

Package ‘extras’

May 8, 2026

Title Helper Functions for Bayesian Analyses

Version 0.8.0

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, ggplot2, 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

RoxygenNote 7.3.2.9000

NeedsCompilation no

Author Nicole Hill [aut, cre] (ORCID: <<https://orcid.org/0000-0002-7623-2153>>),
Joe Thorley [aut] (ORCID: <<https://orcid.org/0000-0002-7683-4592>>),
Kirill Müller [ctb] (ORCID: <<https://orcid.org/0000-0002-1416-3412>>),
Nadine Hussein [ctb] (ORCID: <<https://orcid.org/0000-0003-4470-8361>>),
Poisson Consulting [cph, fnd]

Maintainer Nicole Hill <nicole@poissonconsulting.ca>

Repository CRAN

Date/Publication 2025-01-13 20:10:02 UTC

Contents

as_list	4
as_list_unnamed	5
chk_index	6
chk_indices	6
chk_pars	7
dbern	8
dev_bern	9
dev_beta_binom	10
dev_binom	11
dev_gamma	11
dev_gamma_pois	12
dev_gamma_pois_zi	13
dev_lnorm	14
dev_neg_binom	14
dev_norm	15
dev_pois	16
dev_pois_zi	17
dev_skewnorm	17
dev_student	18
dskewnorm	19
exp10	20
exp2	21
fabs	21
fill_all	22
fill_na	23
ilog	25
ilog10	26
ilog2	26
ilogit	27
invlogit	28
inv_logit	28
inv_odds	29
kurtosis	30
log10<-	30
log2<-	31
log<-	32
logit	33
logit<-	33
log_lik_bern	34
log_lik_beta_binom	35
log_lik_binom	36
log_lik_gamma	36
log_lik_gamma_pois	37
log_lik_gamma_pois_zi	38
log_lik_lnorm	38
log_lik_neg_binom	39

log_lik_norm	40
log_lik_pois	41
log_lik_pois_zi	41
log_lik_skewnorm	42
log_lik_student	43
log_odds	43
log_odds<-	44
log_odds_ratio	45
log_odds_ratio2	45
lower	46
numericise	47
odds	49
odds<-	50
odds_ratio	50
odds_ratio2	51
par_pattern	52
pextreme	52
phi	53
pow	53
proportional_change	54
proportional_change2	55
proportional_difference	55
proportional_difference2	56
pvalue	57
pzeros	58
ran_bern	58
ran_beta_binom	59
ran_binom	60
ran_gamma	61
ran_gamma_pois	61
ran_gamma_pois_zi	62
ran_lnorm	63
ran_neg_binom	64
ran_norm	64
ran_pois	65
ran_pois_zi	66
ran_skewnorm	66
ran_student	67
res_bern	68
res_beta_binom	69
res_binom	70
res_gamma	71
res_gamma_pois	71
res_gamma_pois_zi	72
res_lnorm	73
res_neg_binom	74
res_norm	75
res_pois	75

res_pois_zi	76
res_skewnorm	77
res_student	78
sens_beta	79
sens_exp	79
sens_gamma	80
sens_gamma_pois	81
sens_gamma_pois_zi	82
sens_lnorm	82
sens_neg_binom	83
sens_norm	84
sens_pois	85
sens_skewnorm	85
sens_student	86
sextreme	87
skewness	88
step	88
svalue	89
upper	90
variance	90
xtr_mean	91
xtr_median	92
xtr_sd	92
zeros	93
zscore	94

Index **95**

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))
```

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

Description

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

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`*Check Index*

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

<code>x</code>	An object.
<code>x_name</code>	A string of the name of object <code>x</code> or <code>NULL</code> .

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`*Check Indices*

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 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	<i>Bernoulli Deviances</i>
----------	----------------------------

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_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [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 `theta = 0`, the beta-binomial reverts to the binomial distribution. When `theta = 1` and `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

<code>x</code>	A non-negative whole numeric vector of values.
<code>size</code>	A non-negative whole numeric vector of the number of trials.
<code>prob</code>	A numeric vector of values between 0 and 1 of the probability of success.
<code>theta</code>	A non-negative numeric vector of the dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
<code>res</code>	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_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [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

<code>x</code>	A non-negative whole numeric vector of values.
<code>size</code>	A non-negative whole numeric vector of the number of trials.
<code>prob</code>	A numeric vector of values between 0 and 1 of the probability of success.
<code>res</code>	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 AlsoOther dev_dist: [dev_bern\(\)](#), [dev_beta_binom\(\)](#), [dev_gamma\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [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_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [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_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [dev_skewnorm\(\)](#), [dev_student\(\)](#)

Examples

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

dev_gamma_pois_zi *Zero-Inflated Gamma-Poisson Deviances*

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

Log-Normal Deviances

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_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [dev_skewnorm\(\)](#), [dev_student\(\)](#)

Examples

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

dev_neg_binom

Negative Binomial Deviances

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_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [dev_skewnorm\(\)](#), [dev_student\(\)](#)

Examples

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

dev_norm

Normal Deviances

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_neg_binom\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [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_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois_zi\(\)](#), [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_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_skewnorm\(\)](#), [dev_student\(\)](#)

Examples

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

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

Description

Skew Normal Deviances

Usage

```
dev_skewnorm(x, mean = 0, sd = 1, shape = 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.
shape	A numeric vector of shape.
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_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [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

Student's t Deviances

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_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#), [dev_pois_zi\(\)](#), [dev_skewnorm\(\)](#)

Examples

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

dskewnorm

Skew-Normal Distribution

Description

Skew-Normal Distribution

Usage

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

```
pskewnorm(q, mean = 0, sd = 1, shape = 0)
```

```
qskewnorm(p, mean = 0, sd = 1, shape = 0)
```

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

Arguments

x	A numeric vector of values.
mean	A numeric vector of the means.
sd	A non-negative numeric vector of the standard deviations.
shape	A numeric vector of values.
log	A flag specifying whether to return the log-transformed value.
q	A vector of quantiles.
p	A 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, mean = 0, sd = 1, shape = 0.1)
dskewnorm(x = -2:2, mean = 0, sd = 1, shape = -1)
qskewnorm(p = c(0.1, 0.4), mean = 0, sd = 1, shape = 0.1)
qskewnorm(p = c(0.1, 0.4), mean = 0, sd = 1, shape = -1)
pskewnorm(q = -2:2, mean = 0, sd = 1, shape = 0.1)
pskewnorm(q = -2:2, mean = 0, sd = 1, shape = -1)
rskewnorm(n = 3, mean = 0, sd = 1, shape = 0.1)
rskewnorm(n = 3, mean = 0, sd = 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 = "0", ...)
```

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))
```

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

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))
```

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

Inverse Odds

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))
```

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

Examples

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

log10<-	<i>Log Base 10 Transformation</i>
---------	-----------------------------------

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
```

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
```

logit	<i>Logistic Transformation</i>
-------	--------------------------------

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<-	<i>Logistic Transformation</i>
---------	--------------------------------

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
```

log_lik_bern

Bernoulli Log-Likelihood

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_binom\(\)](#), [log_lik_binom\(\)](#), [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_skewnorm\(\)](#), [log_lik_student\(\)](#)

Examples

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

log_lik_beta_binom *Beta-Binomial Log-Likelihood*

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, 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).
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_binom\(\)](#), [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_skewnorm\(\)](#), [log_lik_student\(\)](#)

Examples

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

log_lik_binom	<i>Binomial Log-Likelihood</i>
---------------	--------------------------------

Description

Binomial Log-Likelihood

Usage

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

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.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_beta_binom\(\)](#), [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_skewnorm\(\)](#), [log_lik_student\(\)](#)

Examples

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

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

Description

Gamma Log-Likelihood

Usage

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

Arguments

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

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_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_skewnorm\(\)](#), [log_lik_student\(\)](#)

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)
```

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).

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_gamma\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#)

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)
```

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.

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_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#)

Examples

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

log_lik_lnorm *Log-Normal Log-Likelihood*

Description

Log-Normal Log-Likelihood

Usage

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

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.

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_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#)

Examples

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

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)
```

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).

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_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#)

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)
```

Arguments

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

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_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_pois\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#)

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)
```

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.

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_gamma\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_gamma_pois_zi\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_skewnorm\(\)](#), [log_lik_student\(\)](#)

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)
```

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.

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_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_skewnorm\(\)](#), [log_lik_student\(\)](#)

Examples

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

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

Description

Skew Normal Log-Likelihood

Usage

```
log_lik_skewnorm(x, mean = 0, sd = 1, shape = 0)
```

Arguments

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

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_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_student\(\)](#)

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)
```

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).

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_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_skewnorm\(\)](#)

Examples

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

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<- *Inverse Log Odds Transformation*

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_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))
```

lower

Lower Credible Limit

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: [kurtosis\(\)](#), [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

Odds

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<-	<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
```

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

Odds Ratio2

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))
```

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	<i>Extreme Probability</i>
----------	----------------------------

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: [sxtreme\(\)](#)

Examples

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

phi	<i>Phi</i>
-----	------------

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)
```

proportional_change *Proportional Change*

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 *Proportional Change2*

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

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_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	<i>Bayesian P-Value</i>
--------	-------------------------

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). By default a p-value of 0.05 indicates that the 95% CRI just includes 0.

Usage

```
pvalue(x, threshold = 0, na_rm = FALSE)
```

Arguments

x	A numeric vector of MCMC values.
threshold	A number of the threshold value.
na_rm	A flag specifying whether to remove missing values.

Value

A number between 0 and 1.

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

Examples

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

pzeros	<i>Proportion of Zeros</i>
--------	----------------------------

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

Examples

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

ran_bern	<i>Bernoulli Random Samples</i>
----------	---------------------------------

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_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [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 `theta = 0`, the beta-binomial reverts to the binomial distribution. When `theta = 1` and `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

<code>n</code>	A non-negative whole number of the number of random samples to generate.
<code>size</code>	A non-negative whole numeric vector of the number of trials.
<code>prob</code>	A numeric vector of values between 0 and 1 of the probability of success.
<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_binom\(\)](#), [ran_gamma\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma_pois_zi\(\)](#), [ran_lnorm\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

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

ran_binom

Binomial Random Samples

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_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

```
ran_binom(10)
```

`ran_gamma`*Gamma Random Samples*

Description

Gamma Random Samples

Usage

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

Arguments

<code>n</code>	A non-negative whole number of the number of random samples to generate.
<code>shape</code>	A non-negative numeric vector of shape.
<code>rate</code>	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_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

```
ran_gamma(10)
```

`ran_gamma_pois`*Gamma-Poisson Random Samples*

Description

Gamma-Poisson Random Samples

Usage

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

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_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

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

ran_gamma_pois_zi *Zero-Inflated Gamma-Poisson Random Samples*

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_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [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

<code>n</code>	A non-negative whole number of the number of random samples to generate.
<code>meanlog</code>	A numeric vector of the means on the log scale.
<code>sdlog</code>	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_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

```
ran_lnorm(10)
```

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

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).

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_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

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

ran_norm	<i>Normal Random Samples</i>
----------	------------------------------

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_neg_binom\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [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_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois_zi\(\)](#), [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_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_skewnorm\(\)](#), [ran_student\(\)](#)

Examples

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

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

Description

Skew Normal Random Samples

Usage

```
ran_skewnorm(n = 1, mean = 0, sd = 1, shape = 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.
shape	A numeric vector of shape.

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_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [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_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#), [ran_pois_zi\(\)](#), [ran_skewnorm\(\)](#)

Examples

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

res_bern

Bernoulli Residuals

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_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [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 `theta = 0`, the beta-binomial reverts to the binomial distribution. When `theta = 1` and `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

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

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_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [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_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [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_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [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_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [res_skewnorm\(\)](#), [res_student\(\)](#)

Examples

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

res_gamma_pois_zi *Zero-Inflated Gamma-Poisson Residuals*

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_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [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_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [res_skewnorm\(\)](#), [res_student\(\)](#)

Examples

```
res_lnorm(10)
```

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

Description

Negative Binomial Residuals

Usage

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

Arguments

<code>x</code>	A non-negative whole numeric vector of values.
<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).
<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.

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_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [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_neg_binom\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [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_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois_zi\(\)](#), [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_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_skewnorm\(\)](#), [res_student\(\)](#)

Examples

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

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

Description

Skew Normal Residuals

Usage

```
res_skewnorm(x, mean = 0, sd = 1, shape = 0, 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.
shape	A numeric vector of shape.
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_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [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_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#), [res_pois_zi\(\)](#), [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_skewnorm\(\)](#), [sens_student\(\)](#)

Examples

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

sens_exp

Adjust Exponential Distribution Parameters for Sensitivity Analyses

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_skewnorm\(\)](#), [sens_student\(\)](#)

Examples

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

sens_gamma

Adjust Gamma Distribution Parameters for Sensitivity Analyses

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_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_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 ($sd_mult > 1$) or reduces ($sd_mult < 1$) 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_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 ($sd_mult > 1$) or reduces ($sd_mult < 1$) the standard deviation of the Log-Normal distribution. With high values of $sdlog$ (i.e., > 9), and $sd_mult > 1$, 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_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 ($sd_mult > 1$) 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_skewnorm\(\)](#), [sens_student\(\)](#)

Examples

```
sens_neg_binom(10, 0.1, 2)
```

sens_norm

Adjust Normal Distribution Parameters for Sensitivity Analyses

Description

Expands ($sd_mult > 1$) or reduces ($sd_mult < 1$) 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_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 ($sd_mult > 1$) or reduces ($sd_mult < 1$) 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_skewnorm\(\)](#), [sens_student\(\)](#)

Examples

```
sens_pois(10, 2)
sens_pois(10, 0.8)
```

 sens_skewnorm

Adjust Skew Normal Distribution Parameters for Sensitivity Analyses

Description

Expands ($sd_mult > 1$) or reduces ($sd_mult < 1$) the standard deviation of the Skew Normal distribution without changing the mean.

Usage

```
sens_skewnorm(mean, sd, shape, sd_mult = 2)
```

Arguments

mean	A numeric vector of the means.
sd	A non-negative numeric vector of the standard deviations.
shape	A non-negative numeric vector of shape.
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_student\(\)](#)

Examples

```
sens_skewnorm(10, 3, -1, 2)
sens_skewnorm(10, 3, 3, 0.8)
```

sens_student

Adjust Student's t Distribution Parameters for Sensitivity Analyses

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_skewnorm\(\)](#)

Examples

```
sens_student(10, 3, 0.1, 2)
sens_student(10, 3, 0.1, 0.8)
```

sextreme

Extreme Surprisal

Description

Calculates the surprisal (in bits) that a cumulative distribution function probability is at least that extreme. **[Deprecated]**

Usage

```
sextreme(x, directional = FALSE)
```

Arguments

<code>x</code>	A numeric vector of values between 0 and 1.
<code>directional</code>	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: [kurtosis\(\)](#), [lower\(\)](#), [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	<i>Surprisal Value</i>
--------	------------------------

Description

The surprisal value (Greenland 2019) is the [pvalue](#) 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.

Usage

```
svalue(x, threshold = 0, na_rm = FALSE)
```

Arguments

<code>x</code>	A numeric object of MCMC values.
<code>threshold</code>	A number of the threshold value.
<code>na_rm</code>	A flag specifying whether to remove missing values.

Value

A non-negative number.

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](https://doi.org/10.1080/00031305.2018.1529625).

See Also

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

Examples

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

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: [kurtosis\(\)](#), [lower\(\)](#), [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: [kurtosis\(\)](#), [lower\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [xtr_mean\(\)](#), [xtr_median\(\)](#), [xtr_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

Examples

```
variance(1:10)
```

xtr_mean	<i>Mean</i>
----------	-------------

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: [kurtosis\(\)](#), [lower\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr_median\(\)](#), [xtr_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

Examples

```
xtr_mean(1:10)
```

xtr_median	<i>Median</i>
------------	---------------

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: [kurtosis\(\)](#), [lower\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr_mean\(\)](#), [xtr_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

Examples

```
xtr_mean(1:10)
```

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: [kurtosis\(\)](#), [lower\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr_mean\(\)](#), [xtr_median\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

Examples

```
xtr_sd(1:10)
```

zeros

Zeros

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: [kurtosis\(\)](#), [lower\(\)](#), [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: [kurtosis\(\)](#), [lower\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr_mean\(\)](#), [xtr_median\(\)](#), [xtr_sd\(\)](#), [zeros\(\)](#)

Examples

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

Index

- * **bern**
 - dbern, 8
- * **dev_dist # make live when complete**
 - dev_gamma_pois_zi, 13
- * **dev_dist**
 - dev_bern, 9
 - dev_beta_binom, 10
 - dev_binom, 11
 - dev_gamma, 11
 - dev_gamma_pois, 12
 - dev_lnorm, 14
 - dev_neg_binom, 14
 - dev_norm, 15
 - dev_pois, 16
 - dev_pois_zi, 17
 - dev_skewnorm, 17
 - dev_student, 18
- * **fill**
 - fill_all, 22
 - fill_na, 23
- * **log_lik_dist**
 - log_lik_bern, 34
 - log_lik_beta_binom, 35
 - log_lik_binom, 36
 - log_lik_gamma, 36
 - log_lik_gamma_pois, 37
 - log_lik_gamma_pois_zi, 38
 - log_lik_lnorm, 38
 - log_lik_neg_binom, 39
 - log_lik_norm, 40
 - log_lik_pois, 41
 - log_lik_pois_zi, 41
 - log_lik_skewnorm, 42
 - log_lik_student, 43
- * **odds fun2**
 - log_odds_ratio2, 45
 - odds_ratio2, 51
- * **odds**
 - inv_odds, 29
 - log_odds, 43
 - log_odds<-, 44
 - log_odds_ratio, 45
 - odds, 49
 - odds<-, 50
 - odds_ratio, 50
- * **proportional fun2**
 - proportional_change2, 55
 - proportional_difference2, 56
- * **proportional**
 - proportional_change, 54
 - proportional_difference, 55
- * **ran_dist**
 - ran_bern, 58
 - ran_beta_binom, 59
 - ran_binom, 60
 - ran_gamma, 61
 - ran_gamma_pois, 61
 - ran_gamma_pois_zi, 62
 - ran_lnorm, 63
 - ran_neg_binom, 64
 - ran_norm, 64
 - ran_pois, 65
 - ran_pois_zi, 66
 - ran_skewnorm, 66
 - ran_student, 67
- * **res_dist**
 - res_bern, 68
 - res_beta_binom, 69
 - res_binom, 70
 - res_gamma, 71
 - res_gamma_pois, 71
 - res_gamma_pois_zi, 72
 - res_lnorm, 73
 - res_neg_binom, 74
 - res_norm, 75
 - res_pois, 75
 - res_pois_zi, 76
 - res_skewnorm, 77

- res_student, 78
- * **residuals**
 - pextreme, 52
 - sextreme, 87
- * **sens_dist**
 - sens_beta, 79
 - sens_exp, 79
 - sens_gamma, 80
 - sens_gamma_pois, 81
 - sens_gamma_pois_zi, 82
 - sens_lnorm, 82
 - sens_neg_binom, 83
 - sens_norm, 84
 - sens_pois, 85
 - sens_skewnrm, 85
 - sens_student, 86
- * **skewnrm**
 - dskewnrm, 19
- * **summary**
 - kurtosis, 30
 - lower, 46
 - pvalue, 57
 - pzeros, 58
 - skewness, 88
 - svalue, 89
 - upper, 90
 - variance, 90
 - xtr_mean, 91
 - xtr_median, 92
 - xtr_sd, 92
 - zeros, 93
 - zscore, 94
- * **translations**
 - exp10, 20
 - exp2, 21
 - fabs, 21
 - ilog, 25
 - ilog10, 26
 - ilog2, 26
 - ilogit, 27
 - inv_logit, 28
 - invlogit, 28
 - log10<-, 30
 - log2<-, 31
 - log<-, 32
 - logit, 33
 - logit<-, 33
 - phi, 53
 - pow, 53
 - step, 88
- abs, 22
- as_list, 4
- as_list_unnamed, 5
- chk_index, 6
- chk_indices, 6
- chk_pars, 7
- dbern, 8
- dev_bern, 9, 10–19
- dev_beta_binom, 9, 10, 11–19
- dev_binom, 9, 10, 11, 12–19
- dev_gamma, 9–11, 11, 13–19
- dev_gamma_pois, 9–12, 12, 14–19
- dev_gamma_pois_zi, 13
- dev_lnorm, 9–13, 14, 15–19
- dev_neg_binom, 9–14, 14, 16–19
- dev_norm, 9–15, 15, 16–19
- dev_pois, 9–16, 16, 17–19
- dev_pois_zi, 9–16, 17, 18, 19
- dev_skewnrm, 9–17, 17, 19
- dev_student, 9–18, 18
- dskewnrm, 19
- exp, 25, 32
- exp10, 20, 21, 22, 25–29, 31–34, 53, 54, 89
- exp2, 20, 21, 22, 25–29, 31–34, 53, 54, 89
- fabs, 20, 21, 21, 25–29, 31–34, 53, 54, 89
- fill_all, 22, 24
- fill_na, 23, 23
- ilog, 20–22, 25, 26–29, 31–34, 53, 54, 89
- ilog10, 20–22, 25, 26, 27–29, 31–34, 53, 54, 89
- ilog2, 20–22, 25, 26, 26, 27–29, 31–34, 53, 54, 89
- ilogit, 20–22, 25–27, 27, 28, 29, 31–34, 53, 54, 89
- inv_logit, 20–22, 25–28, 28, 31–34, 53, 54, 89
- inv_odds, 29, 44, 45, 49–51
- invlogit, 20–22, 25–27, 28, 29, 31–34, 53, 54, 89
- kurtosis, 30, 46, 57, 58, 88–94

- log10<-, 30
- log2<-, 31
- log<-, 32
- log_lik_bern, 34, 35–43
- log_lik_beta_binom, 34, 35, 36–43
- log_lik_binom, 34, 35, 36, 37–43
- log_lik_gamma, 34–36, 36, 37–43
- log_lik_gamma_pois, 34–37, 37, 38–43
- log_lik_gamma_pois_zi, 34–37, 38, 39–43
- log_lik_lnorm, 34–38, 38, 39–43
- log_lik_neg_binom, 34–39, 39, 40–43
- log_lik_norm, 34–39, 40, 41–43
- log_lik_pois, 34–40, 41, 42, 43
- log_lik_pois_zi, 34–41, 41, 42, 43
- log_lik_skewnorm, 34–42, 42, 43
- log_lik_student, 34–42, 43
- log_odds, 29, 43, 44, 45, 49–51
- log_odds<-, 44
- log_odds_ratio, 29, 44, 45, 49–51
- log_odds_ratio2, 45, 51
- logit, 20–22, 25–29, 31, 32, 33, 34, 53, 54, 89
- logit<-, 33
- lower, 30, 46, 57, 58, 88–94

- numericise, 47
- numericize (numericise), 47

- odds, 29, 44, 45, 49, 50, 51
- odds<-, 50
- odds_ratio, 29, 44, 45, 49, 50, 50
- odds_ratio2, 46, 51

- par_pattern, 52
- pbern (dbern), 8
- pextreme, 52, 87
- phi, 20–22, 25–29, 31–34, 53, 54, 89
- pow, 20–22, 25–29, 31–34, 53, 53, 89
- proportional_change, 54, 56
- proportional_change2, 55, 56
- proportional_difference, 54, 55
- proportional_difference2, 55, 56
- pskewnorm (dskewnorm), 19
- pvalue, 30, 46, 57, 58, 88–94
- pzeros, 30, 46, 57, 58, 88–94

- qbern (dbern), 8
- qskewnorm (dskewnorm), 19

- ran_bern, 58, 59–68
- ran_beta_binom, 59, 59, 60–68
- ran_binom, 59, 60, 61–68
- ran_gamma, 59, 60, 61, 62–68
- ran_gamma_pois, 59–61, 61, 63–68
- ran_gamma_pois_zi, 59–62, 62, 63–68
- ran_lnorm, 59–63, 63, 64–68
- ran_neg_binom, 59–63, 64, 65–68
- ran_norm, 59–64, 64, 65–68
- ran_pois, 59–65, 65, 66–68
- ran_pois_zi, 59–65, 66, 67, 68
- ran_skewnorm, 59–66, 66, 68
- ran_student, 59–67, 67
- rbern (dbern), 8
- res_bern, 68, 69–78
- res_beta_binom, 68, 69, 70–78
- res_binom, 68, 69, 70, 71–78
- res_gamma, 68–70, 71, 72–78
- res_gamma_pois, 68–71, 71, 73–78
- res_gamma_pois_zi, 68–72, 72, 74–78
- res_lnorm, 68–73, 73, 74–78
- res_neg_binom, 68–74, 74, 75–78
- res_norm, 68–74, 75, 76–78
- res_pois, 68–75, 75, 77, 78
- res_pois_zi, 68–76, 76, 77, 78
- res_skewnorm, 68–77, 77, 78
- res_student, 68–77, 78
- rskewnorm (dskewnorm), 19

- sens_beta, 79, 80–87
- sens_exp, 79, 79, 80–87
- sens_gamma, 79, 80, 80, 81–87
- sens_gamma_pois, 79, 80, 81, 82–87
- sens_gamma_pois_zi, 79–81, 82, 83–87
- sens_lnorm, 79–82, 82, 84–87
- sens_neg_binom, 79–83, 83, 84–87
- sens_norm, 79–84, 84, 85–87
- sens_pois, 79–84, 85, 86, 87
- sens_skewnorm, 79–85, 85, 87
- sens_student, 79–86, 86
- sextreme, 52, 87
- skewness, 30, 46, 57, 58, 88, 89–94
- stats::plogis(), 27–29
- stats::pnorm(), 53
- stats::qlogis(), 33
- step, 20–22, 25–29, 31–34, 53, 54, 88
- svalue, 30, 46, 57, 58, 88, 89, 90–94

- upper, 30, 46, 57, 58, 88, 89, 90, 91–94

variance, [30](#), [46](#), [57](#), [58](#), [88–90](#), [90](#), [91–94](#)

vld_index (chk_index), [6](#)

vld_indices (chk_indices), [6](#)

vld_pars (chk_pars), [7](#)

xtr_mean, [30](#), [46](#), [57](#), [58](#), [88–91](#), [91](#), [92–94](#)

xtr_median, [30](#), [46](#), [57](#), [58](#), [88–91](#), [92](#), [93](#), [94](#)

xtr_sd, [30](#), [46](#), [57](#), [58](#), [88–92](#), [92](#), [93](#), [94](#)

zeros, [30](#), [46](#), [57](#), [58](#), [88–93](#), [93](#), [94](#)

zscore, [30](#), [46](#), [57](#), [58](#), [88–93](#), [94](#)