Package ‘mcmcr’

June 27, 2019

Title  Manipulate MCMC Samples
Version 0.2.0
Description Functions and classes to store, manipulate and summarise
Monte Carlo Markov Chain (MCMC) samples. For more information see
License MIT + file LICENSE
Depends R (>= 3.5)
Imports abind, err, checkr, coda, utils, stats
Suggests covr, graphics, testthat
URL https://github.com/poissonconsulting/mcmcr
BugReports https://github.com/poissonconsulting/mcmcr/issues
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1
Language en-US
NeedsCompilation no
Author Joe Thorley [aut, cre] (<https://orcid.org/0000-0002-7683-4592>)
Maintainer Joe Thorley <joe@poissonconsulting.ca>
Repository CRAN
Date/Publication 2019-06-27 16:30:03 UTC

R topics documented:

anyNA ................................................................. 3
as.marray ........................................................... 4
as.mcmc .............................................................. 4
as.mcmc.list ....................................................... 5
as.mcmarray ....................................................... 6
as.mcmcr .......................................................... 7
# R topics documented:

- `as.mcmcrs` ................................................. 8
- `as.term` .................................................. 9
- `bind_chains` ............................................ 10
- `bind_dimensions` ....................................... 11
- `bind_dimensions_n` ..................................... 12
- `bind_iterations` ......................................... 12
- `bind_parameters` ......................................... 13
- `check_mcmccarray` ...................................... 14
- `check_mcmcr` ........................................... 15
- `coef` .................................................... 16
- `collapse_chains` ........................................ 17
- `combine_dimensions` ..................................... 18
- `combine_samples` ........................................ 19
- `combine_samples_n` ..................................... 20
- `converged` ............................................... 20
- `dims` ..................................................... 22
- `esr` ..................................................... 22
- `ess` ...................................................... 24
- `estimates` ................................................ 24
- `is.mcmccarray` .......................................... 25
- `is.mcmccr` ............................................... 25
- `is.mcmcrs` ............................................... 26
- `is.mcmcr` ............................................... 26
- `mcmccarray-object` ...................................... 28
- `mcmcr-object` ........................................... 28
- `mcmcrs` .................................................. 29
- `mcmcrs-object` .......................................... 29
- `mcmcr_example` .......................................... 30
- `mcmc_aperm` ............................................. 30
- `mcmc_map` ............................................... 31
- `nchains` .................................................. 32
- `ndims` .................................................... 34
- `niter` .................................................... 34
- `npars` .................................................... 35
- `npdims` ................................................... 36
- `npdims.default` ......................................... 36
- `nsams` .................................................... 37
- `nsims` .................................................... 38
- `nterms` .................................................... 39
- `parameters` .............................................. 40
- `pvalue` .................................................... 41
- `rhat` ...................................................... 42
- `sort` ...................................................... 43
- `split_chains` ............................................. 44
- `subset` .................................................... 45
- `tdims` ..................................................... 46
- `term` ...................................................... 46
anyNA

**Any Missing Values**

**Description**

Test whether there are any missing values.

**Usage**

```r
## S3 method for class 'mcarray'
anyna(x, recursive = FALSE)
```

```r
## S3 method for class 'mcmc'
anyna(x, recursive = FALSE)
```

```r
## S3 method for class 'mcmc.list'
anyna(x, recursive = FALSE)
```

```r
## S3 method for class 'mcmcarray'
anyna(x, recursive = FALSE)
```

```r
## S3 method for class 'mcmcr'
anyna(x, recursive = FALSE)
```

```r
## S3 method for class 'mcmcrs'
anyna(x, recursive = FALSE)
```

**Arguments**

- `x` The object to test.
- `recursive` Unused.

**Value**

A flag indicating whether there are any missing values.

**See Also**

`base::anyNA`
Examples

```r
anyNA(mcmcr_example)
anyNA(mcmcr_example$beta)
```

<table>
<thead>
<tr>
<th>as.mcmcrarray</th>
<th>Coerce to an mcmarray object</th>
</tr>
</thead>
</table>

Description

Coerces MCMC objects to an mcarray object.

Usage

```r
as.mcmarray(x, ...)
```

## S3 method for class 'mcmarray'
```r
as.mcmarray(x, ...)
```

Arguments

- `x` object to coerce.
- `...` Unused.

Methods (by class)

- `mcmarray`: Coerces mcmarray object to an mcarray object

Examples

```r
as.mcmarray(mcmcr_example$beta)
```

<table>
<thead>
<tr>
<th>as.mcmc</th>
<th>Coerce to an mcmc object</th>
</tr>
</thead>
</table>

Description

Coerces MCMC objects to an mcmc object.
Usage

## S3 method for class 'mcarray'
as.mcmc(x, ...)

## S3 method for class 'mcmcarray'
as.mcmc(x, ...)

## S3 method for class 'mcmcr'
as.mcmc(x, ...)

Arguments

x     object to coerce.
...

Methods (by class)

- `mcarray`: Coerces mcarray object (with 1 chain) to an mcmc object
- `mcmcarray`: Coerces mcmcarray object (with 1 chain) to an mcmc object
- `mcmcr`: Coerces mcmcr object (with 1 chain) to an mcmc object

See Also

coda::mcmc

Examples

as.mcmc(subset(mcmcr_example, chains = 1L))

---

**as.mcmc.list**

Coerce to an mcmc.list object

Description

Coerces MCMC objects to an mcmc.list object.

Usage

## S3 method for class 'mcmcarray'
as.mcmc.list(x, ...)

## S3 method for class 'mcmc'
as.mcmc.list(x, ...)

## S3 method for class 'mcmcr'
as.mcmc.list(x, ...)
Arguments

x object to coerce.

... Unused.

Methods (by class)

- mcmcarray: Coerces mcmcarray object to an mcmc.list object
- mcmc: Coerces mcmc object to an mcmc.list object
- mcmcr: Coerces mcmcr object to an mcmc.list object

as.mcmcarray Coerce to an mcmcarray object

Description

Coerces MCMC objects to an mcmcarray-object.

Usage

as.mcmcarray(x, ...)

## Default S3 method:
as.mcmcarray(x, ...)

## S3 method for class 'marray'
as.mcmcarray(x, ...)

## S3 method for class 'mcmc'
as.mcmcarray(x, ...)

## S3 method for class 'mcmc.list'
as.mcmcarray(x, ...)

## S3 method for class 'mcmcr'
as.mcmcarray(x, ...)

Arguments

x object to coerce.

... Unused.
Methods (by class)

- **default**: Coerces vector, matrix or array to an mcmcarray object
- **mcarray**: Coerces mcarray object to an mcmcarray object
- **mcmc**: Coerces mcmc object (with one parameter) to an mcmcarray object
- **mcmc.list**: Coerces mcmc.list object (with one parameter) to an mcmcarray object
- **mcmcr**: Coerces mcmcr object (with one parameter) to an mcmcarray object

Examples

```r
as.mcmcrate(as.mcarray(mcmcr_example$beta))
```

---

**as.mcmcrate**  
*Coerce to an mcmcrate object*

Description

Coerces MCMC objects to an `mcmcrate` object.

Usage

```r
as.mcmcrate(x, ...)
```

### S3 method for class 'list'

```r
as.mcmcrate(x, ...)
```

### S3 method for class 'mcarray'

```r
as.mcmcrate(x, ...)
```

### S3 method for class 'mcmc'

```r
as.mcmcrate(x, ...)
```

### S3 method for class 'mcmc.list'

```r
as.mcmcrate(x, ...)
```

### S3 method for class 'mcmcarray'

```r
as.mcmcrate(x, ...)
```

### S3 method for class 'list'

```r
as.mcmcrate(x, ...)
```

Arguments

- `x`  
  object to coerce.
- `...`  
  Unused.
Methods (by class)

- list: Coerces list (of mcmarray objects) to an mcmcr object
- mcarray: Coerces mcarray object to an mcmcr object
- mcmc: Coerces mcmc object to an mcmcr object
- mcmc.list: Coerces mcmc.list object to an mcmcr object
- mcmcarray: Coerces mcmcarray object to an mcmcr object
- list: Coerces list (of mcmcr objects with the same parameters, chains and iterations) to an mcmcrs object

Examples

```r
as.mcmcr(coda::as.mcmc.list(mcmcr_example))
```

---

as.mcmcrs

Coerce to an mcmcrs object

Description

Coerces MCMC objects to an mcmcrs-object.

Usage

```r
as.mcmcrs(x, ...)
```

Arguments

- `x` object to coerce.
- `...` Unused.

Examples

```r
as.mcmcrs(list(mcmcr_example))
```
Description
Coerces MCMC objects to a term vector.

Usage
as.term(x, ...)

## S3 method for class 'term'
as.character(x, ...)

## S3 method for class 'character'
as.term(x, ...)

## S3 method for class 'mcmc'
as.term(x, ...)

## S3 method for class 'mcmc.list'
as.term(x, ...)

## S3 method for class 'mcmcarray'
as.term(x, ...)

## S3 method for class 'mcmcr'
as.term(x, ...)

Arguments
x The object to coerce
...
Unused.

Methods (by class)
- **term**: Coerces term vector to a character vector
- **character**: Coerces character vector to a term vector
- **mcmc**: Coerces mcmc object to a term vector
- **mcmc.list**: Coerces mcmc.list object to a term vector
- **mcmcarray**: Coerces mcmcarray object to a term vector
- **mcmcr**: Coerces mcmcr object to a term vector

Examples
as.term(mcmcr_example)
bind_chains

Combine MCMC objects by chains.

Description

Combines two MCMC objects (with the same parameters and iterations) by chains.

Usage

bind_chains(x, x2, ...)

## S3 method for class 'mcarray'
bind_chains(x, x2, ...)

## S3 method for class 'mcmc'
bind_chains(x, x2, ...)

## S3 method for class 'mcmc.list'
bind_chains(x, x2, ...)

## S3 method for class 'mcmcarray'
bind_chains(x, x2, ...)

## S3 method for class 'mcmcr'
bind_chains(x, x2, ...)

Arguments

x            an MCMC object.

x2           a second MCMC object

...           Unused.

Methods (by class)

- mcarray: Binds two mcarray objects by their chains
- mcmc: Binds two mcmc objects by their chains
- mcmc.list: Binds two mcmc.list objects by their chains
- mcmcarray: Binds two mcmcarray objects by their chains
- mcmcr: Binds two mcmcr objects by their chains

Examples

bind_chains(mcmcr_example, mcmcr_example)
bind_dimensions

Combine two MCMC objects by dimensions

Description

Combines multiple MCMC objects (with the same parameters, chains and iterations) by parameter dimensions.

Usage

bind_dimensions(x, x2, along = NULL, ...)

## S3 method for class 'mcmcarray'
bind_dimensions(x, x2, along = NULL, ...)

## S3 method for class 'mcmcr'
bind_dimensions(x, x2, along = NULL, ...)

Arguments

x       An MCMC object.
x2      a second MCMC object.
along   A count (or NULL) indicating the parameter dimension to bind along.
...     Unused.

Methods (by class)

- mcmcarray: Binds two mcmcarray objects by their dimensions
- mcmcr: Binds two mcmcr objects by their dimensions

See Also

bind_dimensions_n

Examples

bind_dimensions(mcmcr_example, mcmcr_example)
bind_dimensions_n Combine multiple MCMC objects by parameter dimensions

Description

Combines multiple MCMC objects (with the same parameters, chains and iterations) by parameter dimensions.

Usage

bind_dimensions_n(...)  
## S3 method for class 'mcmcarray'
bind_dimensions_n(...)  
## S3 method for class 'mcmcr'
bind_dimensions_n(...)

Arguments

... one or more MCMC objects

Methods (by class)

• mcmcarray: Binds multiple mcmcarray objects by their dimensions  
• mcmcr: Binds multiple mcmcr objects by their dimensions

See Also

bind_dimensions

Examples

bind_dimensions_n(mcmcr_example, mcmcr_example, mcmcr_example)

bind_iterations Combine two MCMC objects by iterations

Description

Combines two MCMC objects (with the same parameters and chains) by iterations.
Usage

bind_iterations(x, x2, ...)

## S3 method for class 'mcarray'
bind_iterations(x, x2, ...)

## S3 method for class 'mcmc'
bind_iterations(x, x2, ...)

## S3 method for class 'mcmc.list'
bind_iterations(x, x2, ...)

## S3 method for class 'mcmcarray'
bind_iterations(x, x2, ...)

## S3 method for class 'mcmcr'
bind_iterations(x, x2, ...)

Arguments

x an MCMC object
x2 a second MCMC object
... unused

Methods (by class)

- mcarray: Binds two mcarray objects by their iterations
- mcmc: Binds two mcmc objects by their iterations
- mcmc.list: Binds two mcmc.list objects by their iterations
- mcmcarray: Binds two mcmcarray objects by their iterations
- mcmcr: Binds two mcmcr objects by their iterations

Examples

bind_iterations(mcmcr_exampleL mcmcr_example)

bind_parameters Combine two MCMC object by parameters

Description

Combines two MCMC objects (with the same chains and iterations) by their parameters.
bind_parameters(x, ...)

## S3 method for class 'mcmc'
bind_parameters(x, x2, ...)

## S3 method for class 'mcmc.list'
bind_parameters(x, x2, ...)

## S3 method for class 'mcmcr'
bind_parameters(x, x2, ...)

Arguments

- **x**: an MCMC object
- **...**: unused
- **x2**: a second MCMC object

Methods (by class)

- **mcmc**: Binds two mcmc objects by their parameters
- **mcmc.list**: Binds two mcmc.list objects by their parameters
- **mcmcr**: Binds two mcmcr objects by their parameters

Examples

bind_parameters(subset(mcmcr_example, parameters = "sigma"),
    subset(mcmcr_example, parameters = "beta"))

---

check_mcmarray  Check mcmarray

Description

Check mcmarray

Usage

check_mcmarray(x, x_name = substitute(x), error = TRUE)

Arguments

- **x**: The object to check.
- **x_name**: A string of the name of the object.
- **error**: A flag indicating whether to throw an informative error or immediately generate an informative message if the check fails.
check_mcmcr

Value

An invisible copy of x (it if doesn’t throw an error).

Examples

check_mcmcr(mcmcr::mcmcr_example$beta)

Description

Check mcmcr

Usage

check_mcmcr(x, sorted = FALSE, x_name = substitute(x), error = TRUE)

Arguments

x        The object to check.
sorted   A flag specifying whether the parameters must be sorted.
x_name   A string of the name of the object.
error    A flag indicating whether to throw an informative error or immediately generate an informative message if the check fails.

Value

An invisible copy of x (it if doesn’t throw an error).

Examples

check_mcmcr(mcmcr::mcmcr_example)
Term Coefficients

Description

Gets coefficients for all the terms in an MCMC object.

Usage

## S3 method for class 'mcarray'
coef(object, conf_level = 0.95,
     estimate = stats::median, ...)

## S3 method for class 'mcmc'
coef(object, conf_level = 0.95,
     estimate = stats::median, ...)

## S3 method for class 'mcmc.list'
coef(object, conf_level = 0.95,
     estimate = stats::median, ...)

## S3 method for class 'mcmcrarray'
coef(object, conf_level = 0.95,
     estimate = stats::median, ...)

Arguments

object The MCMC object to get the coefficients for
conf_level A number specifying the confidence level. By default 0.95.
estimate The function to use to calculate the estimate.
... Unused

Value

An data frame of the coefficients with the columns indicating the term, estimate, standard deviation (sd), zscore, lower and upper credible intervals and pvalue.

Methods (by class)

- mcarray: Get coefficients for terms in mcarray object
- mcmc: Get coefficients for terms in mcmc object
- mcmcr: Get coefficients for terms in mcmcr object
collapse_chains

- mcmcarray: Get coefficients for terms in mcmcarray object
- mcmcr: Get coefficients for terms in mcmcr object

See Also

stats::coef

Examples

coef(mcmcr_example)

collapse_chains

Description

Collapses an MCMC object’s chains into a single chain.

Usage

collapse_chains(x, ...)

## Default S3 method:
collapse_chains(x, ...)

## S3 method for class 'mcmc.list'
collapse_chains(x, ...)

## S3 method for class 'mcmcr'
collapse_chains(x, ...)

Arguments

x An MCMC object.
...

Methods (by class)

- default: Collapse an MCMC object's chains into a single chain
- mcmc.list: Collapse an mcmc.list object's chains into a single chain
- mcmcr: Collapse an mcmcr object's chains into a single chain

Examples

collapse_chains(mcmcr_example)
combine_dimensions

Combines MCMC object samples by dimensions along along using fun.

Usage

```r
combine_dimensions(x, fun = mean, along = NULL, ...) 
```

## S3 method for class 'mcmcarray'
combine_dimensions(x, fun = mean, along = NULL, ...)

## S3 method for class 'mcmcr'
combine_dimensions(x, fun = mean, along = NULL, ...)

Arguments

- `x` An MCMC object
- `fun` The function to use when combining dimensions
- `along` A positive integer (or NULL) indicating the parameter dimension(s) to bind along.
- `...` Unused

Value

The MCMC object with reduced dimensions.

Methods (by class)

- `mcmcarray`: Combine an mcmcarray object’s samples by dimensions
- `mcmcr`: Combine an mcmcr object’s samples by dimensions

Examples

```r
combine_dimensions(mcmcr_example$alpha)
```
combine_samples

Combine MCMC Samples of Two Objects

Description

Combines samples of two MCMC objects (with the same parameters, chains and iterations) using a function.

Usage

```r
combine_samples(x, x2, fun = mean, ...)
```

```r
## S3 method for class 'mcmcaray'
combine_samples(x, x2, fun = mean, ...)
```

```r
## S3 method for class 'mcmcr'
combine_samples(x, x2, fun = mean, ...)
```

Arguments

- `x`: An MCMC object.
- `x2`: A second MCMC object.
- `fun`: The function to use to combine the samples. The function must return a scalar.
- `...`: Unused.

Value

The combined samples as an MCMC object with the same parameters, chains and iterations as the original objects.

Methods (by class)

- `mcmcaray`: Combine samples of two `mcmcaray` objects
- `mcmcr`: Combine samples of two `mcmcr` objects

Examples

```r
combine_samples(mcmcr_example, mcmcr_example, fun = sum)
```
**combine_samples_n**  
*Combine MCMC Samples of multiple objects*

**Description**  
Combines samples of multiple MCMC objects (with the same parameters, chains and iterations) using a function.

**Usage**  
```r  
combine_samples_n(x, ..., fun = mean)  
```

## default S3 method:  
```r  
combine_samples_n(x, ..., fun = mean)  
```

## S3 method for class 'list'  
```r  
combine_samples_n(x, ..., fun = mean)  
```

**Arguments**  
- `x` An MCMC object (or a list of `mcmc` objects).
- `...` Additional MCMC objects.
- `fun` A function.

**Methods (by class)**  
- `default`: Combine samples of multiple MCMC objects  
- `list`: Combine samples of a list of multiple MCMC objects

**Examples**  
```r  
combine_samples_n(mcmcr_example, mcmcr_example, mcmcr_example, fun = sum)  
```

---

**converged**  
*Object Converged*

**Description**  
Test whether an object has converged.
Usage

converged(x, ...)  

## Default S3 method:
converged(x, rhat = 1.1, esr = 0.33, by = "all",
         as_df = FALSE, ...)  

## S3 method for class 'mcmcrs'
converged(x, rhat = 1.1, esr = 0.33, by = "all",
         as_df = FALSE, bound = FALSE, ...)

Arguments

x  An object.

... Unused.
rhat The maximum rhat value.
esr The minimum effective sampling rate.
by A string indicating whether to determine by "term", "parameter" or "all".
as_df A flag indicating whether to return the values as a data frame versus a named list.
bound flag specifying whether to bind mcmcrs objects by their chains before calculating rhat.

Methods (by class)

- default: Test whether an object has converged
- mcmcrs: Test whether an mcmcrs object has converged

See Also

rhat and esr

Examples

converged(mcmcr_example)
converged(mcmcrs(mcmcr_example, mcmcr_example))
converged(mcmcrs(mcmcr_example, mcmcr_example), bound = TRUE)
## dims

### Dimensions

#### Usage

dims(x)

#### Arguments

- **x**: A vector, matrix or array.

#### Value

An integer vector of the dimensions.

#### Examples

dims(character())
dims(2:3)
dims(matrix(1:6, nrow = 2))

---

## esr

### Effective Sampling Rate

#### Description

Calculates the effective sampling rate (esr) based on the formula

\[
\frac{1}{1 + 2 \sum_{k=1}^{\infty} \rho_k(\theta)}
\]

in Brooks et al. (2011). The infinite sum is truncated at lag \( k \) when \( \rho_{k+1}(\theta) < 0 \).

#### Usage

esr(x, ...)

## S3 method for class 'marray'
esr(x, by = "all", ...)

## S3 method for class 'mcmc'
esr(x, by = "all", ...)
## S3 method for class 'mcmc.list'
esr(x, by = "all", ...)

## S3 method for class 'mcmcarray'
esr(x, by = "all", as_df = FALSE, ...)

## S3 method for class 'mcmcr'
esr(x, by = "all", as_df = FALSE, ...)

## S3 method for class 'mcmcrs'
esr(x, by = "all", as_df = FALSE, ...)

### Arguments

- **x**: An MCMC object.
- **...**: Unused.
- **by**: A string indicating whether to determine by "term", "parameter" or "all".
- **as_df**: A flag indicating whether to return the values as a data frame versus a named list.

### Value

The esr value(s) as a data frame or list

### Methods (by class)

- **mcarray**: Effective Sampling Rate for an mcarray object
- **mcmc**: Effective Sampling Rate for an mcmc object
- **mcmc.list**: Effective Sampling Rate for an mcmc.list object
- **mcmcarray**: Effective Sampling Rate for an mcmcarray object
- **mcmcr**: Effective Sampling Rate for an mcmcr object
- **mcmcrs**: Effective Sampling Rate for an mcmcrs object

### References


### Examples

```r
esr(mcmcr_example)
esr(mcmcrs(mcmcr_example, mcmcr_example))
```
ess | Effective Sample Size
---

**Description**

Calculates the effective sample size based on `esr`.

**Usage**

```r
ess(x, by = "all", as_df = FALSE)
```

**Arguments**

- `x`: An MCMC object.
- `by`: A string indicating whether to determine by "term", "parameter" or "all".
- `as_df`: A flag indicating whether to return the results as a data frame or list.

**Examples**

```r
ess(mcmcr_example)
```

---

estimates | Estimates
---

**Description**

Get the estimates for an MCMC object.

**Usage**

```r
estimates(object, ...) 
```

```r
# S3 method for class 'mcar'
estimates(object, fun = stats::median, as_df = FALSE, ...) 
```

```r
# S3 method for class 'mcmc'
estimates(object, fun = stats::median, as_df = FALSE, ...) 
```

```r
# S3 method for class 'mcmc.list'
estimates(object, fun = stats::median, as_df = FALSE, ...) 
```

```r
# S3 method for class 'mmcarray'
estimates(object, fun = stats::median, as_df = FALSE, ...) 
```
is.mcarray

estimates(object, fun = stats::median, as_df = FALSE, ...)

## S3 method for class 'mcmcr'
estimates(object, fun = stats::median, as_df = FALSE, ...)

Arguments

object
An MCMC object.

... Unused.

fun
The function to use.

as_df
A flag indicating whether to return the estimates as a data frame versus a named list.

Value

A named list or data frame.

Methods (by class)

- marray: Estimates for an marray object
- mcmc: Estimates for an mcmc object
- mcmc.list: Estimates for an mcmc.list object
- mcmcarray: Estimates for an mcmcarray object
- mcmcr: Estimates for an mcmcr object

Examples

estimates(mcmcr_example)
estimates(mcmcr_example, as_df = TRUE)

is.mcarray

Is mcarray Object

Description

Tests whether an object is an mcarray.

Usage

is.mcarray(x)

Arguments

x
The object to test.
is.mcmcr

Value
A flag indicating whether the test was positive.

Examples
is.mcmcr(mcmcr_example)

is.mcmcr
Is mcmcr Object

Description
Tests whether an object is an mcmcr-object.

Usage
is.mcmcr(x)

Arguments
x The object to test.

Value
A flag indicating whether the test was positive.

Examples
is.mcmcr(mcmcr_example$beta)

is.mcmcr
Is mcmcr Object

Description
Tests whether an object is an mcmcr-object.

Usage
is.mcmcr(x)

Arguments
x The object to test.
is.mcmcrs

Value
A flag indicating whether the test was positive.

Examples
is.mcmcr(mcmcr_example)

Description
Tests whether an object is an mcmcrs-object.

Usage
is.mcmcrs(x)

Arguments
x  The object to test.

Value
A flag indicating whether the test was positive.

Examples
is.mcmcr(mcmcr(mcmcr_example))

is.term

Description
Test whether an object is a term.

Usage
is.term(x)

Arguments
x  The object to test.
**Value**

A flag indicating whether the test was positive.

**Examples**

```r
is.term(terms(mcmcr_example))
```

---

**mcmcarray-object mcmcarray**

**Description**

An `mcmcarray` object is an array where the first dimension is the chains, the second dimension is the iterations and the subsequent dimensions represent the dimensionality of the parameter. The name `mcmcarray` reflects the fact that the MCMC dimensions, ie the chains and iterations, precede the parameter dimensions.

**Examples**

```r
mcmcr_example$beta
```

---

**mcmcr-object mcmcr**

**Description**

An `mcmcr` object stores multiple uniquely named `mcmcarray-object` objects with the same number of chains and iterations.

**Details**

`mcmcr` objects allow a set of dimensionality preserving parameters to be manipulated and queried as a whole.

**Examples**

```r
mcmcr_example
```
**mcmcrs**

Create mcmcrs

---

**Description**

Creates an `mcmcrs-object` from multiple `mcmcr-object`s.

**Usage**

```r
mcmcrs(...)```

**Arguments**

- `...`  
  Objects of class mcmcr.

**Value**

An object of class mcmcr

**Examples**

```r
mcmcrs(mcmcr_exampleL mcmcr_example)
```

---

**mcmcrs-object**

**mcmcr**

---

**Description**

An `mcmcrs-object` stores multiple `mcmcr-objects` with the same parameters and the same number of chains and iterations.

**Details**

`mcmcrs` objects allow the results of multiple analyses using the same model to be manipulated and queried as a whole.

**Examples**

```r
mcmcrs(mcmcr_exampleL mcmcr_example)
```
mcmcr_example

An Example mcmcr Object

Description

An example mcmcr-object derived from on coda::line.

Usage

mcmcr_example

Format

An object of class mcmcr of length 3.

Examples

mcmcr_example

mcmc_aperm

MCMC Object Transposition

Description

Transpose an MCMC object by permuting its parameter dimensions.

Usage

mcmc_aperm(x, perm, ...)

## S3 method for class 'mcmcarray'
mcmc_aperm(x, perm = NULL, ...)

## S3 method for class 'mcmc'
mcmc_aperm(x, perm = NULL, ...)

## S3 method for class 'mcmc.list'
mcmc_aperm(x, perm = NULL, ...)

## S3 method for class 'mcmcr'
mcmc_aperm(x, perm = NULL, ...)

## S3 method for class 'mcmcrs'
mcmc_aperm(x, perm = NULL, ...)
mcmc_map

Arguments

- `x` : The MCMC object to transpose.
- `perm` : A integer vector of the new order for the parameter dimensions. Missing parameter dimensions are added on the end. If `perm = NULL` (the default) the parameter dimensions are reversed.
- ... : Unused

Value

The modified MCMC object

Methods (by class)

- `mcmcarray`: Transpose an mcmcarray object
- `mcmc`: Transpose an mcmc object
- `mcmc.list`: Transpose an mcmc.list object
- `mcmcr`: Transpose an mcmcr object
- `mcmcrs`: Transpose an mcmcrs object

Description

Adjust the sample values of an MCMC object using a function.

Usage

```r
mcmc_map(.x, .f, .by = 1:npdims(.x), ...)
```

## S3 method for class 'mcmcarray'
```r
mcmc_map(.x, .f, .by = 1:npdims(.x), ...)
```

## S3 method for class 'mcmc'
```r
mcmc_map(.x, .f, .by = TRUE, ...)
```

## S3 method for class 'mcmc.list'
```r
mcmc_map(.x, .f, .by = TRUE, ...)
```

## S3 method for class 'mcmcr'
```r
mcmc_map(.x, .f, .by = TRUE, ...)
```

## S3 method for class 'mcmcrs'
```r
mcmc_map(.x, .f, .by = TRUE, ...)
```
Arguments

.x An MCMC object
.f The function to use
.by A positive integer vector of the dimensions to apply the function over.
... Additional arguments passed to .f.

Value

The updated MCMC object.

Methods (by class)

- mcmcarray: Adjust the sample values of an MCMC object
- mcmc: Adjust the sample values of an mcmc.list object
- mcmc.list: Adjust the sample values of an mcmc.list object
- mcmcr: Adjust the sample values of an mcmcr object
- mcmcrs: Adjust the sample values of an mcmcrs object

Examples

mcmc_map(mcmcr_example)$beta, exp)

nchains Number of MCMC chains

Description

Gets the number of MCMC chains.

Usage

nchains(x, ...)

## S3 method for class 'marray'
nchains(x, ...)

## S3 method for class 'mcmc'
nchains(x, ...)

## S3 method for class 'mcmc.list'
nchains(x, ...)

## S3 method for class 'mcmcarray'
nchains(x, ...)
## nchains

```r
## S3 method for class 'mcmcr'
nchains(x, ...)
## S3 method for class 'mcmcrs'
nchains(x, ...)
## S3 method for class 'mcarray'
niters(x, ...)
## S3 method for class 'mcmc'
niters(x, ...)
## S3 method for class 'mcmc.list'
niters(x, ...)
## S3 method for class 'mcmccarray'
niters(x, ...)
## S3 method for class 'mcmcr'
niters(x, ...)
## S3 method for class 'mcmcrs'
niters(x, ...)
```

### Arguments

- **x**
  - An MCMC object
- **...**
  - Unused

### Value

A count indicating the number of MCMC chains

### Methods (by class)

- `mcarray`: Number of MCMC chains for an mcarray object
- `mcmc`: Number of MCMC chains for an mcmc object
- `mcmc.list`: Number of MCMC chains for an mcmc.list object
- `mcarray`: Number of MCMC chains for an mcarray object
- `mcmcr`: Number of MCMC chains for an mcmcr object
- `mcmcrs`: Number of MCMC chains for an mcmcrs object
- `mcarray`: Number of MCMC iterations for an mcarray object
- `mcmc`: Number of MCMC iterations for an mcmc object
- `mcmc.list`: Number of MCMC iterations for an mcmc.list object
- `mcarray`: Number of MCMC iterations for an mcarray object
- `mcmcr`: Number of MCMC iterations for an mcmcr object
- `mcmcrs`: Number of MCMC iterations for an mcmcrs object
Examples
nchains(mcmcr_example)

ndims

Number of dimensions

Description
Number of dimensions

Usage
ndims(x)

Arguments
x A vector, matrix or array.

Value
A count of the number of dimensions

nitors

Number of MCMC samples

Description
Gets the number of MCMC iterations (in a chain).

Usage
nitors(x, ...)

Arguments
x The object
... Unused.

Value
A count indicating the number of MCMC iterations.

Examples
nitors(mcmcr_example)
npars  Number of Parameters

Description

Gets the number of parameters for an object.

Usage

npars(x, ...)

## Default S3 method:
npars(x, ...)

## S3 method for class 'mcarray'
npars(x, ...)

## S3 method for class 'mcmcarray'
npars(x, ...)

## S3 method for class 'mcmcr'
npars(x, ...)

## S3 method for class 'mcmcrs'
npars(x, ...)

Arguments

x  The object.
...
  Not used.

Value

A count of the number of parameters.

Methods (by class)

- default: Number of parameters for an object
- mcarray: Number of parameters for an mcarray object
- mcmcarray: Number of parameters for an mcmcarray object
- mcmcr: Number of parameters for an mcmcr object
- mcmcrs: Number of parameters for an mcmcrs object

Examples

npars(mcmcr_example)
npdims

*Number of Parameter Dimensions*

**Description**

Gets the number parameter dimensions of an object.

**Usage**

```
npdims(x, ...)
```

**Arguments**

- `x` The object
- `...` Unused.

**Examples**

```
pdims(mcmcr_example)
```

---

npdims.default

*Parameter Dimensions*

**Description**

Gets the parameter dimensions of an object.

**Usage**

```r
## Default S3 method:
npdims(x, ...)

## S3 method for class 'mcmc.list'
npdims(x, ...)

## S3 method for class 'mcmcr'
npdims(x, ...)

pdims(x, ...)

## S3 method for class 'term'
pdims(x, ...)

## S3 method for class 'mcarray'
pdims(x, ...)
```
## nsams

### Description

The product of the number of simulations and the number of terms.

### Usage

```
nsams(x)
```
Arguments

x  The MCMC object.

Value

A count of the total number of samples.

Examples

nsams(mcmcr_example)

---

nsims  Number of MCMC Simulations

Description

The product of the number of chains and number of iterations (in each chain).

Usage

nsims(x)

Arguments

x  The MCMC object.

Value

A count of the total number of simulations.

Examples

nsims(mcmcr_example)
### Description

Gets the number of terms for an object.

### Usage

```r
nenterms(x, ...)
```

#### Default S3 method:

```r
nenterms(x, ...)
```

#### S3 method for class 'mcmcarray'

```r
nenterms(x, ...)
```

#### S3 method for class 'mcmcr'

```r
nenterms(x, ...)
```

#### S3 method for class 'mcmcrs'

```r
nenterms(x, ...)
```

### Arguments

- **x**: The object.
- **...**: Not used.

### Value

A count of the number of terms.

### Methods (by class)

- **default**: Number of terms for an object
- **mcmcarray**: Number of terms for an mcmcarray object
- **mcmcr**: Number of terms for an mcmcr object
- **mcmcrs**: Number of terms for an mcmcrs object

### Examples

```r
ntenms(mcmcr_example)
```
Parameter Names

Description

Gets or sets the parameter names for an object.

Usage

```r
parameters(x, ...)  
parameters(x) <- value  
set_parameters(x, parameters)
```

```r  
## S3 method for class 'term'  
parameters(x, scalar_only = FALSE, terms = FALSE, ...)
```

```r  
## S3 method for class 'mcmc'  
parameters(x, scalar_only = FALSE, terms = FALSE, ...)
```

```r  
## S3 method for class 'mcmc.list'  
parameters(x, scalar_only = FALSE, terms = FALSE, ...)
```

```r  
## S3 method for class 'mcmcr'  
parameters(x, scalar_only = FALSE, terms = FALSE, ...)
```

```r  
## S3 method for class 'mcmcrs'  
parameters(x, scalar_only = FALSE, terms = FALSE, ...)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>x</code></td>
<td>An MCMC object.</td>
</tr>
<tr>
<td><code>...</code></td>
<td>Not used.</td>
</tr>
<tr>
<td><code>value</code></td>
<td>A character vector of the new parameter names.</td>
</tr>
<tr>
<td><code>parameters</code></td>
<td>A character vector of the new parameter names.</td>
</tr>
<tr>
<td><code>scalar_only</code></td>
<td>A flag indicating whether to only get the names of parameters with one term.</td>
</tr>
<tr>
<td><code>terms</code></td>
<td>A flag indicating whether to return the parameter name for each term.</td>
</tr>
</tbody>
</table>

Value

A character vector of the parameter names.
Methods (by class)

- term: Parameter names for a term vector
- mcmc: Parameter names for an mcmc object
- mcmc.list: Parameter names for an mcmc.list object
- mcmcr: Parameter names for an mcmcr object
- mcmcrs: Parameter names for an mcmcrs object

Examples

```r
parameters(mcmcr_example)
parameters(mcmcr_example) <- c("gamma", "theta", "tau")
parameters(mcmcr_example)
parameters(mcmcr_example, scalar_only = TRUE)
parameters(mcmcr_example, terms = TRUE)
```

---

### pvalue

**P-Value**

<table>
<thead>
<tr>
<th>Description</th>
<th>Usage</th>
<th>Arguments</th>
<th>Value</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculates the p-value.</td>
<td>pvalue(x)</td>
<td>x</td>
<td>A vector of MCMC value</td>
<td>pvalue(as.numeric(0:10))</td>
</tr>
</tbody>
</table>
rhat

*Description*

By default calculates the uncorrected, unfolded, univariate, split R-hat (potential scale reduction factor) values.

*Usage*

```r
rhat(x, ...)  
## S3 method for class 'mcarray'
  rhat(x, by = "all", as_df = FALSE, ...)
  
## S3 method for class 'mcmc'
  rhat(x, by = "all", as_df = FALSE, ...)
  
## S3 method for class 'mcmc.list'
  rhat(x, by = "all", as_df = FALSE, ...)
  
## S3 method for class 'mcmcrarray'
  rhat(x, by = "all", as_df = FALSE, ...)
  
## S3 method for class 'mcmcr'
  rhat(x, by = "all", as_df = FALSE, ...)
  
## S3 method for class 'mcmcrs'
  rhat(x, by = "all", as_df = FALSE, bound = FALSE, ...)
```

*Arguments*

- `x`: An MCMC object.
- `...`: Unused.
- `by`: A string indicating whether to determine by "term", "parameter" or "all".
- `as_df`: A flag indicating whether to return the values as a data frame versus a named list.
- `bound`: flag specifying whether to bind mcmcrs objects by their chains before calculating rhat.

*Value*

The rhat value(s).
Methods (by class)

- `mcarray`: R-hat for an mcarray object
- `mcmc`: R-hat for an mcmc object
- `mcmc.list`: R-hat for an mcmc.list object
- `mcmcarray`: R-hat for an mcmcarray object
- `mcmcr`: R-hat for an mcmcr object
- `mcmcrs`: R-hat for an mcmcrs object

References


Examples

```r
rhat(mcmcr_example)
rhat(mcmcr_example, by = "parameter")
rhat(mcmcr_example, by = "term")
rhat(mcmcr_example, by = "term", as_df = TRUE)
rhat(mcmcrs(mcmcr_example, mcmcr_example))
rhat(mcmcrs(mcmcr_example, mcmcr_example), bound = TRUE)
```

sort  

Sort an MCMC Object

Description

Sorts an MCMC object by its parameter names.

Usage

```r
## S3 method for class 'mcmc'
sort(x, ...)

## S3 method for class 'mcmc.list'
sort(x, ...)

## S3 method for class 'mcmcr'
sort(x, ...)

## S3 method for class 'mcmcrs'
sort(x, ...)
```

Arguments

- `x` The MCMC object to sort
- `...` Unused
Methods (by class)

- `mcmc`: Sort an mcmc object
- `mcmc.list`: Sort an mcmc.list object
- `mcmcr`: Sort an mcmcr object
- `mcmcrs`: Sort an mcmcrs object

Examples

```r
parameters(mcmcr_example)
mcmcr_example <- subset(mcmcr_example, parameters = c("beta", "alpha"))
parameters(mcmcr_example)
mcmcr_example <- sort(mcmcr_example)
parameters(mcmcr_example)
```

---

split_chains  

**Split Chains**

**Description**

Splits each chain in half to double the number chains and halve the number of iterations.

**Usage**

```r
split_chains(x, ...)
```

## S3 method for class 'mcmcarray'
```r
split_chains(x, ...)
```

## S3 method for class 'mcmcr'
```r
split_chains(x, ...)
```

**Arguments**

- `x`  
  An MCMC object.
- `...`  
  Unused.

**Methods (by class)**

- `mcmcarray`: Split chains for an mcmcarray object
- `mcmcr`: Split chains for an mcmcr object

**Examples**

```r
split_chains(mcmcr_example)
```
Description

Subsets an MCMC object by its chains, iterations and/or parameters.

Usage

```r
## S3 method for class 'term'
subset(x, parameters = NULL, ...)

## S3 method for class 'mcmc'
subset(x, iterations = NULL, parameters = NULL, ...)

## S3 method for class 'mcmc.list'
subset(x, chains = NULL, iterations = NULL,
       parameters = NULL, ...)

## S3 method for class 'mcmcarray'
subset(x, chains = NULL, iterations = NULL,
       parameters = NULL, ...)

## S3 method for class 'mcmcr'
subset(x, chains = NULL, iterations = NULL,
       parameters = NULL, ...)

## S3 method for class 'mcmcrs'
subset(x, chains = NULL, iterations = NULL,
       parameters = NULL, ...)
```

Arguments

- `x` The MCMC object to subset
- `parameters` A character vector (or NULL) of the parameters to subset by.
- `iterations` An integer vector (or NULL) of the iterations to subset by.
- `chains` An integer vector (or NULL) of the chains to subset by.

Methods (by class)

- `term`: Subset a term vector
- `mcmc`: Subset an mcmc object
- `mcmc.list`: Subset an mcmc.list object
- `mcmcarray`: Subset an mcmcarray object
- `mcmcr`: Subset an mcmcr object
- `mcmcrs`: Subset an mcmcrs object
### Examples

```r
subset(mcmcr_example, chains = 2L, iterations = 1:100, parameters = c("beta", "alpha"))
```

<table>
<thead>
<tr>
<th>tdims</th>
<th>Term Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Description

Gets the term dimensions of an object.

### Usage

```r
tdims(x, ...)
```

#### ## Default S3 method:
```r
tdims(x, ...)
```

#### ## S3 method for class 'term'
```r
tdims(x, ...)
```

### Arguments

- `x` The object
- `...` Unused.

### Methods (by class)

- default: Term dimensions for an object
- term: Term dimensions for a term vector

### Examples

```r
tdims(mcmcr_example)
```

<table>
<thead>
<tr>
<th>term</th>
<th>Term Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Description

A term vector is a sortable vector of the terms from an analysis.

### Examples

```r
terms <- terms(mcmcr_example)
terms <- rev(terms)
terms
sort(terms)
```
**terms**

*MCMC Object Terms*

**Description**

Gets terms for an MCMC object.

**Usage**

```r
## S3 method for class 'mcmc'
terms(x, ...)
## S3 method for class 'mcmc.list'
terms(x, ...)
## S3 method for class 'mcmcarray'
terms(x, ...)
## S3 method for class 'mcmcr'
terms(x, ...)
```

**Arguments**

- `x` A MCMC object.
- `...` Unused

**Value**

A term vector.

**Methods (by class)**

- `mcmc`: Get terms for an mcmc object
- `mcmc.list`: Get terms for an mcmc.list object
- `mcmcarray`: Get terms for an mcmcarray object
- `mcmcr`: Get terms for an mcmcr object

**See Also**

`stats::terms`

**Examples**

```r
terms(mcmcr_example)
```
Thin MCMC Samples

Description

Thin an MCMC objects samples.

Usage

```r
## S3 method for class 'mcmcarray'
thin(x, nthin = 1L, ...)
```

```r
## S3 method for class 'mcmcr'
thin(x, nthin = 1L, ...)
```

```r
## S3 method for class 'mcmcrs'
thin(x, nthin = 1L, ...)
```

Arguments

- `x` An MCMC object
- `nthin` A count of the thinning rate.
- `...` Unused

Value

The thinned object.

Methods (by class)

- `mcmcarray`: Thin MCMC samples for an mcmcarray object
- `mcmcr`: Thin MCMC samples for an mcmcr object
- `mcmcrs`: Thin MCMC samples for an mcmcrs object

See Also

`coda::thin`

Examples

```r
thin(mcmcr_example, nthin = 10L)
```
Zero MCMC Sample Values

Description

Zeros an MCMC object’s sample values.

Usage

```
zero(x, ...) # S3 method for class 'mcarray'
zero(x, ...) # S3 method for class 'mcmcarray'
zero(x, parameters = NULL, ...) # S3 method for class 'mcmcr'
```

Arguments

- `x`: The MCMC object.
- `...`: Unused
- `parameters`: A character vector (or NULL) of the parameters to zero.

Details

It is used for removing the effect of a random effect where the expected value is 0.

Methods (by class)

- `mcarray`: Zero an mcarray object
- `mcmcarray`: Zero an mcmcarray object
- `mcmcr`: Zero an mcmcr object

Examples

```
zero(mcmcr_example, parameters = "beta")
```
Index

*Topic datasets
  mcmcr_example, 30

anyNA, 3, 3
as.character.term(as.term), 9
as.mcmarray, 4
as.mcmc, 4
as.mcmc.list, 5
as.mcmcaray, 6
as.mcmc, 7
as.mcmcrs, 8
as.mcmcrs.list(as.mcmc), 7
as.term, 9

bind_chains, 10
bind_dimensions, 11, 12
bind_dimensions_n, 11, 12
bind_iterations, 12
bind_parameters, 13

check_mcmcaray, 14
check_mmc, 15
cof, 16, 17
collapse_chains, 17
combine_dimensions, 18
combine_samples, 19
combine_samples_n, 20
converged, 20

dims, 22

esr, 21, 22, 24
ess, 24
estimates, 24

is.mcmarray, 25
is.mcmcaray, 26
is.mcmc, 26
is.mcmcrs, 27
is.term, 27

line, 30

mcmc, 4, 5
mcmc.list, 5
mcmc_aperm, 30
mcmc_map, 31
mcmcaray-object, 28
mcmc-object, 28
mcmcr_example, 30
mcmcrs, 29
mcmcrs-object, 29

nchains, 32
ndims, 34
niter, 34
niter.mcmcaray (nchains), 32
niter.mmc (nchains), 32
niter.mcmcaray (nchains), 32
niter.mmc (nchains), 32
niter.mcmcrs (nchains), 32
npars, 35
npdms, 36
npdms.default, 36
npdms.mmc.list (npdms.default), 36
npdms.mmc (npdms.default), 36
nsams, 37
nsims, 38
nterms, 39

parameters, 40
parameters<-(parameters), 40
pdims (npdms.default), 36
pvalue, 41

rhat, 21, 42

set_parameters (parameters), 40
sort, 43
split_chains, 44
subset, 45
INDEX

tdims, 46
term, 46
terms, 47, 47
thin, 48, 48

zero, 49