

# Package ‘extras’

May 10, 2023

**Title** Helper Functions for Bayesian Analyses

**Version** 0.6.1

**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 (>= 3.6)

**Imports** chk, lifecycle, stats

**Suggests** aods3, covr, hms, ggplot2, knitr, MASS, rlang, rmarkdown, scales, testthat (>= 3.0.0), tibble, tidyr, viridis

**Config/testthat/edition** 3

**Encoding** UTF-8

**Language** en-US

**RoxygenNote** 7.2.3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Joe Thorley [aut, cre] (<<https://orcid.org/0000-0002-7683-4592>>),  
Nicole Hill [aut] (<<https://orcid.org/0000-0002-7623-2153>>),  
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** 2023-05-10 07:30:02 UTC

**R topics documented:**

as_list . . . . .	4
as_list_unnamed . . . . .	4
chk_index . . . . .	5
chk_indices . . . . .	6
chk_pars . . . . .	6
dbern . . . . .	7
dev_bern . . . . .	8
dev_beta_binom . . . . .	9
dev_binom . . . . .	10
dev_gamma . . . . .	11
dev_gamma_pois . . . . .	11
dev_gamma_pois_zi . . . . .	12
dev_lnorm . . . . .	13
dev_neg_binom . . . . .	14
dev_norm . . . . .	14
dev_pois . . . . .	15
dev_pois_zi . . . . .	16
dev_student . . . . .	17
fabs . . . . .	17
fill_all . . . . .	18
fill_na . . . . .	19
ilog . . . . .	21
ilogit . . . . .	22
invlogit . . . . .	22
inv_logit . . . . .	23
inv_odds . . . . .	24
kurtosis . . . . .	24
log<- . . . . .	25
logit . . . . .	26
logit<- . . . . .	26
log_lik_bern . . . . .	27
log_lik_beta_binom . . . . .	28
log_lik_binom . . . . .	29
log_lik_gamma . . . . .	29
log_lik_gamma_pois . . . . .	30
log_lik_gamma_pois_zi . . . . .	31
log_lik_lnorm . . . . .	31
log_lik_neg_binom . . . . .	32
log_lik_norm . . . . .	33
log_lik_pois . . . . .	34
log_lik_pois_zi . . . . .	34
log_lik_student . . . . .	35
log_odds . . . . .	36
log_odds<- . . . . .	36
log_odds_ratio . . . . .	37
log_odds_ratio2 . . . . .	38

lower	38
numericise	39
odds	41
odds<-	42
odds_ratio	43
odds_ratio2	43
par_pattern	44
pextreme	44
phi	45
pow	46
proportional_change	46
proportional_change2	47
proportional_difference	48
proportional_difference2	48
pvalue	49
pzeros	50
ran_bern	50
ran_beta_binom	51
ran_binom	52
ran_gamma	52
ran_gamma_pois	53
ran_gamma_pois_zi	54
ran_lnorm	54
ran_neg_binom	55
ran_norm	56
ran_pois	56
ran_pois_zi	57
ran_student	58
res_bern	58
res_beta_binom	59
res_binom	60
res_gamma	61
res_gamma_pois	62
res_gamma_pois_zi	62
res_lnorm	63
res_neg_binom	64
res_norm	65
res_pois	66
res_pois_zi	66
res_student	67
sextreme	68
skewness	69
step	69
svalue	70
upper	71
variance	71
xtr_mean	72
xtr_median	73

xtr_sd . . . . .	73
zeros . . . . .	74
zscore . . . . .	75

<b>Index</b>	<b>76</b>
--------------	-----------

---

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"))
```

**Description**

Bernoulli Distribution

**Usage**

```

dbern(x, prob, log = FALSE)

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

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

rbern(n, prob)

```

**Arguments**

x	A vector of 0s and 1s.
prob	A numeric vector of values between 0 and 1 of the probability of success.
log	A flag specifying whether to return the log-transformed value.
q	A vector of quantiles.
lower.tail	A flag specifying whether to return the lower or upper tail of the distribution.
p	A vector of probabilities.
n	A non-negative whole number of the number of random samples to generate.

**Value**

An numeric vector of the random samples.

**Examples**

```
dbern(1, 0.5)
```

---

dev\_bern

*Bernoulli Deviances*


---

**Description**

Bernoulli Deviances

**Usage**

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

**Arguments**

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



**Value**

An numeric vector of the corresponding deviances or deviance residuals.

**See Also**

Other dev\_dist: [dev\\_beta\\_binom\(\)](#), [dev\\_binom\(\)](#), [dev\\_gamma\\_pois\(\)](#), [dev\\_gamma\(\)](#), [dev\\_lnorm\(\)](#), [dev\\_neg\\_binom\(\)](#), [dev\\_norm\(\)](#), [dev\\_pois\\_zi\(\)](#), [dev\\_pois\(\)](#), [dev\\_student\(\)](#)

**Examples**

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

---

dev_beta_binom	<i>Beta-Binomial Deviances</i>
----------------	--------------------------------

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding deviances or deviance residuals.

**See Also**

Other dev\_dist: [dev\\_bern\(\)](#), [dev\\_binom\(\)](#), [dev\\_gamma\\_pois\(\)](#), [dev\\_gamma\(\)](#), [dev\\_lnorm\(\)](#), [dev\\_neg\\_binom\(\)](#), [dev\\_norm\(\)](#), [dev\\_pois\\_zi\(\)](#), [dev\\_pois\(\)](#), [dev\\_student\(\)](#)

**Examples**

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

---

dev\_binom

*Binomial Deviances*

---

**Description**

Binomial Deviances

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding deviances or deviance residuals.

**See Also**

Other dev\_dist: [dev\\_bern\(\)](#), [dev\\_beta\\_binom\(\)](#), [dev\\_gamma\\_pois\(\)](#), [dev\\_gamma\(\)](#), [dev\\_lnorm\(\)](#), [dev\\_neg\\_binom\(\)](#), [dev\\_norm\(\)](#), [dev\\_pois\\_zi\(\)](#), [dev\\_pois\(\)](#), [dev\\_student\(\)](#)

**Examples**

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

---

`dev_gamma`*Gamma Deviances*

---

**Description**

Gamma Deviances

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

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

**Value**

An numeric vector of the corresponding deviances or deviance residuals.

**See Also**Other dev\_dist: [dev\\_bern\(\)](#), [dev\\_beta\\_binom\(\)](#), [dev\\_binom\(\)](#), [dev\\_gamma\\_pois\(\)](#), [dev\\_lnorm\(\)](#), [dev\\_neg\\_binom\(\)](#), [dev\\_norm\(\)](#), [dev\\_pois\\_zi\(\)](#), [dev\\_pois\(\)](#), [dev\\_student\(\)](#)**Examples**`dev_gamma(c(0, 1, 2), 1, 2)`

---

`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 dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
res	A flag specifying whether to return the deviance residual as opposed to the deviance.

**Value**

An numeric vector of the corresponding deviances or deviance residuals.

**See Also**

Other dev\_dist: [dev\\_bern\(\)](#), [dev\\_beta\\_binom\(\)](#), [dev\\_binom\(\)](#), [dev\\_gamma\(\)](#), [dev\\_lnorm\(\)](#), [dev\\_neg\\_binom\(\)](#), [dev\\_norm\(\)](#), [dev\\_pois\\_zi\(\)](#), [dev\\_pois\(\)](#), [dev\\_student\(\)](#)

**Examples**

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

---

dev\_gamma\_pois\_zi      *Zero-Inflated Gamma-Poisson Deviances*

---

**Description**

Zero-Inflated Gamma-Poisson Deviances

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding deviances or deviance residuals.

**Examples**

```
dev_gamma_pois_zi(c(1,3.5,4), 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\\_beta\\_binom\(\)](#), [dev\\_binom\(\)](#), [dev\\_gamma\\_pois\(\)](#), [dev\\_gamma\(\)](#), [dev\\_neg\\_binom\(\)](#), [dev\\_norm\(\)](#), [dev\\_pois\\_zi\(\)](#), [dev\\_pois\(\)](#), [dev\\_student\(\)](#)

**Examples**

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

---

dev\_neg\_binom                      *Negative Binomial Deviances*

---

**Description**

Negative Binomial Deviances

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding deviances or deviance residuals.

**See Also**

Other dev\_dist: [dev\\_bern\(\)](#), [dev\\_beta\\_binom\(\)](#), [dev\\_binom\(\)](#), [dev\\_gamma\\_pois\(\)](#), [dev\\_gamma\(\)](#), [dev\\_lnorm\(\)](#), [dev\\_norm\(\)](#), [dev\\_pois\\_zi\(\)](#), [dev\\_pois\(\)](#), [dev\\_student\(\)](#)

**Examples**

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

---

dev\_norm                              *Normal Deviances*

---

**Description**

Normal Deviances

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding deviances or deviance residuals.

**See Also**

Other dev\_dist: [dev\\_bern\(\)](#), [dev\\_beta\\_binom\(\)](#), [dev\\_binom\(\)](#), [dev\\_gamma\\_pois\(\)](#), [dev\\_gamma\(\)](#), [dev\\_lnorm\(\)](#), [dev\\_neg\\_binom\(\)](#), [dev\\_pois\\_zi\(\)](#), [dev\\_pois\(\)](#), [dev\\_student\(\)](#)

**Examples**

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

---

dev\_pois

*Poisson Deviances*


---

**Description**

Poisson Deviances

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding deviances or deviance residuals.

**See Also**

Other dev\_dist: [dev\\_bern\(\)](#), [dev\\_beta\\_binom\(\)](#), [dev\\_binom\(\)](#), [dev\\_gamma\\_pois\(\)](#), [dev\\_gamma\(\)](#), [dev\\_lnorm\(\)](#), [dev\\_neg\\_binom\(\)](#), [dev\\_norm\(\)](#), [dev\\_pois\\_zi\(\)](#), [dev\\_student\(\)](#)

**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**

<code>x</code>	A non-negative whole numeric vector of values.
<code>lambda</code>	A non-negative numeric vector of means.
<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\\_bern\(\)](#), [dev\\_beta\\_binom\(\)](#), [dev\\_binom\(\)](#), [dev\\_gamma\\_pois\(\)](#), [dev\\_gamma\(\)](#), [dev\\_lnorm\(\)](#), [dev\\_neg\\_binom\(\)](#), [dev\\_norm\(\)](#), [dev\\_pois\(\)](#), [dev\\_student\(\)](#)

**Examples**

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



---

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

---

**Description**

Student's t Deviances

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding deviances or deviance residuals.

**See Also**

Other dev\_dist: [dev\\_bern\(\)](#), [dev\\_beta\\_binom\(\)](#), [dev\\_binom\(\)](#), [dev\\_gamma\\_pois\(\)](#), [dev\\_gamma\(\)](#), [dev\\_lnorm\(\)](#), [dev\\_neg\\_binom\(\)](#), [dev\\_norm\(\)](#), [dev\\_pois\\_zi\(\)](#), [dev\\_pois\(\)](#)

**Examples**

```
dev_student(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**

x                    An object.

value                A scalar of the value to replace values with.

...                  Other arguments passed to methods.

nas                  A flag specifying whether to also fill missing values.

**Details**

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

**Value**

The modified object.

**Methods (by class)**

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

**See Also**

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

**Examples**

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

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

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

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

---

fill\_na

*Fill Missing Values*

---

**Description**

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

**Usage**

```
fill_na(x, value, ...)  
  
## S3 method for class 'logical'  
fill_na(x, value = FALSE, ...)  
  
## S3 method for class 'integer'  
fill_na(x, value = 0L, ...)  
  
## S3 method for class 'numeric'  
fill_na(x, value = 0, ...)  
  
## S3 method for class 'character'  
fill_na(x, value = "0", ...)
```

**Arguments**

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

**Details**

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

**Value**

The modified object.

**Methods (by class)**

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

**See Also**

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

**Examples**

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

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

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

---

ilog *Inverse Log Transformation*

---

### Description

Inverse log transforms a numeric atomic object.

### Usage

```
ilog(x)
```

### Arguments

x                    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`*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()`, `ilog()`, `inv_logit()`, `log<-()`, `logit<-()`, `logit()`, `phi()`, `pow()`, `step()`

**Examples**

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

---

inv\_logit

*Inverse Logistic Transformation*

---

**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<-\(\)](#), [log\\_odds\\_ratio\(\)](#), [log\\_odds\(\)](#), [odds<-\(\)](#), [odds\\_ratio\(\)](#), [odds\(\)](#)

**Examples**

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

---

`kurtosis`*Kurtosis*

---

**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<-

*Log Transformation*

---

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

---

**Description**

Logistic Transformation

**Usage**

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

**Arguments**

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

**Details**

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

**Value**

Called for the side effect of updating `x`.

**See Also**

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

**Examples**

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

---

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

---

**Description**

Bernoulli Log-Likelihood

**Usage**

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

**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.

**Value**

An numeric vector of the corresponding log-likelihoods.

**See Also**

Other `log_lik_dist`: `log_lik_beta_binom()`, `log_lik_binom()`, `log_lik_gamma_pois_zi()`, `log_lik_gamma_pois()`, `log_lik_gamma()`, `log_lik_lnorm()`, `log_lik_neg_binom()`, `log_lik_norm()`, `log_lik_pois_zi()`, `log_lik_pois()`, `log_lik_student()`

**Examples**

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

---

log\_lik\_beta\_binom      *Beta-Binomial Log-Likelihood*

---

### Description

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

### Usage

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

### Arguments

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

### 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\\_zi\(\)](#), [log\\_lik\\_gamma\\_pois\(\)](#), [log\\_lik\\_gamma\(\)](#), [log\\_lik\\_lnorm\(\)](#), [log\\_lik\\_neg\\_binom\(\)](#), [log\\_lik\\_norm\(\)](#), [log\\_lik\\_pois\\_zi\(\)](#), [log\\_lik\\_pois\(\)](#), [log\\_lik\\_student\(\)](#)

### Examples

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

---

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

---

**Description**

Binomial Log-Likelihood

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding log-likelihoods.

**See Also**

Other `log_lik_dist`: [log\\_lik\\_bern\(\)](#), [log\\_lik\\_beta\\_binom\(\)](#), [log\\_lik\\_gamma\\_pois\\_zi\(\)](#), [log\\_lik\\_gamma\\_pois\(\)](#), [log\\_lik\\_gamma\(\)](#), [log\\_lik\\_lnorm\(\)](#), [log\\_lik\\_neg\\_binom\(\)](#), [log\\_lik\\_norm\(\)](#), [log\\_lik\\_pois\\_zi\(\)](#), [log\\_lik\\_pois\(\)](#), [log\\_lik\\_student\(\)](#)

**Examples**

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

---

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

---

**Description**

Gamma Log-Likelihood

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding log-likelihoods.

**See Also**

Other log\_lik\_dist: [log\\_lik\\_bern\(\)](#), [log\\_lik\\_beta\\_binom\(\)](#), [log\\_lik\\_binom\(\)](#), [log\\_lik\\_gamma\\_pois\\_zi\(\)](#), [log\\_lik\\_gamma\\_pois\(\)](#), [log\\_lik\\_lnorm\(\)](#), [log\\_lik\\_neg\\_binom\(\)](#), [log\\_lik\\_norm\(\)](#), [log\\_lik\\_pois\\_zi\(\)](#), [log\\_lik\\_pois\(\)](#), [log\\_lik\\_student\(\)](#)

**Examples**

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

---

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

---

**Description**

Gamma-Poisson Log-Likelihood

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding log-likelihoods.

**See Also**

Other log\_lik\_dist: [log\\_lik\\_bern\(\)](#), [log\\_lik\\_beta\\_binom\(\)](#), [log\\_lik\\_binom\(\)](#), [log\\_lik\\_gamma\\_pois\\_zi\(\)](#), [log\\_lik\\_gamma\(\)](#), [log\\_lik\\_lnorm\(\)](#), [log\\_lik\\_neg\\_binom\(\)](#), [log\\_lik\\_norm\(\)](#), [log\\_lik\\_pois\\_zi\(\)](#), [log\\_lik\\_pois\(\)](#), [log\\_lik\\_student\(\)](#)

**Examples**

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

---

log\_lik\_gamma\_pois\_zi *Zero-Inflated Gamma-Poisson Log-Likelihood*

---

**Description**

Zero-Inflated Gamma-Poisson Log-Likelihood

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding log-likelihoods.

**See Also**

Other log\_lik\_dist: [log\\_lik\\_bern\(\)](#), [log\\_lik\\_beta\\_binom\(\)](#), [log\\_lik\\_binom\(\)](#), [log\\_lik\\_gamma\\_pois\(\)](#), [log\\_lik\\_gamma\(\)](#), [log\\_lik\\_lnorm\(\)](#), [log\\_lik\\_neg\\_binom\(\)](#), [log\\_lik\\_norm\(\)](#), [log\\_lik\\_pois\\_zi\(\)](#), [log\\_lik\\_pois\(\)](#), [log\\_lik\\_student\(\)](#)

**Examples**

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

---

log\_lik\_lnorm *Log-Normal Log-Likelihood*

---

**Description**

Log-Normal Log-Likelihood

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding log-likelihoods.

**See Also**

Other log\_lik\_dist: [log\\_lik\\_bern\(\)](#), [log\\_lik\\_beta\\_binom\(\)](#), [log\\_lik\\_binom\(\)](#), [log\\_lik\\_gamma\\_pois\\_zi\(\)](#), [log\\_lik\\_gamma\\_pois\(\)](#), [log\\_lik\\_gamma\(\)](#), [log\\_lik\\_neg\\_binom\(\)](#), [log\\_lik\\_norm\(\)](#), [log\\_lik\\_pois\\_zi\(\)](#), [log\\_lik\\_pois\(\)](#), [log\\_lik\\_student\(\)](#)

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

**Value**

An numeric vector of the corresponding log-likelihoods.

**See Also**

Other log\_lik\_dist: [log\\_lik\\_bern\(\)](#), [log\\_lik\\_beta\\_binom\(\)](#), [log\\_lik\\_binom\(\)](#), [log\\_lik\\_gamma\\_pois\\_zi\(\)](#), [log\\_lik\\_gamma\\_pois\(\)](#), [log\\_lik\\_gamma\(\)](#), [log\\_lik\\_lnorm\(\)](#), [log\\_lik\\_norm\(\)](#), [log\\_lik\\_pois\\_zi\(\)](#), [log\\_lik\\_pois\(\)](#), [log\\_lik\\_student\(\)](#)



**Examples**

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

---

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

---

**Description**

Normal Log-Likelihood

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding log-likelihoods.

**See Also**

Other log\_lik\_dist: [log\\_lik\\_bern\(\)](#), [log\\_lik\\_beta\\_binom\(\)](#), [log\\_lik\\_binom\(\)](#), [log\\_lik\\_gamma\\_pois\\_zi\(\)](#), [log\\_lik\\_gamma\\_pois\(\)](#), [log\\_lik\\_gamma\(\)](#), [log\\_lik\\_lnorm\(\)](#), [log\\_lik\\_neg\\_binom\(\)](#), [log\\_lik\\_pois\\_zi\(\)](#), [log\\_lik\\_pois\(\)](#), [log\\_lik\\_student\(\)](#)

**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\\_beta\\_binom\(\)](#), [log\\_lik\\_binom\(\)](#), [log\\_lik\\_gamma\\_pois\\_zi\(\)](#), [log\\_lik\\_gamma\\_pois\(\)](#), [log\\_lik\\_gamma\(\)](#), [log\\_lik\\_lnorm\(\)](#), [log\\_lik\\_neg\\_binom\(\)](#), [log\\_lik\\_norm\(\)](#), [log\\_lik\\_pois\\_zi\(\)](#), [log\\_lik\\_student\(\)](#)

**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\\_beta\\_binom\(\)](#), [log\\_lik\\_binom\(\)](#), [log\\_lik\\_gamma\\_pois\\_zi\(\)](#), [log\\_lik\\_gamma\\_pois\(\)](#), [log\\_lik\\_gamma\(\)](#), [log\\_lik\\_lnorm\(\)](#), [log\\_lik\\_neg\\_binom\(\)](#), [log\\_lik\\_norm\(\)](#), [log\\_lik\\_pois\(\)](#), [log\\_lik\\_student\(\)](#)

**Examples**

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

---

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

---

**Description**

Student's t Log-Likelihood

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding log-likelihoods.

**See Also**

Other log\_lik\_dist: [log\\_lik\\_bern\(\)](#), [log\\_lik\\_beta\\_binom\(\)](#), [log\\_lik\\_binom\(\)](#), [log\\_lik\\_gamma\\_pois\\_zi\(\)](#), [log\\_lik\\_gamma\\_pois\(\)](#), [log\\_lik\\_gamma\(\)](#), [log\\_lik\\_lnorm\(\)](#), [log\\_lik\\_neg\\_binom\(\)](#), [log\\_lik\\_norm\(\)](#), [log\\_lik\\_pois\\_zi\(\)](#), [log\\_lik\\_pois\(\)](#)

**Examples**

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

---

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

---

**Description**

Calculates the log odds for probabilities.

**Usage**

```
log_odds(x)
```

**Arguments**

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

**Value**

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

**See Also**

Other odds: [inv\\_odds\(\)](#), [log\\_odds<-\(\)](#), [log\\_odds\\_ratio\(\)](#), [odds<-\(\)](#), [odds\\_ratio\(\)](#), [odds\(\)](#)

**Examples**

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

---

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

---

**Description**

Replaces an object with the inverse log odds of value.

**Usage**

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

**Arguments**

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

**Value**

Called for the side effect of updating x.

**See Also**

Other odds: [inv\\_odds\(\)](#), [log\\_odds\\_ratio\(\)](#), [log\\_odds\(\)](#), [odds<-\(\)](#), [odds\\_ratio\(\)](#), [odds\(\)](#)

**Examples**

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

---

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

---

**Description**

Calculates the log odds ratio for two probabilities.

**Usage**

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

**Arguments**

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

**Value**

A numeric object of the log odds ratios.

**See Also**

Other odds: [inv\\_odds\(\)](#), [log\\_odds<-\(\)](#), [log\\_odds\(\)](#), [odds<-\(\)](#), [odds\\_ratio\(\)](#), [odds\(\)](#)

**Examples**

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

---

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

---

**Description**

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

**Usage**

```
log_odds_ratio2(x)
```

**Arguments**

x                    A numeric vector of length 2.

**Value**

A number.

**See Also**

Other odds fun2: [odds\\_ratio2\(\)](#)

**Examples**

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

---

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

---

**Description**

Coerce an R object to a numeric atomic object.

**Usage**

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

```
## S3 method for class 'hms'  
numericise(x, ...)  
  
## S3 method for class 'matrix'  
numericise(x, ...)  
  
## S3 method for class 'array'  
numericise(x, ...)  
  
## S3 method for class 'data.frame'  
numericise(x, ...)
```

### Arguments

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

### Details

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

### Value

A numeric atomic object.

### Methods (by class)

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

### Examples

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



```
numericise(2L)

# double
numericise(c(1, 3))

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

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

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

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

# matrix
numericise(matrix(TRUE))

# array
numericise(array(TRUE))

# data.frame
numericise(data.frame(
  logical = c(TRUE, FALSE, NA),
  integer = 1:3,
  numeric = c(4, 10, NA),
  factor = as.factor(c("c", "A", "green"))
))
```

---

odds

*Odds*

---

### **Description**

Calculates the odds for probabilities.

### **Usage**

```
odds(x)
```

### **Arguments**

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

### **Value**

A numeric object of the the odds for each probability.

**See Also**

Other odds: [inv\\_odds\(\)](#), [log\\_odds<-\(\)](#), [log\\_odds\\_ratio\(\)](#), [log\\_odds\(\)](#), [odds<-\(\)](#), [odds\\_ratio\(\)](#)

**Examples**

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

---

odds<- *Inverse Odds Transformation*

---

**Description**

Replaces an object with the inverse odds of value.

**Usage**

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

**Arguments**

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

**Value**

Called for the side effect of updating x.

**See Also**

Other odds: [inv\\_odds\(\)](#), [log\\_odds<-\(\)](#), [log\\_odds\\_ratio\(\)](#), [log\\_odds\(\)](#), [odds\\_ratio\(\)](#), [odds\(\)](#)

**Examples**

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

---

`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: [inv\\_odds\(\)](#), [log\\_odds<-\(\)](#), [log\\_odds\\_ratio\(\)](#), [log\\_odds\(\)](#), [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

*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\(\)](#), [ilog\(\)](#), [inv\\_logit\(\)](#), [invlogit\(\)](#), [log<-\(\)](#), [logit<-\(\)](#), [logit\(\)](#), [pow\(\)](#), [step\(\)](#)

**Examples**

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

pow

*Power*

---

**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

*Bayesian P-Value*

---

**Description**

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

A numeric vector of the random samples.

**See Also**

Other ran\_dist: [ran\\_beta\\_binom\(\)](#), [ran\\_binom\(\)](#), [ran\\_gamma\\_pois\\_zi\(\)](#), [ran\\_gamma\\_pois\(\)](#), [ran\\_gamma\(\)](#), [ran\\_lnorm\(\)](#), [ran\\_neg\\_binom\(\)](#), [ran\\_norm\(\)](#), [ran\\_pois\\_zi\(\)](#), [ran\\_pois\(\)](#), [ran\\_student\(\)](#)

**Examples**

```
ran_bern(10)
```

---

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

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

A numeric vector of the random samples.

**See Also**

Other ran\_dist: [ran\\_bern\(\)](#), [ran\\_binom\(\)](#), [ran\\_gamma\\_pois\\_zi\(\)](#), [ran\\_gamma\\_pois\(\)](#), [ran\\_gamma\(\)](#), [ran\\_lnorm\(\)](#), [ran\\_neg\\_binom\(\)](#), [ran\\_norm\(\)](#), [ran\\_pois\\_zi\(\)](#), [ran\\_pois\(\)](#), [ran\\_student\(\)](#)

**Examples**

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

---

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

---

**Description**

Binomial Random Samples

**Usage**

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

**Arguments**

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

**Value**

A numeric vector of the random samples.

**See Also**

Other ran\_dist: [ran\\_bern\(\)](#), [ran\\_beta\\_binom\(\)](#), [ran\\_gamma\\_pois\\_zi\(\)](#), [ran\\_gamma\\_pois\(\)](#), [ran\\_gamma\(\)](#), [ran\\_lnorm\(\)](#), [ran\\_neg\\_binom\(\)](#), [ran\\_norm\(\)](#), [ran\\_pois\\_zi\(\)](#), [ran\\_pois\(\)](#), [ran\\_student\(\)](#)

**Examples**

```
ran_binom(10)
```

---

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

---

**Description**

Gamma Random Samples

**Usage**

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

**Arguments**

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

**Value**

A numeric vector of the random samples.

**See Also**

Other `ran_dist`: [ran\\_bern\(\)](#), [ran\\_beta\\_binom\(\)](#), [ran\\_binom\(\)](#), [ran\\_gamma\\_pois\\_zi\(\)](#), [ran\\_gamma\\_pois\(\)](#), [ran\\_lnorm\(\)](#), [ran\\_neg\\_binom\(\)](#), [ran\\_norm\(\)](#), [ran\\_pois\\_zi\(\)](#), [ran\\_pois\(\)](#), [ran\\_student\(\)](#)

**Examples**

```
ran_gamma(10)
```

---

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

---

**Description**

Gamma-Poisson Random Samples

**Usage**

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

**Arguments**

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

**Value**

A numeric vector of the random samples.

**See Also**

Other `ran_dist`: [ran\\_bern\(\)](#), [ran\\_beta\\_binom\(\)](#), [ran\\_binom\(\)](#), [ran\\_gamma\\_pois\\_zi\(\)](#), [ran\\_gamma\(\)](#), [ran\\_lnorm\(\)](#), [ran\\_neg\\_binom\(\)](#), [ran\\_norm\(\)](#), [ran\\_pois\\_zi\(\)](#), [ran\\_pois\(\)](#), [ran\\_student\(\)](#)

**Examples**

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

---

ran\_gamma\_pois\_zi      *Zero-Inflated Gamma-Poisson Random Samples*

---

**Description**

Zero-Inflated Gamma-Poisson Random Samples

**Usage**

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

**Arguments**

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

**Value**

A numeric vector of the random samples.

**See Also**

Other ran\_dist: [ran\\_bern\(\)](#), [ran\\_beta\\_binom\(\)](#), [ran\\_binom\(\)](#), [ran\\_gamma\\_pois\(\)](#), [ran\\_gamma\(\)](#), [ran\\_lnorm\(\)](#), [ran\\_neg\\_binom\(\)](#), [ran\\_norm\(\)](#), [ran\\_pois\\_zi\(\)](#), [ran\\_pois\(\)](#), [ran\\_student\(\)](#)

**Examples**

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

---

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**

A numeric vector of the random samples.

**See Also**

Other ran\_dist: [ran\\_bern\(\)](#), [ran\\_beta\\_binom\(\)](#), [ran\\_binom\(\)](#), [ran\\_gamma\\_pois\\_zi\(\)](#), [ran\\_gamma\\_pois\(\)](#), [ran\\_gamma\(\)](#), [ran\\_neg\\_binom\(\)](#), [ran\\_norm\(\)](#), [ran\\_pois\\_zi\(\)](#), [ran\\_pois\(\)](#), [ran\\_student\(\)](#)

**Examples**

```
ran_lnorm(10)
```

---

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

---

**Description**

Identical to Gamma-Poisson Random Samples.

**Usage**

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

**Arguments**

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

**Value**

A numeric vector of the random samples.

**See Also**

Other ran\_dist: [ran\\_bern\(\)](#), [ran\\_beta\\_binom\(\)](#), [ran\\_binom\(\)](#), [ran\\_gamma\\_pois\\_zi\(\)](#), [ran\\_gamma\\_pois\(\)](#), [ran\\_gamma\(\)](#), [ran\\_lnorm\(\)](#), [ran\\_norm\(\)](#), [ran\\_pois\\_zi\(\)](#), [ran\\_pois\(\)](#), [ran\\_student\(\)](#)

**Examples**

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

---

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

---

**Description**

Normal Random Samples

**Usage**

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

**Arguments**

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

**Value**

A numeric vector of the random samples.

**See Also**

Other ran\_dist: [ran\\_bern\(\)](#), [ran\\_beta\\_binom\(\)](#), [ran\\_binom\(\)](#), [ran\\_gamma\\_pois\\_zi\(\)](#), [ran\\_gamma\\_pois\(\)](#), [ran\\_gamma\(\)](#), [ran\\_lnorm\(\)](#), [ran\\_neg\\_binom\(\)](#), [ran\\_pois\\_zi\(\)](#), [ran\\_pois\(\)](#), [ran\\_student\(\)](#)

**Examples**

```
ran_norm(10)
```

---

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

---

**Description**

Poisson Random Samples

**Usage**

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

**Arguments**

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



**Value**

A numeric vector of the random samples.

**See Also**

Other ran\_dist: [ran\\_bern\(\)](#), [ran\\_beta\\_binom\(\)](#), [ran\\_binom\(\)](#), [ran\\_gamma\\_pois\\_zi\(\)](#), [ran\\_gamma\\_pois\(\)](#), [ran\\_gamma\(\)](#), [ran\\_lnorm\(\)](#), [ran\\_neg\\_binom\(\)](#), [ran\\_norm\(\)](#), [ran\\_pois\\_zi\(\)](#), [ran\\_student\(\)](#)

**Examples**

```
ran_pois(10)
```

---

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

---

**Description**

Zero-Inflated Poisson Random Samples

**Usage**

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

**Arguments**

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

**Value**

A numeric vector of the random samples.

**See Also**

Other ran\_dist: [ran\\_bern\(\)](#), [ran\\_beta\\_binom\(\)](#), [ran\\_binom\(\)](#), [ran\\_gamma\\_pois\\_zi\(\)](#), [ran\\_gamma\\_pois\(\)](#), [ran\\_gamma\(\)](#), [ran\\_lnorm\(\)](#), [ran\\_neg\\_binom\(\)](#), [ran\\_norm\(\)](#), [ran\\_pois\(\)](#), [ran\\_student\(\)](#)

**Examples**

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

---

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

---

**Description**

Student's t Random Samples

**Usage**

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

**Arguments**

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

**Value**

A numeric vector of the random samples.

**See Also**

Other ran\_dist: [ran\\_bern\(\)](#), [ran\\_beta\\_binom\(\)](#), [ran\\_binom\(\)](#), [ran\\_gamma\\_pois\\_zi\(\)](#), [ran\\_gamma\\_pois\(\)](#), [ran\\_gamma\(\)](#), [ran\\_lnorm\(\)](#), [ran\\_neg\\_binom\(\)](#), [ran\\_norm\(\)](#), [ran\\_pois\\_zi\(\)](#), [ran\\_pois\(\)](#)

**Examples**

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

---

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

---

**Description**

Bernoulli Residuals

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding residuals.

**See Also**

Other res\_dist: [res\\_beta\\_binom\(\)](#), [res\\_binom\(\)](#), [res\\_gamma\\_pois\\_zi\(\)](#), [res\\_gamma\\_pois\(\)](#), [res\\_gamma\(\)](#), [res\\_lnorm\(\)](#), [res\\_neg\\_binom\(\)](#), [res\\_norm\(\)](#), [res\\_pois\\_zi\(\)](#), [res\\_pois\(\)](#), [res\\_student\(\)](#)

**Examples**

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

---

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

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding residuals.

**See Also**

Other res\_dist: [res\\_bern\(\)](#), [res\\_binom\(\)](#), [res\\_gamma\\_pois\\_zi\(\)](#), [res\\_gamma\\_pois\(\)](#), [res\\_gamma\(\)](#), [res\\_lnorm\(\)](#), [res\\_neg\\_binom\(\)](#), [res\\_norm\(\)](#), [res\\_pois\\_zi\(\)](#), [res\\_pois\(\)](#), [res\\_student\(\)](#)

**Examples**

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

---

res\_binom

*Binomial Residuals*


---

**Description**

Binomial Residuals

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding residuals.

**See Also**

Other res\_dist: [res\\_bern\(\)](#), [res\\_beta\\_binom\(\)](#), [res\\_gamma\\_pois\\_zi\(\)](#), [res\\_gamma\\_pois\(\)](#), [res\\_gamma\(\)](#), [res\\_lnorm\(\)](#), [res\\_neg\\_binom\(\)](#), [res\\_norm\(\)](#), [res\\_pois\\_zi\(\)](#), [res\\_pois\(\)](#), [res\\_student\(\)](#)

**Examples**

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

---

 res\_gamma

*Gamma Residuals*


---

**Description**

Gamma Residuals

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding residuals.

**See Also**

Other res\_dist: [res\\_bern\(\)](#), [res\\_beta\\_binom\(\)](#), [res\\_binom\(\)](#), [res\\_gamma\\_pois\\_zi\(\)](#), [res\\_gamma\\_pois\(\)](#), [res\\_lnorm\(\)](#), [res\\_neg\\_binom\(\)](#), [res\\_norm\(\)](#), [res\\_pois\\_zi\(\)](#), [res\\_pois\(\)](#), [res\\_student\(\)](#)

**Examples**

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

---

res\_gamma\_pois      *Gamma-Poisson Residuals*

---

**Description**

Gamma-Poisson Residuals

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding residuals.

**See Also**

Other res\_dist: [res\\_bern\(\)](#), [res\\_beta\\_binom\(\)](#), [res\\_binom\(\)](#), [res\\_gamma\\_pois\\_zi\(\)](#), [res\\_gamma\(\)](#), [res\\_lnorm\(\)](#), [res\\_neg\\_binom\(\)](#), [res\\_norm\(\)](#), [res\\_pois\\_zi\(\)](#), [res\\_pois\(\)](#), [res\\_student\(\)](#)

**Examples**

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

---

res\_gamma\_pois\_zi      *Zero-Inflated Gamma-Poisson Residuals*

---

**Description**

Zero-Inflated Gamma-Poisson Residuals

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding residuals.

**See Also**

Other res\_dist: [res\\_bern\(\)](#), [res\\_beta\\_binom\(\)](#), [res\\_binom\(\)](#), [res\\_gamma\\_pois\(\)](#), [res\\_gamma\(\)](#), [res\\_lnorm\(\)](#), [res\\_neg\\_binom\(\)](#), [res\\_norm\(\)](#), [res\\_pois\\_zi\(\)](#), [res\\_pois\(\)](#), [res\\_student\(\)](#)

**Examples**

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

---

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

---

**Description**

Log-Normal Residuals

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding residuals.

**See Also**

Other res\_dist: [res\\_bern\(\)](#), [res\\_beta\\_binom\(\)](#), [res\\_binom\(\)](#), [res\\_gamma\\_pois\\_zi\(\)](#), [res\\_gamma\\_pois\(\)](#), [res\\_gamma\(\)](#), [res\\_neg\\_binom\(\)](#), [res\\_norm\(\)](#), [res\\_pois\\_zi\(\)](#), [res\\_pois\(\)](#), [res\\_student\(\)](#)

**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 dispersion for the mixture models (student, gamma-Poisson and beta-binomial).
type	A string of the residual type. 'raw' for raw residuals 'dev' for deviance residuals and 'data' for the data.
simulate	A flag specifying whether to simulate residuals.

**Value**

An numeric vector of the corresponding residuals.



**See Also**

Other res\_dist: [res\\_bern\(\)](#), [res\\_beta\\_binom\(\)](#), [res\\_binom\(\)](#), [res\\_gamma\\_pois\\_zi\(\)](#), [res\\_gamma\\_pois\(\)](#), [res\\_gamma\(\)](#), [res\\_lnorm\(\)](#), [res\\_norm\(\)](#), [res\\_pois\\_zi\(\)](#), [res\\_pois\(\)](#), [res\\_student\(\)](#)

**Examples**

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

---

 res\_norm

*Normal Residuals*


---

**Description**

Normal Residuals

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding residuals.

**See Also**

Other res\_dist: [res\\_bern\(\)](#), [res\\_beta\\_binom\(\)](#), [res\\_binom\(\)](#), [res\\_gamma\\_pois\\_zi\(\)](#), [res\\_gamma\\_pois\(\)](#), [res\\_gamma\(\)](#), [res\\_lnorm\(\)](#), [res\\_neg\\_binom\(\)](#), [res\\_pois\\_zi\(\)](#), [res\\_pois\(\)](#), [res\\_student\(\)](#)

**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\\_beta\\_binom\(\)](#), [res\\_binom\(\)](#), [res\\_gamma\\_pois\\_zi\(\)](#), [res\\_gamma\\_pois\(\)](#), [res\\_gamma\(\)](#), [res\\_lnorm\(\)](#), [res\\_neg\\_binom\(\)](#), [res\\_norm\(\)](#), [res\\_pois\\_zi\(\)](#), [res\\_student\(\)](#)

**Examples**

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

---

 res\_pois\_zi

*Zero-Inflated Poisson Residuals*


---

**Description**

Zero-Inflated Poisson Residuals

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding residuals.

**See Also**

Other res\_dist: [res\\_bern\(\)](#), [res\\_beta\\_binom\(\)](#), [res\\_binom\(\)](#), [res\\_gamma\\_pois\\_zi\(\)](#), [res\\_gamma\\_pois\(\)](#), [res\\_gamma\(\)](#), [res\\_lnorm\(\)](#), [res\\_neg\\_binom\(\)](#), [res\\_norm\(\)](#), [res\\_pois\(\)](#), [res\\_student\(\)](#)

**Examples**

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

---

res\_student

*Student's t Residuals*


---

**Description**

Student's t Residuals

**Usage**

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

**Arguments**

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

**Value**

An numeric vector of the corresponding residuals.

**See Also**

Other `res_dist`: [res\\_bern\(\)](#), [res\\_beta\\_binom\(\)](#), [res\\_binom\(\)](#), [res\\_gamma\\_pois\\_zi\(\)](#), [res\\_gamma\\_pois\(\)](#), [res\\_gamma\(\)](#), [res\\_lnorm\(\)](#), [res\\_neg\\_binom\(\)](#), [res\\_norm\(\)](#), [res\\_pois\\_zi\(\)](#), [res\\_pois\(\)](#)

**Examples**

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

---

 sextreme

*Extreme Surprisal*


---

**Description**

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

**Usage**

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

**Arguments**

`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\(\)](#), [pzeros\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr\\_mean\(\)](#), [xtr\\_median\(\)](#), [xtr\\_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

**Examples**

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

---

step	<i>Step</i>
------	-------------

---

**Description**

Step

**Usage**

```
step(x)
```

**Arguments**

x	A numeric atomic object.
---	--------------------------

**Value**

A logical value.

**See Also**

Other translations: [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**

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

**Details**

By default it returns the 95% credible limit which corresponds to the 97.5% quantile.

**Value**

A number.

**See Also**

Other summary: [kurtosis\(\)](#), [lower\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [variance\(\)](#), [xtr\\_mean\(\)](#), [xtr\\_median\(\)](#), [xtr\\_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

**Examples**

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

---

variance	<i>Variance</i>
----------	-----------------

---

**Description**

Variance

**Usage**

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

**Arguments**

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

**Value**

A number.

**See Also**

Other summary: [kurtosis\(\)](#), [lower\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [xtr\\_mean\(\)](#), [xtr\\_median\(\)](#), [xtr\\_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

**Examples**

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

---

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

---

**Description**

Mean

**Usage**

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

**Arguments**

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

**Value**

A number.

**See Also**

Other summary: [kurtosis\(\)](#), [lower\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr\\_median\(\)](#), [xtr\\_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

**Examples**

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



---

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

---

**Description**

Median

**Usage**

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

**Arguments**

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

**Value**

A number.

**See Also**

Other summary: [kurtosis\(\)](#), [lower\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr\\_mean\(\)](#), [xtr\\_sd\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

**Examples**

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

---

xtr_sd	<i>Standard Deviation</i>
--------	---------------------------

---

**Description**

Standard Deviation

**Usage**

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

**Arguments**

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

**Value**

A number.

**See Also**

Other summary: [kurtosis\(\)](#), [lower\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr\\_mean\(\)](#), [xtr\\_median\(\)](#), [zeros\(\)](#), [zscore\(\)](#)

**Examples**

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

---

zeros

*Zeros*

---

**Description**

The number of zeros in an numeric object.

**Usage**

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

**Arguments**

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

**Value**

A non-negative integer.

**See Also**

Other summary: [kurtosis\(\)](#), [lower\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr\\_mean\(\)](#), [xtr\\_median\(\)](#), [xtr\\_sd\(\)](#), [zscore\(\)](#)

**Examples**

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

---

zscore	<i>Z-Score</i>
--------	----------------

---

**Description**

The Bayesian z-score is here defined as the number of standard deviations from the mean estimate to zero.

**Usage**

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

**Arguments**

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

**Value**

A number.

**See Also**

Other summary: [kurtosis\(\)](#), [lower\(\)](#), [pvalue\(\)](#), [pzeros\(\)](#), [skewness\(\)](#), [svalue\(\)](#), [upper\(\)](#), [variance\(\)](#), [xtr\\_mean\(\)](#), [xtr\\_median\(\)](#), [xtr\\_sd\(\)](#), [zeros\(\)](#)

**Examples**

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

# Index

- \* **bern**
  - dbern, 7
- \* **dev\_dist # make live when complete**
  - dev\_gamma\_pois\_zi, 12
- \* **dev\_dist**
  - dev\_bern, 8
  - dev\_beta\_binom, 9
  - dev\_binom, 10
  - dev\_gamma, 11
  - dev\_gamma\_pois, 11
  - dev\_lnorm, 13
  - dev\_neg\_binom, 14
  - dev\_norm, 14
  - dev\_pois, 15
  - dev\_pois\_zi, 16
  - dev\_student, 17
- \* **fill**
  - fill\_all, 18
  - fill\_na, 19
- \* **log\_lik\_dist**
  - log\_lik\_bern, 27
  - log\_lik\_beta\_binom, 28
  - log\_lik\_binom, 29
  - log\_lik\_gamma, 29
  - log\_lik\_gamma\_pois, 30
  - log\_lik\_gamma\_pois\_zi, 31
  - log\_lik\_lnorm, 31
  - log\_lik\_neg\_binom, 32
  - log\_lik\_norm, 33
  - log\_lik\_pois, 34
  - log\_lik\_pois\_zi, 34
  - log\_lik\_student, 35
- \* **odds fun2**
  - log\_odds\_ratio2, 38
  - odds\_ratio2, 43
- \* **odds**
  - inv\_odds, 24
  - log\_odds, 36
  - log\_odds<-, 36
  - log\_odds\_ratio, 37
  - odds, 41
  - odds<-, 42
  - odds\_ratio, 43
- \* **proportional fun2**
  - proportional\_change2, 47
  - proportional\_difference2, 48
- \* **proportional**
  - proportional\_change, 46
  - proportional\_difference, 48
- \* **ran\_dist**
  - ran\_bern, 50
  - ran\_beta\_binom, 51
  - ran\_binom, 52
  - ran\_gamma, 52
  - ran\_gamma\_pois, 53
  - ran\_gamma\_pois\_zi, 54
  - ran\_lnorm, 54
  - ran\_neg\_binom, 55
  - ran\_norm, 56
  - ran\_pois, 56
  - ran\_pois\_zi, 57
  - ran\_student, 58
- \* **res\_dist**
  - res\_bern, 58
  - res\_beta\_binom, 59
  - res\_binom, 60
  - res\_gamma, 61
  - res\_gamma\_pois, 62
  - res\_gamma\_pois\_zi, 62
  - res\_lnorm, 63
  - res\_neg\_binom, 64
  - res\_norm, 65
  - res\_pois, 66
  - res\_pois\_zi, 66
  - res\_student, 67
- \* **residuals**
  - pextreme, 44
  - sextreme, 68

- \* **summary**
  - kurtosis, 24
  - lower, 38
  - pvalue, 49
  - pzeros, 50
  - skewness, 69
  - svalue, 70
  - upper, 71
  - variance, 71
  - xtr\_mean, 72
  - xtr\_median, 73
  - xtr\_sd, 73
  - zeros, 74
  - zscore, 75
- \* **translations**
  - fabs, 17
  - ilog, 21
  - ilogit, 22
  - inv\_logit, 23
  - invlogit, 22
  - log<-, 25
  - logit, 26
  - logit<-, 26
  - phi, 45
  - pow, 46
  - step, 69
- abs, 18
- as\_list, 4
- as\_list\_unnamed, 4
- chk\_index, 5
- chk\_indices, 6
- chk\_pars, 6
- dbern, 7
- dev\_bern, 8, 9–17
- dev\_beta\_binom, 9, 9, 10–17
- dev\_binom, 9, 10, 11–17
- dev\_gamma, 9, 10, 11, 12–17
- dev\_gamma\_pois, 9–11, 11, 13–17
- dev\_gamma\_pois\_zi, 12
- dev\_lnorm, 9–12, 13, 14–17
- dev\_neg\_binom, 9–13, 14, 15–17
- dev\_norm, 9–14, 14, 15–17
- dev\_pois, 9–15, 15, 16, 17
- dev\_pois\_zi, 9–15, 16, 17
- dev\_student, 9–16, 17
- exp, 21, 25
- fabs, 17, 21–23, 25–27, 45, 46, 70
- fill\_all, 18, 20
- fill\_na, 19, 19
- ilog, 18, 21, 22, 23, 25–27, 45, 46, 70
- ilogit, 18, 21, 22, 23, 25–27, 45, 46, 70
- inv\_logit, 18, 21–23, 23, 25–27, 45, 46, 70
- inv\_odds, 24, 36, 37, 42, 43
- invlogit, 18, 21, 22, 22, 23, 25–27, 45, 46, 70
- kurtosis, 24, 39, 49, 50, 69–75
- log<-, 25
- log\_lik\_bern, 27, 28–35
- log\_lik\_beta\_binom, 27, 28, 29–35
- log\_lik\_binom, 27, 28, 29, 30–35
- log\_lik\_gamma, 27–29, 29, 30–35
- log\_lik\_gamma\_pois, 27–30, 30, 31–35
- log\_lik\_gamma\_pois\_zi, 27–30, 31, 32–35
- log\_lik\_lnorm, 27–31, 31, 32–35
- log\_lik\_neg\_binom, 27–32, 32, 33–35
- log\_lik\_norm, 27–32, 33, 34, 35
- log\_lik\_pois, 27–33, 34, 35
- log\_lik\_pois\_zi, 27–34, 34, 35
- log\_lik\_student, 27–35, 35
- log\_odds, 24, 36, 37, 42, 43
- log\_odds<-, 36
- log\_odds\_ratio, 24, 36, 37, 37, 42, 43
- log\_odds\_ratio2, 38, 44
- logit, 18, 21–23, 25, 26, 27, 45, 46, 70
- logit<-, 26
- lower, 25, 38, 49, 50, 69–75
- numericise, 39
- numericize (numericise), 39
- odds, 24, 36, 37, 41, 42, 43
- odds<-, 42
- odds\_ratio, 24, 36, 37, 42, 43
- odds\_ratio2, 38, 43
- par\_pattern, 44
- pbern (dbern), 7
- pextreme, 44, 68
- phi, 18, 21–23, 25–27, 45, 46, 70
- pow, 18, 21–23, 25–27, 45, 46, 70
- proportional\_change, 46, 48
- proportional\_change2, 47, 49
- proportional\_difference, 47, 48
- proportional\_difference2, 47, 48

- pvalue, [25](#), [39](#), [49](#), [50](#), [69–75](#)
- pzeros, [25](#), [39](#), [49](#), [50](#), [69–75](#)
  
- qbern (dbern), [7](#)
  
- ran\_bern, [50](#), [51–58](#)
- ran\_beta\_binom, [51](#), [51](#), [52–58](#)
- ran\_binom, [51](#), [52](#), [53–58](#)
- ran\_gamma, [51](#), [52](#), [52](#), [53–58](#)
- ran\_gamma\_pois, [51–53](#), [53](#), [54–58](#)
- ran\_gamma\_pois\_zi, [51–53](#), [54](#), [55–58](#)
- ran\_lnorm, [51–54](#), [54](#), [55–58](#)
- ran\_neg\_binom, [51–55](#), [55](#), [56–58](#)
- ran\_norm, [51–55](#), [56](#), [57](#), [58](#)
- ran\_pois, [51–56](#), [56](#), [57](#), [58](#)
- ran\_pois\_zi, [51–57](#), [57](#), [58](#)
- ran\_student, [51–57](#), [58](#)
- rbern (dbern), [7](#)
- res\_bern, [58](#), [60–68](#)
- res\_beta\_binom, [59](#), [59](#), [61–68](#)
- res\_binom, [59](#), [60](#), [60](#), [61–68](#)
- res\_gamma, [59–61](#), [61](#), [62–68](#)
- res\_gamma\_pois, [59–61](#), [62](#), [63–68](#)
- res\_gamma\_pois\_zi, [59–62](#), [62](#), [64–68](#)
- res\_lnorm, [59–63](#), [63](#), [65–68](#)
- res\_neg\_binom, [59–64](#), [64](#), [65–68](#)
- res\_norm, [59–65](#), [65](#), [66–68](#)
- res\_pois, [59–65](#), [66](#), [67](#), [68](#)
- res\_pois\_zi, [59–66](#), [66](#), [68](#)
- res\_student, [59–67](#), [67](#)
  
- sextreme, [45](#), [68](#)
- skewness, [25](#), [39](#), [49](#), [50](#), [69](#), [70–75](#)
- stats::plogis(), [22](#), [23](#)
- stats::pnorm(), [45](#)
- stats::qlogis(), [26](#)
- step, [18](#), [21–23](#), [25–27](#), [45](#), [46](#), [69](#)
- svalue, [25](#), [39](#), [49](#), [50](#), [69](#), [70](#), [71–75](#)
  
- upper, [25](#), [39](#), [49](#), [50](#), [69](#), [70](#), [71](#), [72–75](#)
  
- variance, [25](#), [39](#), [49](#), [50](#), [69–71](#), [71](#), [72–75](#)
- vld\_index (chk\_index), [5](#)
- vld\_indices (chk\_indices), [6](#)
- vld\_pars (chk\_pars), [6](#)
  
- xtr\_mean, [25](#), [39](#), [49](#), [50](#), [69–72](#), [72](#), [73–75](#)
- xtr\_median, [25](#), [39](#), [49](#), [50](#), [69–72](#), [73](#), [74](#), [75](#)
- xtr\_sd, [25](#), [39](#), [49](#), [50](#), [69–73](#), [73](#), [74](#), [75](#)