negative multiplier on the standard deviation of the distribution.

Value

A named list of the adjusted distribution's parameters.

See Also

Other sens_dist: [sens_beta\(\)](#), [sens_exp\(\)](#), [sens_gamma\(\)](#), [sens_gamma_pois\(\)](#), [sens_gamma_pois_zi\(\)](#), [sens_lnorm\(\)](#), [sens_norm\(\)](#), [sens_pois\(\)](#), [sens_skewlnorm\(\)](#), [sens_skewnorm\(\)](#), [sens_student\(\)](#)

Examples

```
sens_neg_binom(10, 0.1, 2)
```

sens_norm	<i>Adjust Normal Distribution Parameters for Sensitivity Analyses</i>
-----------	---

Description

Expands ($\text{sd_mult} > 1$) or reduces ($\text{sd_mult} < 1$) the standard deviation of the Normal distribution without changing the mean.

Usage

```
sens_norm(mean, sd, sd_mult = 2)
```

Arguments

mean	A numeric vector of the means.
sd	A non-negative numeric vector of the standard deviations.
sd_mult	A non-negative multiplier on the standard deviation of the distribution.

Value

A named list of the adjusted distribution's parameters.

See Also

Other sens_dist: [sens_beta\(\)](#), [sens_exp\(\)](#), [sens_gamma\(\)](#), [sens_gamma_pois\(\)](#), [sens_gamma_pois_zi\(\)](#), [sens_lnorm\(\)](#), [sens_neg_binom\(\)](#), [sens_pois\(\)](#), [sens_skewlnorm\(\)](#), [sens_skewnorm\(\)](#), [sens_student\(\)](#)

Examples

```
sens_norm(10, 3, 2)
sens_norm(10, 3, 0.8)
```

sens_pois

Adjust Poisson Distribution Parameters for Sensitivity Analyses

Description

Expands ($\text{sd_mult} > 1$) or reduces ($\text{sd_mult} < 1$) the standard deviation of the Poisson distribution. Due to the parameterization of this distribution, adjusting the standard deviation necessarily changes the mean value.

Usage

```
sens_pois(lambda, sd_mult = 2)
```

Arguments

lambda	A non-negative numeric vector of means.
sd_mult	A non-negative multiplier on the standard deviation of the distribution.

Value

A named list of the adjusted distribution's parameters.

See Also

Other sens_dist: [sens_beta\(\)](#), [sens_exp\(\)](#), [sens_gamma\(\)](#), [sens_gamma_pois\(\)](#), [sens_gamma_pois_zi\(\)](#), [sens_lnorm\(\)](#), [sens_neg_binom\(\)](#), [sens_norm\(\)](#), [sens_skewlnorm\(\)](#), [sens_skewnorm\(\)](#), [sens_student\(\)](#)

Examples

```
sens_pois(10, 2)
sens_pois(10, 0.8)
```

sens_skewlnorm	<i>Adjust Skew-Lognormal Distribution Parameters for Sensitivity Analyses</i>
----------------	---

Description

Expands (`scale_mult > 1`) or reduces (`scale_mult < 1`) the scale parameter of the Skew-Lognormal distribution while preserving its mean and `shapelog`. The adjustment is made on the natural scale (i.e. for `x`, not for `log(x)`), mirroring `sens_lnorm()`, to which it reduces when `shapelog = 0`.

Usage

```
sens_skewlnorm(
  locationlog,
  scalelog,
  shapelog,
  scale_mult = 2,
  ...,
  meanlog,
  sdlog,
  shape,
  sd_mult
)
```

Arguments

locationlog	A numeric vector of location parameters of <code>log(x)</code> .
scalelog	A non-negative numeric vector of scale parameters of <code>log(x)</code> .
shapelog	A numeric vector of shape parameters of <code>log(x)</code> . Negative values result in leftward skew, while positive values result in rightward skew.
scale_mult	A non-negative multiplier on the scale of the distribution.
...	Unused.
meanlog	[Deprecated] A numeric vector of location parameters of <code>log(x)</code> . Described as "a numeric vector of the means on the log scale" prior to v. 0.10.1. Will be removed in a future version.
sdlog	[Deprecated] A non-negative numeric vector of scale parameters of <code>log(x)</code> . Described as "a non-negative numeric vector of the standard deviations on the log scale" prior to v. 0.10.1. Will be removed in a future version.
shape	[Deprecated] A numeric vector of shape parameters of <code>log(x)</code> . Will be removed in a future version.
sd_mult	[Deprecated] A non-negative multiplier on the scale of the distribution.

Value

A named list of the adjusted distribution's parameters.

See Also

Other sens_dist: [sens_beta\(\)](#), [sens_exp\(\)](#), [sens_gamma\(\)](#), [sens_gamma_pois\(\)](#), [sens_gamma_pois_zi\(\)](#), [sens_lnorm\(\)](#), [sens_neg_binom\(\)](#), [sens_norm\(\)](#), [sens_pois\(\)](#), [sens_skewnorm\(\)](#), [sens_student\(\)](#)

Examples

```
sens_skewlnorm(0, 1, 2, 2)
sens_skewlnorm(0, 1, 2, 0.8)
```

sens_skewnorm	<i>Adjust Skew Normal Distribution Parameters for Sensitivity Analyses</i>
---------------	--

Description

Expands ($\text{scale_mult} > 1$) or reduces ($\text{scale_mult} < 1$) the scale parameter of the Skew Normal distribution without changing the mean.

Usage

```
sens_skewnorm(location, scale, shape, scale_mult = 2, ..., mean, sd, sd_mult)
```

Arguments

location	A numeric vector of the location parameter.
scale	A non-negative numeric vector of the scale parameter.
shape	A numeric vector of shape. Negative values result in leftward skew, while positive values result in rightward skew.
scale_mult	A non-negative multiplier on the scale of the distribution.
...	Unused.
mean	[Deprecated] A numeric vector of the location parameter. Described as "a numeric vector of the means" prior to v. 0.10.1. Will be removed in a future version.
sd	[Deprecated] A non-negative numeric vector of the scale parameter. Described as "a non-negative numeric vector of the standard deviations" prior to v. 0.10.1. Will be removed in a future version.
sd_mult	[Deprecated] A non-negative multiplier on the scale of the distribution.

Value

A named list of the adjusted distribution's parameters.

See Also

Other sens_dist: [sens_beta\(\)](#), [sens_exp\(\)](#), [sens_gamma\(\)](#), [sens_gamma_pois\(\)](#), [sens_gamma_pois_zi\(\)](#), [sens_lnorm\(\)](#), [sens_neg_binom\(\)](#), [sens_norm\(\)](#), [sens_pois\(\)](#), [sens_skewlnorm\(\)](#), [sens_student\(\)](#)

Examples

```
sens_skewnorm(10, 3, -1, 2)
sens_skewnorm(10, 3, 3, 0.8)
```

sens_student	<i>Adjust Student's t Distribution Parameters for Sensitivity Analyses</i>
--------------	--

Description

Expands (`sd_mult > 1`) or reduces (`sd_mult < 1`) the standard deviation of the Student's t distribution. Because the variance of this distribution is not defined for every degree of freedom, the adjustment to the standard deviation is approximate, and the mean of the adjusted distribution can be expected to have shifted.

Usage

```
sens_student(mean, sd, theta, sd_mult = 2)
```

Arguments

mean	A numeric vector of the means.
sd	A non-negative numeric vector of the standard deviations.
theta	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
sd_mult	A non-negative multiplier on the standard deviation of the distribution.

Value

A named list of the adjusted distribution's parameters.

See Also

Other sens_dist: [sens_beta\(\)](#), [sens_exp\(\)](#), [sens_gamma\(\)](#), [sens_gamma_pois\(\)](#), [sens_gamma_pois_zi\(\)](#), [sens_lnorm\(\)](#), [sens_neg_binom\(\)](#), [sens_norm\(\)](#), [sens_pois\(\)](#), [sens_skewlnorm\(\)](#), [sens_skewnorm\(\)](#)

Examples

```
sens_student(10, 3, 0.1, 2)
sens_student(10, 3, 0.1, 0.8)
```

sextreme	<i>Extreme Surprisal</i>
----------	--------------------------

Description

Calculates the surprisal (in bits) that a cumulative distribution function probability is at least that extreme. **[Deprecated]**

Usage

```
sextreme(x, directional = FALSE)
```

Arguments

x	A numeric vector of values between 0 and 1.
directional	A flag specifying whether probabilities less than 0.5 should be returned as negative values.

Value

A numeric vector of surprisal values.

See Also

Other residuals: [pextreme\(\)](#)

Examples

```
sextreme(seq(0.1, 0.9, by = 0.1))
sextreme(seq(0.1, 0.9, by = 0.1), directional = TRUE)
```

skewness	<i>Skewness</i>
----------	-----------------

Description

Skewness

Usage

```
skewness(x, na_rm = FALSE)
```

Arguments

x	A numeric object of MCMC values.
na_rm	A flag specifying whether to remove missing values.

Value

A number.

See Also

Other summary: [direction\(\)](#), [directional_information\(\)](#), [kurtosis\(\)](#), [lower\(\)](#), [probability_direction\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr_mean\(\)](#), [xtr_median\(\)](#), [xtr_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

Examples

```
skewness(1:10)
```

step	<i>Step</i>
------	-------------

Description

Step

Usage

```
step(x)
```

Arguments

x A numeric atomic object.

Value

A logical value.

See Also

Other translations: [exp10\(\)](#), [exp2\(\)](#), [fabs\(\)](#), [ilog\(\)](#), [ilog10\(\)](#), [ilog2\(\)](#), [ilogit\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [log10<-\(\)](#), [log2<-\(\)](#), [log<-\(\)](#), [logit\(\)](#), [logit<-\(\)](#), [phi\(\)](#), [pow\(\)](#)

Examples

```
step(1)
```

svalue

*Surprisal Value***Description**

The **surprisal value** (Greenland 2019) is a probability expressed in terms of how many consecutive heads would have to be thrown on a fair coin in a single attempt to achieve the same probability: $-\log_2(p)$, where p is the p-value of interest. See the details section for some examples.

Usage

```
svalue(x, ..., side = "both", threshold = 0, skeptical = TRUE, na_rm = FALSE)

p2svalue(p)
```

Arguments

<code>x</code>	A numeric object of MCMC values.
<code>...</code>	Unused.
<code>side</code>	A character indicating whether to calculate s-values using p-values for the left tail ("left"), right tail ("right"), or both tails ("both"; default).
<code>threshold</code>	A number of the threshold value.
<code>skeptical</code>	A flag specifying whether or not to add one sample to the empty side of the threshold when 100% of samples are on one side. Avoids zero p-values and infinite s-values, and also imposes stronger bounds on directional information than $[-n, n]$, which assume the MCMC samples are independent and representative.
<code>na_rm</code>	A flag specifying whether to remove missing values.
<code>p</code>	A numeric vector of probabilities.

Details

A near-certain event has an s-value near 0 because it is similar to getting 0 successful coin flips out of 0 tosses, which is certain and unsurprising.

An event with a probability of 0.5 is as surprising as getting a successful coin toss.

A near-impossible event has a very large s-value because its occurrence would be extremely surprising, like observing many consecutive successes on a fair coin.

When `skeptical = TRUE` (default), a ceiling of $\log_2(n+1)$ is applied to the s-value to avoid s-values of Inf when all samples are on one side of the threshold. When `skeptical = FALSE`, s-values of Inf are allowed.

Value

A non-negative number. If `x` has NA values but `na_rm` is FALSE, returns `NA_real`.

Functions

- `svalue()`: Calculate an s-value from a posterior distribution.
- `p2svalue()`: Calculate an s-value from a vector of probabilities.

References

Greenland, S. 2019. Valid P-Values Behave Exactly as They Should: Some Misleading Criticisms of P-Values and Their Resolution With S-Values. *The American Statistician* 73(sup1): 106–114. doi:10.1080/00031305.2018.1529625.

See Also

Other summary: `direction()`, `directional_information()`, `kurtosis()`, `lower()`, `probability_direction()`, `pvalue()`, `pzeros()`, `skewness()`, `upper()`, `variance()`, `xtr_mean()`, `xtr_median()`, `xtr_sd()`, `zeros()`, `zscore()`

Examples

```
svalue(as.numeric(0:100))
svalue(as.numeric(0:100), side = "left")
svalue(as.numeric(0:100), side = "right")
svalue(rnorm(1e4, mean = 1), side = "left")
svalue(rnorm(1e4, mean = 1), side = "right")
svalue(rep(1, 10)) # skeptical = TRUE (default) avoids Inf
svalue(rep(1, 10), skeptical = FALSE) # skeptical = FALSE allows Inf

p2svalue(seq(0, 1, by = 0.1))
```

upper	<i>Upper Credible Limit</i>
-------	-----------------------------

Description

Calculates the quantile-based upper credible limit.

Usage

```
upper(x, conf_level = 0.95, na_rm = FALSE)
```

Arguments

- | | |
|------------|---|
| x | A numeric vector of MCMC values. |
| conf_level | A numeric scalar between 0 and 1 specifying the confidence level. |
| na_rm | A flag specifying whether to remove missing values. |

Details

By default it returns the 95% credible limit which corresponds to the 97.5% quantile.

Value

A number.

See Also

Other summary: [direction\(\)](#), [directional_information\(\)](#), [kurtosis\(\)](#), [lower\(\)](#), [probability_direction\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [variance\(\)](#), [xtr_mean\(\)](#), [xtr_median\(\)](#), [xtr_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

Examples

```
upper(as.numeric(0:100))
```

variance	<i>Variance</i>
----------	-----------------

Description

Variance

Usage

```
variance(x, na_rm = FALSE)
```

Arguments

- x A numeric object of MCMC values.
- na_rm A flag specifying whether to remove missing values.

Value

A number.

See Also

Other summary: [direction\(\)](#), [directional_information\(\)](#), [kurtosis\(\)](#), [lower\(\)](#), [probability_direction\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [xtr_mean\(\)](#), [xtr_median\(\)](#), [xtr_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

Examples

```
variance(1:10)
```

`xtr_ci`*Credible Intervals*

Description

Calculates Bayesian credible intervals (CI) using one of the available methods:

- "HDI": highest density interval (see `xtr_ci_hdi()`),
- "ETI": equal tailed intervals (see `xtr_ci_eti()`).

Usage

```
xtr_ci(x, level = 0.95, ..., type = "HDI", na_rm = FALSE)
```

Arguments

<code>x</code>	A numeric vector of MCMC samples.
<code>level</code>	A number > 0 and ≤ 1 specifying the probability coverage of the interval.
<code>...</code>	Currently unused.
<code>type</code>	A string indicating which type of CI to return. Currently allows Highest Density Intervals ("HDI"; default) and Equal-Tailed Intervals ("ETI").
<code>na_rm</code>	A flag indicating whether to remove missing values.

Value

A [data.frame](#) of the lower and upper limits for the credible interval. Note that the interval is not guaranteed to be one-sided or two-sided.

See Also

[mcmcrr::coef](#)

Examples

```
xtr_ci(rnorm(1e4), type = "HDI")
```

xtr_ci_eti	<i>Equal-Tailed Interval</i>
------------	------------------------------

Description

Calculates Bayesian credible intervals using the equal-tailed interval (ETI), i.e., the CI such that the left and right tails outside the CI have the same coverage.

Usage

```
xtr_ci_eti(x, level = 0.95, ..., na_rm = FALSE)
```

Arguments

x	A numeric vector of MCMC samples.
level	A number > 0 and <= 1 specifying the probability coverage of the interval.
...	Currently unused.
na_rm	A flag indicating whether to remove missing values.

Details

The interval is guaranteed to be two-sided, unlike `[xtr_ci_hdi()]`. Does not return integer outputs even if the input data are integers, unlike `xtr_ci_hdi()`. The interval limits are always real (double) numeric values.

Value

A [data.frame](#) of the lower and upper limits for the credible interval.

See Also

[xtr_ci\(\)](#) and [xtr_ci_hdi\(\)](#)

Examples

```
xtr_ci_eti(rnorm(1e4))
```

xtr_ci_hdi	<i>Highest Density Interval</i>
------------	---------------------------------

Description

Calculates Bayesian credible intervals using the highest density interval (HDI), i.e., the narrowest CI with the specified minimum coverage.

Usage

```
xtr_ci_hdi(x, level = 0.95, ..., na_rm = FALSE)
```

Arguments

x	A numeric vector of MCMC samples.
level	A number > 0 and <= 1 specifying the probability coverage of the interval.
...	Currently unused.
na_rm	A flag indicating whether to remove missing values.

Value

A [data.frame](#) of the lower and upper limits for the credible interval. Note that the interval is not guaranteed to be one-sided or two-sided. Returns integer limits if the input data are integers and double otherwise.

See Also

[xtr_ci\(\)](#) and [xtr_ci_eti\(\)](#)

Examples

```
xtr_ci_hdi(1:10, level = 0.1) # only 10% of values inside
xtr_ci_hdi(1:10, level = 0.2) # only 20% of values inside
xtr_ci_hdi(1:10, level = 0.2 + 0.01) # at least 20.1% of values inside
xtr_ci_hdi(1:100) # inclusive interval [3, 98] with 95% of values inside
```

xtr_mean	Mean
----------	------

Description

Mean

Usage

```
xtr_mean(x, na_rm = FALSE)
```

Arguments

x	A numeric object of MCMC values.
na_rm	A flag specifying whether to remove missing values.

Value

A number.

See Also

Other summary: [direction\(\)](#), [directional_information\(\)](#), [kurtosis\(\)](#), [lower\(\)](#), [probability_direction\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr_median\(\)](#), [xtr_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

Examples

```
xtr_mean(1:10)
```

xtr_median	Median
------------	--------

Description

Median

Usage

```
xtr_median(x, na_rm = FALSE)
```

Arguments

x	A numeric object of MCMC values.
na_rm	A flag specifying whether to remove missing values.

Value

A number.

See Also

Other summary: [direction\(\)](#), [directional_information\(\)](#), [kurtosis\(\)](#), [lower\(\)](#), [probability_direction\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr_mean\(\)](#), [xtr_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

Examples

```
xtr_mean(1:10)
```

xtr_rope	<i>Region of Practical Equivalence</i>
----------	--

Description

Calculates the proportion of the posterior (or credible interval if `level < 1`) that falls within the region of practical equivalence (ROPE) to a threshold.

Usage

```
xtr_rope(  
  x,  
  threshold = 0,  
  interval = c(-0.1, 0.1),  
  ...,  
  level = 1,  
  type = "HDI",  
  na_rm = FALSE  
)
```

Arguments

x	A numeric vector of MCMC samples.
threshold	A number specifying the center of the ROPE.
interval	A numeric vector of length 2 to be added to threshold to calculate the ROPE. Generally, <code>threshold[1] == - threshold[2]</code> and <code>threshold[1] < 0</code> should both be true.
...	Currently unused.
level	A number > 0 and <= 1 specifying the probability coverage of the interval to use. The default of 1 uses the full posterior.
type	A string indicating which type of CI to return. Currently allows Highest Density Intervals ("HDI"; default) and Equal-Tailed Intervals ("ETI").
na_rm	A flag indicating whether to remove missing values.

Details

The ROPE is calculated as threshold + interval. Note that the default is not appropriate for all models, since the interval is sensitive to unit conversions.

Value

A number indicating the proportion of the posterior (or credible interval) within the region of practical equivalence.

See Also

[xtr_ci](#)

Examples

```
xtr_rope(c(-Inf, -1, -0.1, 0.1, 1, Inf))
xtr_rope(rnorm(1e4))
```

xtr_sd	<i>Standard Deviation</i>
--------	---------------------------

Description

Standard Deviation

Usage

```
xtr_sd(x, na_rm = FALSE)
```

Arguments

x	A numeric object of MCMC values.
na_rm	A flag specifying whether to remove missing values.

Value

A number.

See Also

Other summary: [direction\(\)](#), [directional_information\(\)](#), [kurtosis\(\)](#), [lower\(\)](#), [probability_direction\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr_mean\(\)](#), [xtr_median\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

Examples

```
xtr_sd(1:10)
```

zeros	<i>Zeros</i>
-------	--------------

Description

The number of zeros in an numeric object.

Usage

```
zeros(x, na_rm = FALSE)
```

Arguments

- x A numeric object of MCMC values.
- na_rm A flag specifying whether to remove missing values.

Value

A non-negative integer.

See Also

Other summary: [direction\(\)](#), [directional_information\(\)](#), [kurtosis\(\)](#), [lower\(\)](#), [probability_direction\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr_mean\(\)](#), [xtr_median\(\)](#), [xtr_sd\(\)](#), [zscore\(\)](#)

Examples

```
zeros(c(0:2))
```

zscore	<i>Z-Score</i>
--------	----------------

Description

The Bayesian z-score is here defined as the number of standard deviations from the mean estimate to zero.

Usage

```
zscore(x, na_rm = FALSE)
```

Arguments

- x A numeric object of MCMC values.
- na_rm A flag specifying whether to remove missing values.

Value

A number.

See Also

Other summary: [direction\(\)](#), [directional_information\(\)](#), [kurtosis\(\)](#), [lower\(\)](#), [probability_direction\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr_mean\(\)](#), [xtr_median\(\)](#), [xtr_sd\(\)](#), [zeros\(\)](#)

Examples

```
zscore(as.numeric(0:100))
```

Index

- * **bern**
 - dbern, 9
- * **dev_dist # make live when complete**
 - dev_gamma_pois_zi, 14
- * **dev_dist**
 - dev_bern, 10
 - dev_beta_binom, 11
 - dev_binom, 12
 - dev_gamma, 12
 - dev_gamma_pois, 13
 - dev_lnorm, 15
 - dev_multinom, 15
 - dev_neg_binom, 17
 - dev_norm, 17
 - dev_pois, 18
 - dev_pois_zi, 19
 - dev_skewlnorm, 20
 - dev_skewnorm, 21
 - dev_student, 22
- * **fill**
 - fill_all, 31
 - fill_na, 32
- * **log_lik_dist**
 - log_lik_bern, 39
 - log_lik_beta, 40
 - log_lik_beta_binom, 41
 - log_lik_binom, 42
 - log_lik_exp, 43
 - log_lik_gamma, 43
 - log_lik_gamma_pois, 44
 - log_lik_gamma_pois_zi, 45
 - log_lik_lnorm, 46
 - log_lik_multinom, 47
 - log_lik_neg_binom, 48
 - log_lik_norm, 49
 - log_lik_pois, 50
 - log_lik_pois_zi, 50
 - log_lik_skewlnorm, 51
 - log_lik_skewnorm, 52
 - log_lik_student, 54
 - log_lik_unif, 55
- * **odds fun2**
 - log_odds_ratio2, 57
 - odds_ratio2, 66
- * **odds**
 - inv_odds, 37
 - log_odds, 55
 - log_odds<-, 57
 - log_odds_ratio, 56
 - odds, 65
 - odds<-, 67
 - odds_ratio, 65
- * **prob_dist**
 - prob_bern, 70
 - prob_beta, 70
 - prob_beta_binom, 71
 - prob_binom, 72
 - prob_exp, 73
 - prob_gamma, 73
 - prob_gamma_pois, 74
 - prob_gamma_pois_zi, 75
 - prob_lnorm, 75
 - prob_neg_binom, 76
 - prob_norm, 77
 - prob_pois, 78
 - prob_pois_zi, 78
 - prob_skewlnorm, 79
 - prob_skewnorm, 80
 - prob_student, 81
 - prob_unif, 82
- * **proportional fun2**
 - proportional_change2, 84
 - proportional_difference2, 86
- * **proportional**
 - proportional_change, 84
 - proportional_difference, 85
- * **quant_dist**
 - quant_bern, 88

- quant_beta, 89
- quant_binom, 90
- quant_exp, 90
- quant_gamma, 91
- quant_gamma_pois, 92
- quant_gamma_pois_zi, 92
- quant_lnorm, 93
- quant_neg_binom, 94
- quant_norm, 95
- quant_pois, 95
- quant_pois_zi, 96
- quant_skewlnorm, 97
- quant_skewnorm, 98
- quant_student, 99
- quant_unif, 99
- * **ran_dist**
 - ran_bern, 100
 - ran_beta_binom, 101
 - ran_binom, 102
 - ran_gamma, 102
 - ran_gamma_pois, 103
 - ran_gamma_pois_zi, 104
 - ran_lnorm, 104
 - ran_multinom, 105
 - ran_neg_binom, 106
 - ran_norm, 107
 - ran_pois, 108
 - ran_pois_zi, 108
 - ran_skewlnorm, 109
 - ran_skewnorm, 110
 - ran_student, 111
- * **res_dist**
 - res_bern, 112
 - res_beta_binom, 112
 - res_binom, 113
 - res_gamma, 114
 - res_gamma_pois, 115
 - res_gamma_pois_zi, 116
 - res_lnorm, 117
 - res_multinom, 117
 - res_neg_binom, 119
 - res_norm, 120
 - res_pois, 120
 - res_pois_zi, 121
 - res_skewlnorm, 122
 - res_skewnorm, 123
 - res_student, 124
- * **residuals**
 - pextreme, 68
 - sextreme, 135
- * **sens_dist**
 - sens_beta, 125
 - sens_exp, 126
 - sens_gamma, 127
 - sens_gamma_pois, 127
 - sens_gamma_pois_zi, 128
 - sens_lnorm, 129
 - sens_neg_binom, 130
 - sens_norm, 130
 - sens_pois, 131
 - sens_skewlnorm, 132
 - sens_skewnorm, 133
 - sens_student, 134
- * **skewlnorm**
 - dskewlnorm, 26
- * **skewnorm**
 - dskewnorm, 28
- * **summary**
 - direction, 23
 - directional_information, 24
 - kurtosis, 39
 - lower, 62
 - probability_direction, 82
 - pvalue, 86
 - pzeros, 88
 - skewness, 135
 - svalue, 137
 - upper, 138
 - variance, 139
 - xtr_mean, 143
 - xtr_median, 143
 - xtr_sd, 145
 - zeros, 146
 - zscore, 146
- * **translations**
 - exp10, 29
 - exp2, 30
 - fabs, 30
 - ilog, 34
 - ilog10, 35
 - ilog2, 35
 - ilogit, 36
 - inv_logit, 37
 - invlogit, 38
 - log10<-, 59
 - log2<-, 59

- log<-, 58
- logit, 60
- logit<-, 61
- phi, 68
- pow, 69
- step, 136

- abs, 31
- as_list, 5
- as_list_unnamed, 6

- chk_index, 7
- chk_indices, 7
- chk_pars, 8

- data.frame, 140–142
- dbern, 9
- dev_bern, 10
- dev_bern(), 11–22
- dev_beta_binom, 11
- dev_beta_binom(), 10, 12–22
- dev_binom, 12
- dev_binom(), 10, 11, 13–22
- dev_gamma, 12
- dev_gamma(), 10–12, 14–22
- dev_gamma_pois, 13
- dev_gamma_pois(), 10–13, 15–22
- dev_gamma_pois_zi, 14
- dev_lnorm, 15
- dev_lnorm(), 10–14, 16–22
- dev_multinom, 15
- dev_multinom(), 10–15, 17–22
- dev_neg_binom, 17
- dev_neg_binom(), 10–16, 18–22
- dev_norm, 17
- dev_norm(), 10–22
- dev_pois, 18
- dev_pois(), 10–22
- dev_pois_zi, 19
- dev_pois_zi(), 10–18, 20–22
- dev_skewlnorm, 20
- dev_skewlnorm(), 10–19, 21, 22
- dev_skewnorm, 21
- dev_skewnorm(), 10–20, 22
- dev_student, 22
- dev_student(), 10–21
- direction, 23
- direction(), 24, 25, 39, 62, 83, 87, 88, 136, 138, 139, 143–147

- directional_information, 24
- directional_information(), 23, 39, 62, 83, 87, 88, 136, 138, 139, 143–147
- dskewlnorm, 26
- dskewnorm, 28

- exp, 34, 58
- exp10, 29, 35, 59
- exp10(), 30, 31, 34–38, 58–61, 69, 136
- exp2, 30, 36, 60
- exp2(), 29, 31, 34–38, 58–61, 69, 136

- fabs, 30
- fabs(), 29, 30, 34–38, 58–61, 69, 136
- fill_all, 31
- fill_all(), 33
- fill_na, 32
- fill_na(), 32

- ilog, 34
- ilog(), 29–31, 35–38, 58–61, 69, 136
- ilog10, 35
- ilog10(), 29–31, 34, 36–38, 58–61, 69, 136
- ilog2, 35
- ilog2(), 29–31, 34–38, 58–61, 69, 136
- ilogit, 36
- ilogit(), 29–31, 34–38, 58–61, 69, 136
- inv_logit, 37
- inv_logit(), 29–31, 34–36, 38, 58–61, 69, 136
- inv_odds, 37
- inv_odds(), 56, 58, 65–67
- invlogit, 38
- invlogit(), 29–31, 34–37, 58–61, 69, 136

- kurtosis, 39
- kurtosis(), 23, 25, 62, 83, 87, 88, 136, 138, 139, 143–147

- log10<-, 59
- log2<-, 59
- log<-, 58
- log_lik_bern, 39
- log_lik_bern(), 40–46, 48–55
- log_lik_beta, 40
- log_lik_beta(), 40–46, 48–55
- log_lik_beta_binom, 41
- log_lik_beta_binom(), 40, 42–46, 48–55
- log_lik_binom, 42

- `log_lik_binom()`, 40, 41, 43–46, 48–55
- `log_lik_exp`, 43
- `log_lik_exp()`, 40–42, 44–46, 48–55
- `log_lik_gamma`, 43
- `log_lik_gamma()`, 40–43, 45, 46, 48–55
- `log_lik_gamma_pois`, 44
- `log_lik_gamma_pois()`, 40–46, 48–55
- `log_lik_gamma_pois_zi`, 45
- `log_lik_gamma_pois_zi()`, 40–46, 48–55
- `log_lik_lnorm`, 46
- `log_lik_lnorm()`, 40–45, 48–55
- `log_lik_multinom`, 47
- `log_lik_multinom()`, 16, 40–46, 49–55, 117
- `log_lik_neg_binom`, 48
- `log_lik_neg_binom()`, 40–46, 48–55
- `log_lik_norm`, 49
- `log_lik_norm()`, 40–46, 48–55
- `log_lik_pois`, 50
- `log_lik_pois()`, 40–46, 48, 49, 51–55
- `log_lik_pois_zi`, 50
- `log_lik_pois_zi()`, 40–46, 48–50, 52–55
- `log_lik_skewlnorm`, 51
- `log_lik_skewlnorm()`, 40–46, 48–51, 53–55
- `log_lik_skewnorm`, 52
- `log_lik_skewnorm()`, 40–46, 48–52, 54, 55
- `log_lik_student`, 54
- `log_lik_student()`, 40–46, 48–53, 55
- `log_lik_unif`, 55
- `log_lik_unif()`, 40–46, 48–54
- `log_odds`, 55
- `log_odds()`, 38, 56, 58, 65–67
- `log_odds<-`, 57
- `log_odds_ratio`, 56
- `log_odds_ratio()`, 38, 56, 58, 65–67
- `log_odds_ratio2`, 57
- `log_odds_ratio2()`, 66
- `logit`, 60
- `logit()`, 29–31, 34–38, 58–61, 69, 136
- `logit<-`, 61
- `lower`, 62
- `lower()`, 23, 25, 39, 83, 87, 88, 136, 138, 139, 143–147
-
- `mcmcrr::coef`, 140
-
- `numericise`, 62
- `numericize (numericise)`, 62
-
- `odds`, 65
-
- `odds()`, 38, 56, 58, 66, 67
- `odds<-`, 67
- `odds_ratio`, 65
- `odds_ratio()`, 38, 56, 58, 65, 67
- `odds_ratio2`, 66
- `odds_ratio2()`, 57
-
- `p2info (directional_information)`, 24
- `p2svalue (svalue)`, 137
- `par_pattern`, 67
- `pbern (dbern)`, 9
- `pextreme`, 68
- `pextreme()`, 135
- `phi`, 68
- `phi()`, 29–31, 34–38, 58–61, 69, 136
- `pow`, 69
- `pow()`, 29–31, 34–38, 58–61, 69, 136
- `prob_bern`, 70
- `prob_bern()`, 71–82
- `prob_beta`, 70
- `prob_beta()`, 70–82
- `prob_beta_binom`, 71
- `prob_beta_binom()`, 70–82
- `prob_binom`, 72
- `prob_binom()`, 70, 71, 73–82
- `prob_exp`, 73
- `prob_exp()`, 70–72, 74–82
- `prob_gamma`, 73
- `prob_gamma()`, 70–82
- `prob_gamma_pois`, 74
- `prob_gamma_pois()`, 70–82
- `prob_gamma_pois_zi`, 75
- `prob_gamma_pois_zi()`, 70–74, 76–82
- `prob_lnorm`, 75
- `prob_lnorm()`, 70–82
- `prob_neg_binom`, 76
- `prob_neg_binom()`, 70–82
- `prob_norm`, 77
- `prob_norm()`, 70–76, 78–82
- `prob_pois`, 78
- `prob_pois()`, 70–77, 79–82
- `prob_pois_zi`, 78
- `prob_pois_zi()`, 70–78, 80–82
- `prob_skewlnorm`, 79
- `prob_skewlnorm()`, 70–79, 81, 82
- `prob_skewnorm`, 80
- `prob_skewnorm()`, 70–82
- `prob_student`, 81
- `prob_student()`, 70–82

- prob_unif, 82
- prob_unif(), 70–81
- probability_direction, 82
- probability_direction(), 23, 25, 39, 62, 87, 88, 136, 138, 139, 143–147
- proportional_change, 84
- proportional_change(), 85
- proportional_change2, 84
- proportional_change2(), 86
- proportional_difference, 85
- proportional_difference(), 84
- proportional_difference2, 86
- proportional_difference2(), 85
- pskewlnorm (dskewlnorm), 26
- pskewnorm (dskewnorm), 28
- pvalue, 86
- pvalue(), 23, 25, 39, 62, 83, 88, 136, 138, 139, 143–147
- pzeros, 88
- pzeros(), 23, 25, 39, 62, 83, 87, 136, 138, 139, 143–147
- qbern (dbern), 9
- qskewlnorm (dskewlnorm), 26
- qskewnorm (dskewnorm), 28
- quant_bern, 88
- quant_bern(), 89–100
- quant_beta, 89
- quant_beta(), 89–100
- quant_binom, 90
- quant_binom(), 89, 91–100
- quant_exp, 90
- quant_exp(), 89–100
- quant_gamma, 91
- quant_gamma(), 89–100
- quant_gamma_pois, 92
- quant_gamma_pois(), 89–91, 93–100
- quant_gamma_pois_zi, 92
- quant_gamma_pois_zi(), 89–100
- quant_lnorm, 93
- quant_lnorm(), 89–100
- quant_neg_binom, 94
- quant_neg_binom(), 89–93, 95–100
- quant_norm, 95
- quant_norm(), 89–94, 96–100
- quant_pois, 95
- quant_pois(), 89–100
- quant_pois_zi, 96
- quant_pois_zi(), 89–100
- quant_skewlnorm, 97
- quant_skewlnorm(), 89–96, 98–100
- quant_skewnorm, 98
- quant_skewnorm(), 89–97, 99, 100
- quant_student, 99
- quant_student(), 89–98, 100
- quant_unif, 99
- quant_unif(), 89–99
- ran_bern, 100
- ran_bern(), 101–111
- ran_beta_binom, 101
- ran_beta_binom(), 100, 102–111
- ran_binom, 102
- ran_binom(), 100, 101, 103–111
- ran_gamma, 102
- ran_gamma(), 100–111
- ran_gamma_pois, 103
- ran_gamma_pois(), 100–111
- ran_gamma_pois_zi, 104
- ran_gamma_pois_zi(), 100–103, 105–111
- ran_lnorm, 104
- ran_lnorm(), 100–104, 106–111
- ran_multinom, 105
- ran_multinom(), 100–105, 107–111, 118
- ran_neg_binom, 106
- ran_neg_binom(), 100–111
- ran_norm, 107
- ran_norm(), 100–111
- ran_pois, 108
- ran_pois(), 100–107, 109–111
- ran_pois_zi, 108
- ran_pois_zi(), 100–108, 110, 111
- ran_skewlnorm, 109
- ran_skewlnorm(), 100–109, 111
- ran_skewnorm, 110
- ran_skewnorm(), 100–111
- ran_student, 111
- ran_student(), 100–111
- rbern (dbern), 9
- res_bern, 112
- res_bern(), 113–125
- res_beta_binom, 112
- res_beta_binom(), 112, 114–125
- res_binom, 113
- res_binom(), 112, 113, 115–125
- res_gamma, 114
- res_gamma(), 112–125
- res_gamma_pois, 115

- `res_gamma_pois()`, 112–125
- `res_gamma_pois_zi`, 116
- `res_gamma_pois_zi()`, 112–115, 117–125
- `res_lnorm`, 117
- `res_lnorm()`, 112–116, 118–125
- `res_multinom`, 117
- `res_multinom()`, 112–117, 119–125
- `res_neg_binom`, 119
- `res_neg_binom()`, 112–118, 120–125
- `res_norm`, 120
- `res_norm()`, 112–119, 121–125
- `res_pois`, 120
- `res_pois()`, 112–120, 122–125
- `res_pois_zi`, 121
- `res_pois_zi()`, 112–121, 123–125
- `res_skewlnorm`, 122
- `res_skewlnorm()`, 112–122, 124, 125
- `res_skewnorm`, 123
- `res_skewnorm()`, 112–123, 125
- `res_student`, 124
- `res_student()`, 112–124
- `rskewlnorm(dskewlnorm)`, 26
- `rskewnorm(dskewnorm)`, 28
-
- `sens_beta`, 125
- `sens_beta()`, 126–131, 133, 134
- `sens_exp`, 126
- `sens_exp()`, 126–131, 133, 134
- `sens_gamma`, 127
- `sens_gamma()`, 126, 128–131, 133, 134
- `sens_gamma_pois`, 127
- `sens_gamma_pois()`, 126, 127, 129–131, 133, 134
- `sens_gamma_pois_zi`, 128
- `sens_gamma_pois_zi()`, 126–131, 133, 134
- `sens_lnorm`, 129
- `sens_lnorm()`, 126–134
- `sens_neg_binom`, 130
- `sens_neg_binom()`, 126–129, 131, 133, 134
- `sens_norm`, 130
- `sens_norm()`, 126–131, 133, 134
- `sens_pois`, 131
- `sens_pois()`, 126–131, 133, 134
- `sens_skewlnorm`, 132
- `sens_skewlnorm()`, 126–131, 134
- `sens_skewnorm`, 133
- `sens_skewnorm()`, 126–131, 133, 134
- `sens_student`, 134
- `sens_student()`, 126–131, 133, 134
-
- `sextreme`, 135
- `sextreme()`, 68
- Skew-Normal, 26
- skewness, 135
- `skewness()`, 23, 25, 39, 62, 83, 87, 88, 138, 139, 143–147
- `stats::plogis()`, 36–38
- `stats::pnorm()`, 68
- `stats::qlogis()`, 60
- `step`, 136
- `step()`, 29–31, 34–38, 58–61, 69
- `svalue`, 137
- `svalue()`, 23, 25, 39, 62, 83, 87, 88, 136, 139, 143–147
-
- `upper`, 138
- `upper()`, 23, 25, 39, 62, 83, 87, 88, 136, 138, 139, 143–147
-
- variance, 139
- `variance()`, 23, 25, 39, 62, 83, 87, 88, 136, 138, 139, 143–147
- `vld_index(chk_index)`, 7
- `vld_indices(chk_indices)`, 7
- `vld_pars(chk_pars)`, 8
-
- `xtr_ci`, 140, 145
- `xtr_ci()`, 141, 142
- `xtr_ci_eti`, 141
- `xtr_ci_eti()`, 140, 142
- `xtr_ci_hdi`, 142
- `xtr_ci_hdi()`, 140, 141
- `xtr_mean`, 143
- `xtr_mean()`, 23, 25, 39, 62, 83, 87, 88, 136, 138, 139, 144–147
- `xtr_median`, 143
- `xtr_median()`, 23, 25, 39, 62, 83, 87, 88, 136, 138, 139, 143, 145–147
- `xtr_rope`, 144
- `xtr_sd`, 145
- `xtr_sd()`, 23, 25, 39, 62, 83, 87, 88, 136, 138, 139, 143, 144, 146, 147
-
- `zeros`, 146
- `zeros()`, 23, 25, 39, 62, 83, 87, 88, 136, 138, 139, 143–145, 147
- `zscore`, 146
- `zscore()`, 23, 25, 39, 62, 83, 87, 88, 136, 138, 139, 143–146