

Package ‘extras’

June 8, 2022

Title Helper Functions for Bayesian Analyses

Version 0.3.0

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 (>= 3.4)

Imports chk, lifecycle, stats

Suggests covr, hms, knitr, rlang, testthat (>= 3.0.0), tibble

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.2.0

NeedsCompilation no

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

Maintainer Joe Thorley <joe@poissonconsulting.ca>

Repository CRAN

Date/Publication 2022-06-08 15:30:02 UTC

R topics documented:

as_list	3
as_list_unnamed	4
chk_index	5
chk_indices	5
chk_pars	6
dev_bern	7
dev_binom	8
dev_gamma_pois	9
dev_lnorm	9
dev_neg_binom	10
dev_norm	11
dev_pois	12
dev_pois_zi	12
fabs	13
fill_all	14
fill_na	15
ilog	16
ilogit	17
invlogit	18
inv_logit	18
inv_odds	19
kurtosis	20
log<-	20
logit	21
logit<-	22
log_lik_bern	22
log_lik_binom	23
log_lik_gamma_pois	24
log_lik_lnorm	24
log_lik_neg_binom	25
log_lik_norm	26
log_lik_pois	26
log_lik_pois_zi	27
log_odds_ratio	28
log_odds_ratio2	28
lower	29
numericise	30
odds	32
odds<-	33
odds_ratio	33
odds_ratio2	34
par_pattern	35
pextreme	35
phi	36
pow	36
proportional_change	37

proportional_change2	38
proportional_difference	38
proportional_difference2	39
pvalue	40
pzeros	41
ran_bern	41
ran_binom	42
ran_gamma	43
ran_gamma_pois	43
ran_lnorm	44
ran_neg_binom	45
ran_norm	45
ran_pois	46
ran_pois_zi	47
res_bern	47
res_binom	48
res_gamma_pois	49
res_lnorm	50
res_neg_binom	50
res_norm	51
res_pois	52
res_pois_zi	53
sextreme	53
skewness	54
step	55
svalue	55
upper	56
variance	57
xtr_mean	57
xtr_median	58
xtr_sd	59
zeros	59
zscore	60

Index	61
--------------	-----------

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

Description

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

Usage

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

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

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

Arguments

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

Value

A list.

Examples

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

as_list_unnamed *As List*

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"))
```

dev_bern

Bernoulli Deviances

Description

Bernoulli Deviances

Usage

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

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>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_binom\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois_zi\(\)](#), [dev_pois\(\)](#)

Examples

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

dev_binom

Binomial Deviances

Description

Binomial Deviances

Usage

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

Arguments

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

Value

An numeric vector of the corresponding deviances or deviance residuals.

See Also

Other dev_dist: [dev_bern\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois_zi\(\)](#), [dev_pois\(\)](#)

Examples

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

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 shape and 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_binom\(\)](#), [dev_lnorm\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois_zi\(\)](#), [dev_pois\(\)](#)

Examples

```
dev_gamma_pois(c(1,3.5,4), 3, 2)
```

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

Description

Log-Normal Deviances

Usage

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

Arguments

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

Value

An numeric vector of the corresponding deviances or deviance residuals.

See Also

Other dev_dist: [dev_bern\(\)](#), [dev_binom\(\)](#), [dev_gamma_pois\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois_zi\(\)](#), [dev_pois\(\)](#)

Examples

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

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

Description

Negative Binomial Deviances

Usage

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

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the shape and 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_binom\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_norm\(\)](#), [dev_pois_zi\(\)](#), [dev_pois\(\)](#)

Examples

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

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

Description

Normal Deviances

Usage

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

Arguments

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

Value

An numeric vector of the corresponding deviances or deviance residuals.

See Also

Other dev_dist: [dev_bern\(\)](#), [dev_binom\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_neg_binom\(\)](#), [dev_pois_zi\(\)](#), [dev_pois\(\)](#)

Examples

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

`dev_pois`*Poisson Deviances*

Description

Poisson Deviances

Usage`dev_pois(x, lambda, res = FALSE)`**Arguments**

<code>x</code>	A non-negative whole numeric vector of values.
<code>lambda</code>	A non-negative numeric vector of means.
<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_binom\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois_zi\(\)](#)**Examples**`dev_pois(c(1,3.5,4), 3)`

`dev_pois_zi`*Zero-Inflated Poisson Deviances*

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_binom\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#)

Examples

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

fabs	<i>Absolute</i>
------	-----------------

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\(\)](#).

See Also

Other translations: [ilogit\(\)](#), [ilog\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [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

<code>x</code>	An object.
<code>value</code>	A scalar of the value to replace values with.
<code>...</code>	Other arguments passed to methods.
<code>nas</code>	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)

- `logical`: Fill All for logical Objects
- `integer`: Fill All for integer Objects
- `numeric`: Fill All for numeric Objects
- `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)

- `logical`: Fill Missing Values for logical Objects
- `integer`: Fill Missing Values for integer Objects
- `numeric`: Fill Missing Values for numeric Objects
- `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` A numeric atomic object.

Details

A wrapper on `exp(value)`.

Value

A numeric atomic object.

See Also

Other translations: `fabs()`, `ilogit()`, `inv_logit()`, `invlogit()`, `log<-()`, `logit<-()`, `logit()`, `phi()`, `pow()`, `step()`

Examples

```
x <- 1
ilog(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: `fabs()`, `ilog()`, `inv_logit()`, `invlogit()`, `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: [fabs\(\)](#), [ilogit\(\)](#), [ilog\(\)](#), [inv_logit\(\)](#), [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: `fabs()`, `ilogit()`, `ilog()`, `invlogit()`, `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_ratio()`, `odds<-()`, `odds_ratio()`, `odds()`

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)
```

log<-	<i>Log Transformation</i>
-------	---------------------------

Description

Replaces a object with the exponent of value.

Usage

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

Arguments

x	An existing R 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: `fabs()`, `ilogit()`, `ilog()`, `inv_logit()`, `invlogit()`, `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: `fabs()`, `ilogit()`, `ilog()`, `inv_logit()`, `invlogit()`, `log<-()`, `logit<-()`, `phi()`, `pow()`, `step()`

Examples

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

logit<- *Logistic Transformation*

Description

Logistic Transformation

Usage

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

Arguments

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

Details

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

Value

Called for the side effect of updating x.

See Also

Other translations: [fabs\(\)](#), [ilogit\(\)](#), [ilog\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [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_binom\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_pois\(\)](#)

Examples

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

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_gamma_pois\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_pois\(\)](#)

Examples

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

log_lik_gamma_pois *Gamma Poisson Log-Likelihood*

Description

Gamma Poisson Log-Likelihood

Usage

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

Arguments

x A vector of 0s and 1s.
lambda A non-negative numeric vector of means.
theta A non-negative numeric vector of the shape and rate.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_binom\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_pois\(\)](#)

Examples

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

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_binom\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_pois\(\)](#)

Examples

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

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 vector of 0s and 1s.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the shape and rate.

Value

An numeric vector of the corresponding log-likelihoods.

See Also

Other log_lik_dist: [log_lik_bern\(\)](#), [log_lik_binom\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_pois\(\)](#)

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_binom\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_pois_zi\(\)](#), [log_lik_pois\(\)](#)

Examples

```
dev_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_binom\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois_zi\(\)](#)

Examples

```
log_lik_pois(c(1,3.5,4), 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_binom\(\)](#), [log_lik_gamma_pois\(\)](#), [log_lik_lnorm\(\)](#), [log_lik_neg_binom\(\)](#), [log_lik_norm\(\)](#), [log_lik_pois\(\)](#)

Examples

```
log_lik_pois_zi(c(1,3.5,4), 3, prob = 0.5)
```

log_odds_ratio *Log-Odds Ratio*

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\(\)](#), [odds<-\(\)](#), [odds_ratio\(\)](#), [odds\(\)](#)

Examples

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

log_odds_ratio2 *Log Odds Ratio2*

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

Description

Calculates the quantile-based lower credible limit.

Usage

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

Arguments

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

Details

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

Value

A number.

See Also

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

Examples

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

numericise *Numericise (or Numericize)*

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)

- logical: Numericise a logical Object
- integer: Numericise an integer Object
- double: Numericise an double Object
- factor: Numericise a factor
- Date: Numericise a Date vector
- POSIXct: Numericise a POSIXct vector
- hms: Numericise a hms vector
- matrix: Numericise a matrix
- array: Numericise an array
- 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_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_ratio\(\)](#), [odds_ratio\(\)](#), [odds\(\)](#)

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_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: [sextreme\(\)](#)

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: [fabs\(\)](#), [ilogit\(\)](#), [ilog\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [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: [fabs\(\)](#), [ilogit\(\)](#), [ilog\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [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/de/projekte/publikationen/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

An numeric vector of the random samples.

See Also

Other ran_dist: [ran_binom\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma\(\)](#), [ran_lnorm\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois_zi\(\)](#), [ran_pois\(\)](#)

Examples

```
ran_bern(10)
```

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

An numeric vector of the random samples.

See Also

Other ran_dist: [ran_bern\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma\(\)](#), [ran_lnorm\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois_zi\(\)](#), [ran_pois\(\)](#)

Examples

```
ran_binom(10)
```

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

Description

Gamma Random Samples

Usage

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

Arguments

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

Value

An numeric vector of the random samples.

See Also

Other `ran_dist`: [ran_bern\(\)](#), [ran_binom\(\)](#), [ran_gamma_pois\(\)](#), [ran_lnorm\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois_zi\(\)](#), [ran_pois\(\)](#)

Examples

```
ran_gamma(10)
```

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

Description

Gamma-Poisson Random Samples

Usage

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

Arguments

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 shape and rate.

Value

An numeric vector of the random samples.

See Also

Other `ran_dist`: [ran_bern\(\)](#), [ran_binom\(\)](#), [ran_gamma\(\)](#), [ran_lnorm\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois_zi\(\)](#), [ran_pois\(\)](#)

Examples

```
ran_gamma_pois(10, 1, 1)
```

ran_lnorm

Log-Normal Random Samples

Description

Log-Normal Random Samples

Usage

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

Arguments

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

Value

An numeric vector of the random samples.

See Also

Other `ran_dist`: [ran_bern\(\)](#), [ran_binom\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois_zi\(\)](#), [ran_pois\(\)](#)

Examples

```
ran_lnorm(10)
```

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

Description

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 shape and rate.

Value

An numeric vector of the random samples.

See Also

Other `ran_dist`: [ran_bern\(\)](#), [ran_binom\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma\(\)](#), [ran_lnorm\(\)](#), [ran_norm\(\)](#), [ran_pois_zi\(\)](#), [ran_pois\(\)](#)

Examples

```
ran_neg_binom(10, 1, 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

An numeric vector of the random samples.

See Also

Other `ran_dist`: [ran_bern\(\)](#), [ran_binom\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma\(\)](#), [ran_lnorm\(\)](#), [ran_neg_binom\(\)](#), [ran_pois_zi\(\)](#), [ran_pois\(\)](#)

Examples

```
ran_norm(10)
```

ran_pois

Poisson Random Samples

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

An numeric vector of the random samples.

See Also

Other `ran_dist`: [ran_bern\(\)](#), [ran_binom\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma\(\)](#), [ran_lnorm\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois_zi\(\)](#)

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

An numeric vector of the random samples.

See Also

Other ran_dist: [ran_bern\(\)](#), [ran_binom\(\)](#), [ran_gamma_pois\(\)](#), [ran_gamma\(\)](#), [ran_lnorm\(\)](#), [ran_neg_binom\(\)](#), [ran_norm\(\)](#), [ran_pois\(\)](#)

Examples

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

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

Description

Bernoulli Residuals

Usage

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

Arguments

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

Value

An numeric vector of the corresponding residuals.

See Also

Other res_dist: [res_binom\(\)](#), [res_gamma_pois\(\)](#), [res_lnorm\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois_zi\(\)](#), [res_pois\(\)](#)

Examples

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

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_gamma_pois\(\)](#), [res_lnorm\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois_zi\(\)](#), [res_pois\(\)](#)

Examples

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

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 shape and 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_binom\(\)](#), [res_lnorm\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois_zi\(\)](#), [res_pois\(\)](#)

Examples

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

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_binom\(\)](#), [res_gamma_pois\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois_zi\(\)](#), [res_pois\(\)](#)

Examples

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

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

Description

Negative Binomial Residuals

Usage

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

Arguments

x	A non-negative whole numeric vector of values.
lambda	A non-negative numeric vector of means.
theta	A non-negative numeric vector of the shape and 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_binom\(\)](#), [res_gamma_pois\(\)](#), [res_lnorm\(\)](#), [res_norm\(\)](#), [res_pois_zi\(\)](#), [res_pois\(\)](#)

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_binom\(\)](#), [res_gamma_pois\(\)](#), [res_lnorm\(\)](#), [res_neg_binom\(\)](#), [res_pois_zi\(\)](#), [res_pois\(\)](#)

Examples

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

res_pois

Poisson Residuals

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_binom\(\)](#), [res_gamma_pois\(\)](#), [res_lnorm\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois_zi\(\)](#)

Examples

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

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

Description

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 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_binom\(\)](#), [res_gamma_pois\(\)](#), [res_lnorm\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#), [res_pois\(\)](#)

Examples

```
res_pois_zi(c(1,3.5,4), 6, 0.5, type = "raw")
```

sextreme	<i>Extreme Surprisal</i>
----------	--------------------------

Description

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

Usage

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

Arguments

- `x` A numeric vector of values between 0 and 1.
- `directional` A flag specifying whether probabilities less than 0.5 should be returned as negative values.

Value

A numeric vector of surprisal values.

See Also

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

Examples

```
sextreme(seq(0.1, 0.9, by = 0.1))
sextreme(seq(0.1, 0.9, by = 0.1), directional = TRUE)
```

skewness

Skewness

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

*Step***Description**

Step

Usage

step(x)

Arguments

x A numeric atomic object.

Value

A logical value.

See Also

Other translations: [fabs\(\)](#), [ilogit\(\)](#), [ilog\(\)](#), [inv_logit\(\)](#), [invlogit\(\)](#), [log<-\(\)](#), [logit<-\(\)](#), [logit\(\)](#), [phi\(\)](#), [pow\(\)](#)

Examples

step(1)

 svalue

*Surprisal Value***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

x A numeric object of MCMC values.
 threshold A number of the threshold value.
 na_rm 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

<code>x</code>	A numeric vector of MCMC values.
<code>conf_level</code>	A numeric scalar between 0 and 1 specifying the confidence level.
<code>na_rm</code>	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	<i>Zeros</i>
-------	--------------

Description

The number of zeros in an numeric object.

Usage

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

Arguments

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

Value

A non-negative integer.

See Also

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

Examples

```
zeros(c(0:2))
```

zscore

Z-Score

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

- * **dev_dist**
 - dev_bern, 7
 - dev_binom, 8
 - dev_gamma_pois, 9
 - dev_lnorm, 9
 - dev_neg_binom, 10
 - dev_norm, 11
 - dev_pois, 12
 - dev_pois_zi, 12
- * **fill**
 - fill_all, 14
 - fill_na, 15
- * **log_lik_dist**
 - log_lik_bern, 22
 - log_lik_binom, 23
 - log_lik_gamma_pois, 24
 - log_lik_lnorm, 24
 - log_lik_neg_binom, 25
 - log_lik_norm, 26
 - log_lik_pois, 26
 - log_lik_pois_zi, 27
- * **odds fun2**
 - log_odds_ratio2, 28
 - odds_ratio2, 34
- * **odds**
 - inv_odds, 19
 - log_odds_ratio, 28
 - odds, 32
 - odds<-, 33
 - odds_ratio, 33
- * **proportional fun2**
 - proportional_change2, 38
 - proportional_difference2, 39
- * **proportional**
 - proportional_change, 37
 - proportional_difference, 38
- * **ran_dist**
 - ran_bern, 41
 - ran_binom, 42
 - ran_gamma, 43
 - ran_gamma_pois, 43
 - ran_lnorm, 44
 - ran_neg_binom, 45
 - ran_norm, 45
 - ran_pois, 46
 - ran_pois_zi, 47
- * **res_dist**
 - res_bern, 47
 - res_binom, 48
 - res_gamma_pois, 49
 - res_lnorm, 50
 - res_neg_binom, 50
 - res_norm, 51
 - res_pois, 52
 - res_pois_zi, 53
- * **residuals**
 - pextreme, 35
 - sextreme, 53
- * **summary**
 - kurtosis, 20
 - lower, 29
 - pvalue, 40
 - pzeros, 41
 - skewness, 54
 - svalue, 55
 - upper, 56
 - variance, 57
 - xtr_mean, 57
 - xtr_median, 58
 - xtr_sd, 59
 - zeros, 59
 - zscore, 60
- * **translations**
 - fabs, 13
 - ilog, 16
 - ilogit, 17
 - inv_logit, 18
 - invlogit, 18

- log<-, 20
- logit, 21
- logit<-, 22
- phi, 36
- pow, 36
- step, 55
- abs, 13
- as_list, 3
- as_list_unnamed, 4
- chk_index, 5
- chk_indices, 5
- chk_pars, 6
- dev_bern, 7, 8–13
- dev_binom, 8, 8, 9–13
- dev_gamma_pois, 8, 9, 10–13
- dev_lnorm, 8, 9, 9, 10–13
- dev_neg_binom, 8–10, 10, 11–13
- dev_norm, 8–10, 11, 12, 13
- dev_pois, 8–11, 12, 13
- dev_pois_zi, 8–12, 12
- exp, 17, 21
- fabs, 13, 17–19, 21, 22, 36, 37, 55
- fill_all, 14, 16
- fill_na, 15, 15
- ilog, 13, 16, 17–19, 21, 22, 36, 37, 55
- ilogit, 13, 17, 17, 18, 19, 21, 22, 36, 37, 55
- inv_logit, 13, 17, 18, 18, 21, 22, 36, 37, 55
- inv_odds, 19, 28, 32–34
- invlogit, 13, 17, 18, 19, 21, 22, 36, 37, 55
- kurtosis, 20, 29, 40, 41, 54, 56–60
- log<-, 20
- log_lik_bern, 22, 23–27
- log_lik_binom, 23, 23, 24–27
- log_lik_gamma_pois, 23, 24, 25–27
- log_lik_lnorm, 23, 24, 24, 25–27
- log_lik_neg_binom, 23–25, 25, 26, 27
- log_lik_norm, 23–25, 26, 27
- log_lik_pois, 23–26, 26, 27
- log_lik_pois_zi, 23–27, 27
- log_odds_ratio, 19, 28, 32–34
- log_odds_ratio2, 28, 34
- logit, 13, 17–19, 21, 21, 22, 36, 37, 55
- logit<-, 22
- lower, 20, 29, 40, 41, 54, 56–60
- numericise, 30
- numericize (numericise), 30
- odds, 19, 28, 32, 33, 34
- odds<-, 33
- odds_ratio, 19, 28, 32, 33, 33
- odds_ratio2, 29, 34
- par_pattern, 35
- pextreme, 35, 54
- phi, 13, 17–19, 21, 22, 36, 37, 55
- pow, 13, 17–19, 21, 22, 36, 36, 55
- proportional_change, 37, 39
- proportional_change2, 38, 39
- proportional_difference, 37, 38
- proportional_difference2, 38, 39
- pvalue, 20, 29, 40, 41, 54–60
- pzeros, 20, 29, 40, 41, 54, 56–60
- ran_bern, 41, 42–47
- ran_binom, 42, 42, 43–47
- ran_gamma, 42, 43, 44–47
- ran_gamma_pois, 42, 43, 43, 44–47
- ran_lnorm, 42–44, 44, 45–47
- ran_neg_binom, 42–44, 45, 46, 47
- ran_norm, 42–45, 45, 46, 47
- ran_pois, 42–46, 46, 47
- ran_pois_zi, 42–46, 47
- res_bern, 47, 49–53
- res_binom, 48, 48, 49–53
- res_gamma_pois, 48, 49, 49, 50–53
- res_lnorm, 48, 49, 50, 51–53
- res_neg_binom, 48–50, 50, 52, 53
- res_norm, 48–51, 51, 52, 53
- res_pois, 48–52, 52, 53
- res_pois_zi, 48–52, 53
- sextreme, 35, 53
- skewness, 20, 29, 40, 41, 54, 56–60
- stats::plogis(), 17–19
- stats::pnorm(), 36
- stats::qlogis(), 21
- step, 13, 17–19, 21, 22, 36, 37, 55
- svalue, 20, 29, 40, 41, 54, 55, 56–60
- upper, 20, 29, 40, 41, 54, 56, 56, 57–60

variance, [20](#), [29](#), [40](#), [41](#), [54](#), [56](#), [57](#), [58–60](#)
vld_index (chk_index), [5](#)
vld_indices (chk_indices), [5](#)
vld_pars (chk_pars), [6](#)

xtr_mean, [20](#), [29](#), [40](#), [41](#), [54](#), [56](#), [57](#), [57](#), [58–60](#)
xtr_median, [20](#), [29](#), [40](#), [41](#), [54](#), [56–58](#), [58](#), [59](#),
[60](#)
xtr_sd, [20](#), [29](#), [40](#), [41](#), [54](#), [56–58](#), [59](#), [60](#)

zeros, [20](#), [29](#), [40](#), [41](#), [54](#), [56–59](#), [59](#), [60](#)
zscore, [20](#), [29](#), [40](#), [41](#), [54](#), [56–60](#), [60](#)