

Package ‘extras’

August 5, 2021

Title Helper Functions for Bayesian Analyses

Version 0.2.0

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 'BUGS' (Bayesian Using Gibbs Sampling) and 'JAGS' (Just Another Gibbs Sampler) 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, tibble

Encoding UTF-8

Language en-US

RoxygenNote 7.1.1

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 2021-08-05 04:20:31 UTC

R topics documented:

as_list	3
as_list_unnamed	4
chk_index	4
chk_indices	5
chk_pars	6
dev_bern	7
dev_binom	7
dev_gamma_pois	8
dev_lnorm	9
dev_neg_binom	10
dev_norm	10
dev_pois	11
fabs	12
fill_all	12
fill_na	14
ilogit	15
invlogit	16
kurtosis	17
log<-	17
logit	18
logit<-	19
log_lik_bern	19
log_lik_binom	20
log_lik_gamma_pois	21
log_lik_lnorm	21
log_lik_neg_binom	22
log_lik_norm	23
log_lik_pois	23
log_odds_ratio	24
lower	25
numericise	25
odds	28
odds_ratio	28
par_pattern	29
pextreme	29
phi	30
pow	31
pvalue	31
ran_bern	32
ran_binom	33
ran_gamma	33
ran_gamma_pois	34
ran_lnorm	35
ran_neg_binom	35
ran_norm	36
ran_pois	37

res_bern	37
res_binom	38
res_gamma_pois	39
res_lnorm	39
res_neg_binom	40
res_norm	41
res_pois	42
sextreme	42
skewness	43
svalue	44
upper	45
variance	45
xtr_mean	46
xtr_median	47
xtr_sd	47
zeros	48
zscore	49

Index	50
--------------	-----------

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

Description

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

Usage

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

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

Arguments

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

Value

A list.

Examples

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

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

Description

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

Usage

```
as_list_unnamed(x, ...)
```

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

Arguments

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

Value

A list.

Examples

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

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

Description

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

Usage

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

```
vld_index(x)
```

Arguments

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

Value

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

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

Functions

- `vld_index`: Validate Index

Examples

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

`chk_indices`*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	<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_binom\(\)](#), [dev_gamma_pois\(\)](#), [dev_lnorm\(\)](#), [dev_neg_binom\(\)](#), [dev_norm\(\)](#), [dev_pois\(\)](#)

Examples

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

dev_binom	<i>Binomial Deviances</i>
-----------	---------------------------

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\(\)](#)

Examples

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

dev_gamma_pois

Gamma-Poisson Deviances

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\(\)](#)

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\(\)](#)

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\(\)](#)

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\(\)](#)

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

Examples

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

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\(\)](#).

See Also

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

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)

- 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)
```

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\(\)](#), [invlogit\(\)](#), [log<-\(\)](#), [logit<-\(\)](#), [logit\(\)](#), [phi\(\)](#), [pow\(\)](#)

Examples

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

invlogit

Inverse Logistic Transformation

Description

Inverse logistically transforms a numeric atomic object.

Usage

```
invlogit(x)
```

Arguments

x A numeric atomic object.

Details

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

Value

A numeric atomic object.

See Also

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

Examples

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

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

Description

Kurtosis

Usage

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

Arguments

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

Value

A number.

See Also

Other summary: [lower\(\)](#), [pvalue\(\)](#), [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()`, `invlogit()`, `logit<-()`, `logit()`, `phi()`, `pow()`

Examples

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

logit

Logistic Transformation

Description

Logistic transforms a numeric atomic object.

Usage

```
logit(x)
```

Arguments

`x` A numeric atomic object.

Details

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

Value

The logistically transformed numeric atomic object.

See Also

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

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\(\)](#), [invlogit\(\)](#), [log<-\(\)](#), [logit\(\)](#), [phi\(\)](#), [pow\(\)](#)

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\(\)](#)

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\(\)](#)

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\(\)](#)

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\(\)](#)

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\(\)](#)

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\(\)](#)

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\(\)](#)

Examples

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

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: [odds_ratio\(\)](#), [odds\(\)](#)

Examples

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

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\(\)](#), [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)

- 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: [log_odds_ratio\(\)](#), [odds_ratio\(\)](#)

Examples

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

odds_ratio

Odds Ratio

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: [log_odds_ratio\(\)](#), [odds\(\)](#)

Examples

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

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

Phi

Description

The standard normal cumulative density function.

Usage

```
phi(x)
```

Arguments

x A numeric atomic object.

Details

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

Value

A numeric atomic object.

See Also

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

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\(\)](#), [invlogit\(\)](#), [log<-\(\)](#), [logit<-\(\)](#), [logit\(\)](#), [phi\(\)](#)

Examples

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

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

Examples

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

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\(\)](#)

Examples

```
ran_bern(10)
```

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

Description

Binomial Random Samples

Usage

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

Arguments

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

Value

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\(\)](#)

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\(\)](#)

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\(\)](#)

Examples

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

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

Description

Log-Normal Random Samples

Usage

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

Arguments

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

Value

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\(\)](#)

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\(\)](#)

Examples

```
ran_neg_binom(10, 1, 1)
```

ran_norm

Normal Random Samples

Description

Normal Random Samples

Usage

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

Arguments

n A non-negative whole number of the number of random samples to generate.

mean A numeric vector of the means.

sd A non-negative numeric vector of the standard deviations.

Value

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\(\)](#)

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

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\(\)](#)

Examples

```
ran_pois(10)
```

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' or 'dev').
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\(\)](#)

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' or 'dev').
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\(\)](#)

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' or 'dev').
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\(\)](#)

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' or 'dev').
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\(\)](#)

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' or 'dev').
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\(\)](#)

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' or 'dev').
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\(\)](#)

Examples

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

`res_pois`*Poisson Residuals*

Description

Poisson Residuals

Usage

```
res_pois(x, lambda = 1, 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>type</code>	A string of the residual type ('raw' or 'dev').
<code>simulate</code>	A flag specifying whether to simulate residuals.

Value

An numeric vector of the corresponding residuals.

See AlsoOther `res_dist`: [res_bern\(\)](#), [res_binom\(\)](#), [res_gamma_pois\(\)](#), [res_lnorm\(\)](#), [res_neg_binom\(\)](#), [res_norm\(\)](#)**Examples**

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

`sextreme`*Extreme Surprisal*

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

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

Arguments

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

Value

A numeric vector of surprisal values.

See Also

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

Examples

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

skewness	<i>Skewness</i>
----------	-----------------

Description

Skewness

Usage

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

Arguments

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

Value

A number.

See Also

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

Examples

```
skewness(1:10)
```

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

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\(\)](#), [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\(\)](#), [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\(\)](#), [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\(\)](#), [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\(\)](#), [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\(\)](#), [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\(\)](#), [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\(\)](#), [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, 7
 - dev_gamma_pois, 8
 - dev_lnorm, 9
 - dev_neg_binom, 10
 - dev_norm, 10
 - dev_pois, 11
- * **fill**
 - fill_all, 12
 - fill_na, 14
- * **log_lik_dist**
 - log_lik_bern, 19
 - log_lik_binom, 20
 - log_lik_gamma_pois, 21
 - log_lik_lnorm, 21
 - log_lik_neg_binom, 22
 - log_lik_norm, 23
 - log_lik_pois, 23
- * **odds**
 - log_odds_ratio, 24
 - odds, 28
 - odds_ratio, 28
- * **ran_dist**
 - ran_bern, 32
 - ran_binom, 33
 - ran_gamma, 33
 - ran_gamma_pois, 34
 - ran_lnorm, 35
 - ran_neg_binom, 35
 - ran_norm, 36
 - ran_pois, 37
- * **res_dist**
 - res_bern, 37
 - res_binom, 38
 - res_gamma_pois, 39
 - res_lnorm, 39
 - res_neg_binom, 40
 - res_norm, 41
 - res_pois, 42
- * **residuals**
 - pextreme, 29
 - sextreme, 42
- * **summary**
 - kurtosis, 17
 - lower, 25
 - pvalue, 31
 - skewness, 43
 - svalue, 44
 - upper, 45
 - variance, 45
 - xtr_mean, 46
 - xtr_median, 47
 - xtr_sd, 47
 - zeros, 48
 - zscore, 49
- * **translations**
 - fabs, 12
 - ilogit, 15
 - invlogit, 16
 - log<-, 17
 - logit, 18
 - logit<-, 19
 - phi, 30
 - pow, 31
- abs, 12
- as_list, 3
- as_list_unnamed, 4
- chk_index, 4
- chk_indices, 5
- chk_pars, 6
- dev_bern, 7, 8–11
- dev_binom, 7, 7, 8–11
- dev_gamma_pois, 7, 8, 8, 9–11
- dev_lnorm, 7, 8, 9, 10, 11
- dev_neg_binom, 7–9, 10, 11

- dev_norm, 7–10, 10, 11
- dev_pois, 7–11, 11
- exp, 18
- fabs, 12, 16, 18, 19, 30, 31
- fill_all, 12, 15
- fill_na, 13, 14
- ilogit, 12, 15, 16, 18, 19, 30, 31
- invlogit, 12, 16, 16, 18, 19, 30, 31
- kurtosis, 17, 25, 32, 43–49
- log<-, 17
- log_lik_bern, 19, 20–24
- log_lik_binom, 20, 20, 21–24
- log_lik_gamma_pois, 20, 21, 22–24
- log_lik_lnorm, 20, 21, 21, 22–24
- log_lik_neg_binom, 20–22, 22, 23, 24
- log_lik_norm, 20–22, 23, 24
- log_lik_pois, 20–23, 23
- log_odds_ratio, 24, 28, 29
- logit, 12, 16, 18, 18, 19, 30, 31
- logit<-, 19
- lower, 17, 25, 32, 43–49
- numericise, 25
- numericize (numericise), 25
- odds, 24, 28, 29
- odds_ratio, 24, 28, 28
- par_pattern, 29
- pextreme, 29, 43
- phi, 12, 16, 18, 19, 30, 31
- pow, 12, 16, 18, 19, 30, 31
- pvalue, 17, 25, 31, 43–49
- ran_bern, 32, 33–37
- ran_binom, 32, 33, 34–37
- ran_gamma, 32, 33, 33, 34–37
- ran_gamma_pois, 32–34, 34, 35–37
- ran_lnorm, 32–34, 35, 36, 37
- ran_neg_binom, 32–35, 35, 36, 37
- ran_norm, 32–36, 36, 37
- ran_pois, 32–36, 37
- res_bern, 37, 38–42
- res_binom, 38, 38, 39–42
- res_gamma_pois, 38, 39, 40–42
- res_lnorm, 38, 39, 39, 40–42
- res_neg_binom, 38–40, 40, 41, 42
- res_norm, 38–40, 41, 42
- res_pois, 38–41, 42
- sextreme, 30, 42
- skewness, 17, 25, 32, 43, 44–49
- stats::plogis(), 15, 16
- stats::pnorm(), 30
- stats::qlogis(), 18
- svalue, 17, 25, 32, 43, 44, 45–49
- upper, 17, 25, 32, 43, 44, 45, 46–49
- variance, 17, 25, 32, 43–45, 45, 46–49
- vld_index (chk_index), 4
- vld_indices (chk_indices), 5
- vld_pars (chk_pars), 6
- xtr_mean, 17, 25, 32, 43–46, 46, 47–49
- xtr_median, 17, 25, 32, 43–46, 47, 48, 49
- xtr_sd, 17, 25, 32, 43–47, 47, 48, 49
- zeros, 17, 25, 32, 43–48, 48, 49
- zscore, 17, 25, 32, 43–48, 49