Package 'extras'

September 26, 2020

Title Helper Functions for Bayesian Analyses

Version 0.1.0

Description Functions to 'numericise' 'R' objects (coerce to numeric objects) and summarise 'MCMC' (Monte Carlo Markov Chain) samples 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 (>= 3.3)

Imports chk, stats

Suggests covr, hms, knitr, testthat, tibble

Encoding UTF-8

Language en-US

LazyData true

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>), Poisson Consulting [cph, fnd]

Maintainer Joe Thorley <joe@poissonconsulting.ca>

Repository CRAN

Date/Publication 2020-09-26 04:30:08 UTC

R topics documented:

`as_list_unnamed` *As List*

Description

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

Usage

```
as_list_unnamed(x, ...)

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

Arguments

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

Value

A list.

Examples

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

chk_index *Check Index*

Description

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

Usage

```
chk_index(x, x_name = NULL)

vld_index(x)
```

Arguments

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"))
```

---

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.

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 an object's missing values while preserving the object's 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.

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: `log<-()`, `logit<-()`, `logit()`, `phi()`, `pow()`

Examples

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

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: `ilogit()`, `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: ilogit(), 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: ilogit(), log<-(), logit(), phi(), pow()

Examples

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

`lower` *Lower Credible Limit*

Description

Calculates the quantile-based lower credible limit.

Usage

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

Arguments

`x`	A numeric vector of MCMC values.
`conf_level`	A numeric scalar between 0 and 1 specifying the confidence level.

Details

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

Value

A number.

See Also

Other summary: `pvalue()`, `svalue()`, `upper()`, `zscore()`

Examples

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

`numericise` *Numericise (or Numericize)*

Description

Coerce an R object to a numeric atomic object.

Usage

```
numericise(x, ...)

numericize(x, ...)

## S3 method for class 'logical'
numericise(x, ...)

## S3 method for class 'integer'
numericise(x, ...)

## S3 method for class 'double'
numericise(x, ...)

## S3 method for class 'factor'
numericise(x, ...)

## S3 method for class 'Date'
numericise(x, ...)

## S3 method for class 'POSIXct'
numericise(x, ...)

## S3 method for class 'hms'
numericise(x, ...)

## S3 method for class 'matrix'
numericise(x, ...)

## S3 method for class 'array'
numericise(x, ...)

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

Arguments

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

Details

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

Value

A numeric atomic object.

Methods (by class)

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

Examples

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

# integer
numericise(2L)

# double
numericise(c(1, 3))

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

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

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

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

# matrix
numericise(matrix(TRUE))
```

```
# array
numericise(array(TRUE))

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

`par_pattern` *Parameter Pattern*

Description

Parameter Pattern

Usage

```
par_pattern()
```

Value

A string of the regular expression for a parameter name.

Examples

```
par_pattern()
```

`pextreme` *Extreme Probability*

Description

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

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: `ilogit()`, `log<-()`, `logit<-()`, `logit()`, `pow()`

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: `ilogit()`, `log<-()`, `logit<-()`, `logit()`, `phi()`

Examples

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

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 0 (Kery and Schaub 2011). In other words a p-value of 0.05 indicates that the 95% CRI just includes 0.

Usage

```
pvalue(x)
```

Arguments

x	A numeric vector of MCMC 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: `lower()`, `svalue()`, `upper()`, `zscore()`

Examples

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

---

sextreme *Extreme Surprisal*

Description

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

Usage

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

Arguments

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

Details

Useful for treating the cdf probabilities as residuals particularly when directional = TRUE.

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)
```

---

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)
```

Arguments

x	A numeric object of MCMC 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. https://doi.org/10.1080/00031305.2018.1529625.

See Also

Other summary: lower(), pvalue(), upper(), zscore()

Examples

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

upper *Upper Credible Limit*

Description

Calculates the quantile-based upper credible limit.

Usage

```
upper(x, conf_level = 0.95)
```

Arguments

x	A numeric vector of MCMC values.
conf_level	A numeric scalar between 0 and 1 specifying the confidence level.

Details

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

Value

A number.

See Also

Other summary: `lower()`, `pvalue()`, `svalue()`, `zscore()`

Examples

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

zscore *Z-Score*

Description

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

Usage

```
zscore(x)
```

Arguments

x	A numeric object of MCMC values.

Value

A number.

See Also

Other summary: `lower()`, `pvalue()`, `svalue()`, `upper()`

Examples

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

Index