

Package ‘mcmcr’

February 11, 2021

Title Manipulate MCMC Samples

Version 0.5.0

Description Functions and classes to store, manipulate and summarise Monte Carlo Markov Chain (MCMC) samples. For more information see Brooks et al. (2011) <isbn:978-1-4200-7941-8>.

License MIT + file LICENSE

URL <https://github.com/poissonconsulting/mcmcr>

BugReports <https://github.com/poissonconsulting/mcmcr/issues>

Depends R (>= 3.5)

Imports abind,
chk,
coda,
extras,
generics,
lifecycle,
nlist (>= 0.3.1),
purrr,
stats,
term,
universals,
utils

Suggests covr,
graphics,
rlang,
testthat,
tibble

RdMacros lifecycle

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

R topics documented:

as.marray	3
as.mcmc.marray	4
as.mcmc.mcmc	5
as.mcmc.mcmcarray	6
as.mcmc.mcmcr	6
as.mcmcarray	7
as.mcmcr	8
as.mcmcrs	9
as_nlist.mcmcr	10
as_nlists.mcmcr	10
bind_chains.marray	11
bind_chains.mcmc	12
bind_chains.mcmc.list	12
bind_chains.mcmcarray	13
bind_chains.mcmcr	13
bind_dimensions	14
bind_dimensions_n	14
bind_parameters	15
check_mcmcarray	15
check_mcmcr	16
chk_mcmcr	16
coef	17
collapse_chains.mcmcr	18
combine_dimensions	19
combine_samples	19
combine_samples_n	20
converged.default	20
converged.mcmcrs	21
esr.marray	22
esr.mcmc	23
esr.mcmc.list	24
esr.mcmcarray	25
esr.mcmcr	26
esr.mcmcrs	27
ess	28
estimates.marray	28
estimates.mcmc	29
estimates.mcmc.list	29
estimates.mcmcarray	30
estimates.mcmcr	30
fill_all.marray	31
fill_all.mcmcarray	32
fill_all.mcmcr	33
is.marray	34
is.mcmcarray	35
is.mcmcr	35
is.mcmcrs	36
mcmcarray-object	36
mcmcr-object	37
mcmcrs	37

mcmcrs-object	38
mcmcr_example	38
mcmc_aperm	39
mcmc_map	39
nchains.marray	40
nchains.mcmarray	40
nchains.mcmcr	41
nchains.mcmcrs	41
niters.marray	42
niters.mcmarray	42
niters.mcmcr	43
niters.mcmcrs	43
npars.marray	44
npars.mcmarray	44
npars.mcmcr	45
npdims.mcmarray	46
npdims.mcmcr	46
nterms.mcmarray	47
nterms.mcmcr	47
nterms.mcmcrs	48
params	48
pars.mcmcr	49
pars.mcmcrs	49
pdims.marray	50
pdims.mcmarray	51
pdims.mcmcr	51
rhat.marray	52
rhat.mcmc	53
rhat.mcmc.list	54
rhat.mcmarray	55
rhat.mcmcr	56
rhat.mcmcrs	57
set_pars.mcmcr	58
set_pars.mcmcrs	58
split_chains.mcmarray	59
split_chains.mcmcr	60
subset	60
vld_mcmcr	61
zero	62

Index **64**

<code>as.marray</code>	<i>Coerce to an marray object</i>
------------------------	-----------------------------------

Description

Coerces MCMC objects to an marray object.

Usage

```
as.marray(x, ...)

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

Arguments

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

Methods (by class)

- `list`: Convert a list of uniquely named objects that can be coerced to [mcmarray-object]s to an mcmcr object

Examples

```
as.marray(mcmcr_example$beta)
```

as.mcmc.marray

Markov Chain Monte Carlo Objects

Description

The function `mcmc` is used to create a Markov Chain Monte Carlo object. The input data are taken to be a vector, or a matrix with one column per variable.

If the optional arguments `start`, `end`, and `thin` are omitted then the chain is assumed to start with iteration 1 and have thinning interval 1. If data represents a chain that starts at a later iteration, the first iteration in the chain should be given as the `start` argument. Likewise, if data represents a chain that has already been thinned, the thinning interval should be given as the `thin` argument.

An `mcmc` object may be summarized by the `summary` function and visualized with the `plot` function.

MCMC objects resemble time series (`ts`) objects and have methods for the generic functions `time`, `start`, `end`, `frequency` and `window`.

Usage

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

Arguments

```
x          An object that may be coerced to an mcmc object
...        Further arguments to be passed to specific methods
```

Author(s)

Martyn Plummer

See Also

[mcmc.list](#), [mcmcUpgrade](#), [thin](#), [window.mcmc](#), [summary.mcmc](#), [plot.mcmc](#).

as.mcmc.mcmc

Markov Chain Monte Carlo Objects

Description

The function `mcmc` is used to create a Markov Chain Monte Carlo object. The input data are taken to be a vector, or a matrix with one column per variable.

If the optional arguments `start`, `end`, and `thin` are omitted then the chain is assumed to start with iteration 1 and have thinning interval 1. If `data` represents a chain that starts at a later iteration, the first iteration in the chain should be given as the `start` argument. Likewise, if `data` represents a chain that has already been thinned, the thinning interval should be given as the `thin` argument.

An `mcmc` object may be summarized by the `summary` function and visualized with the `plot` function.

MCMC objects resemble time series (`ts`) objects and have methods for the generic functions `time`, `start`, `end`, `frequency` and `window`.

Usage

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

Arguments

<code>x</code>	An object that may be coerced to an <code>mcmc</code> object
<code>...</code>	Further arguments to be passed to specific methods

Author(s)

Martyn Plummer

See Also

[mcmc.list](#), [mcmcUpgrade](#), [thin](#), [window.mcmc](#), [summary.mcmc](#), [plot.mcmc](#).

as.mcmc.mcmcarray *Markov Chain Monte Carlo Objects*

Description

The function `mcmc` is used to create a Markov Chain Monte Carlo object. The input data are taken to be a vector, or a matrix with one column per variable.

If the optional arguments `start`, `end`, and `thin` are omitted then the chain is assumed to start with iteration 1 and have thinning interval 1. If data represents a chain that starts at a later iteration, the first iteration in the chain should be given as the `start` argument. Likewise, if data represents a chain that has already been thinned, the thinning interval should be given as the `thin` argument.

An `mcmc` object may be summarized by the `summary` function and visualized with the `plot` function.

MCMC objects resemble time series (`ts`) objects and have methods for the generic functions `time`, `start`, `end`, `frequency` and `window`.

Usage

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

Arguments

`x` An object that may be coerced to an `mcmc` object
`...` Further arguments to be passed to specific methods

Author(s)

Martyn Plummer

See Also

[mcmc.list](#), [mcmcUpgrade](#), [thin](#), [window.mcmc](#), [summary.mcmc](#), [plot.mcmc](#).

as.mcmc.mcmcr *Markov Chain Monte Carlo Objects*

Description

The function `mcmc` is used to create a Markov Chain Monte Carlo object. The input data are taken to be a vector, or a matrix with one column per variable.

If the optional arguments `start`, `end`, and `thin` are omitted then the chain is assumed to start with iteration 1 and have thinning interval 1. If data represents a chain that starts at a later iteration, the first iteration in the chain should be given as the `start` argument. Likewise, if data represents a chain that has already been thinned, the thinning interval should be given as the `thin` argument.

An `mcmc` object may be summarized by the `summary` function and visualized with the `plot` function.

MCMC objects resemble time series (`ts`) objects and have methods for the generic functions `time`, `start`, `end`, `frequency` and `window`.

Usage

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

Arguments

x An object that may be coerced to an mcmc object
... Further arguments to be passed to specific methods

Author(s)

Martyn Plummer

See Also

[mcmc.list](#), [mcmcUpgrade](#), [thin](#), [window.mcmc](#), [summary.mcmc](#), [plot.mcmc](#).

as.mcmcarray	<i>Coerce to an mcmcarray object</i>
--------------	--------------------------------------

Description

Coerces MCMC objects to an [mcmcarray-object\(\)](#).

Usage

```
as.mcmcarray(x, ...)
```

Arguments

x object to coerce.
... Unused.

Examples

```
as.mcmcarray(as.marray(mcmcr_example$beta))
```

`as.mcmc`*Convert to an mcmc Object*

Description

Converts an MCMC object to an `mcmc-object()`.

Usage

```
as.mcmc(x, ...)  
  
## S3 method for class 'marray'  
as.mcmc(x, name = "par", ...)  
  
## S3 method for class 'mcmarray'  
as.mcmc(x, name = "par", ...)  
  
## S3 method for class 'nlist'  
as.mcmc(x, ...)  
  
## S3 method for class 'nlists'  
as.mcmc(x, ...)  
  
## S3 method for class 'mcmc'  
as.mcmc(x, ...)  
  
## S3 method for class 'mcmc.list'  
as.mcmc(x, ...)  
  
## S3 method for class 'mcmcrs'  
as.mcmc(x, ...)
```

Arguments

<code>x</code>	An MCMC object.
<code>...</code>	Unused.
<code>name</code>	A string specifying the parameter name.

Value

An `mcmc` object.

Methods (by class)

- `marray`: Convert an `marray` object to an `mcmc` object
- `mcmarray`: Convert an `mcmarray-object()` to an `mcmc` object
- `nlist`: Convert an `nlist::nlist-object()` to an `mcmc` object
- `nlists`: Convert an `nlist::nlists-object()` to an `mcmc` object
- `mcmc`: Convert an `coda::mcmc()` object to an `mcmc` object
- `mcmc.list`: Convert an `coda::mcmc.list()` object to an `mcmc` object
- `mcmcrs`: Convert an `mcmcrs-object()` to an `mcmc` object

Examples

```
mcmc.list <- coda::as.mcmc.list(mcmc::mcmc_example)
as.mcmcrc(mcmc.list)
```

as.mcmcrcs	<i>Convert to an mcmcrcs object</i>
------------	-------------------------------------

Description

Converts an MCMC object to an `mcmcrcs-object()`.

Usage

```
as.mcmcrcs(x, ...)
```

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

```
## S3 method for class 'mcmcrc'
as.mcmcrcs(x, name = "mcmcrc1", ...)
```

Arguments

x	An MCMC object.
...	Unused.
name	A string specifying the element name.

Value

An `mcmcrcs` object.

Methods (by class)

- `list`: Convert a list of [`mcmcrc-object`]s to an `mcmcrcs` object
- `mcmcrc`: Convert an `mcmcrc-object()` to an `mcmcrcs` object

Examples

```
as.mcmcrcs(mcmc::mcmcrc_example)
```

as_nlist.mcmc *Coerce to nlist*

Description

Coerce an R object to an `nlist_object()`.

Usage

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

Arguments

<code>x</code>	An object.
<code>...</code>	Unused.

Value

An nlist object.

Methods (by class)

- `numeric`: Coerce named numeric vector to nlist
- `list`: Coerce list to nlist
- `data.frame`: Coerce data.frame to nlist
- `mcmc`: Coerce mcmc (with one iteration) to nlist
- `mcmc.list`: Coerce mcmc.list (with one iteration) to nlist

Examples

```
as_nlist(list(x = 1:4))
as_nlist(c(`a[2]` = 3, `a[1]` = 2))
```

as_nlists.mcmc *Coerce to nlists*

Description

Coerce an R object to an `nlists_object()`.

Usage

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

Arguments

<code>x</code>	An object.
<code>...</code>	Unused.

Value

An nlists object.

Methods (by class)

- `list`: Coerce list to nlists
- `mcmc`: Coerce mcmc to nlists
- `mcmc.list`: Coerce mcmc.list to nlists
- `nlist`: Coerce nlist to nlists

Examples

```
as_nlists(list(nlist(x = c(1, 5)), nlist(x = c(2, 3)), nlist(x = c(3, 2))))
```

`bind_chains.marray` *Bind by Chains.*

Description

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

Usage

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

Arguments

<code>x</code>	An object.
<code>x2</code>	A second object.
<code>...</code>	Other arguments passed to methods.

Value

The combined object.

See Also

Other MCMC manipulations: [bind_iterations\(\)](#), [collapse_chains\(\)](#), [estimates\(\)](#), [split_chains\(\)](#)

bind_chains.mcmc *Bind by Chains.*

Description

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

Usage

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

Arguments

x	An object.
x2	A second object.
...	Other arguments passed to methods.

Value

The combined object.

See Also

Other MCMC manipulations: [bind_iterations\(\)](#), [collapse_chains\(\)](#), [estimates\(\)](#), [split_chains\(\)](#)

bind_chains.mcmc.list *Bind by Chains.*

Description

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

Usage

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

Arguments

x	An object.
x2	A second object.
...	Other arguments passed to methods.

Value

The combined object.

See Also

Other MCMC manipulations: [bind_iterations\(\)](#), [collapse_chains\(\)](#), [estimates\(\)](#), [split_chains\(\)](#)

bind_chains.mcmcarray *Bind by Chains.*

Description

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

Usage

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

Arguments

x	An object.
x2	A second object.
...	Other arguments passed to methods.

Value

The combined object.

See Also

Other MCMC manipulations: [bind_iterations\(\)](#), [collapse_chains\(\)](#), [estimates\(\)](#), [split_chains\(\)](#)

bind_chains.mcmcr *Bind by Chains.*

Description

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

Usage

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

Arguments

x	An object.
x2	A second object.
...	Other arguments passed to methods.

Value

The combined object.

See Also

Other MCMC manipulations: [bind_iterations\(\)](#), [collapse_chains\(\)](#), [estimates\(\)](#), [split_chains\(\)](#)

bind_dimensions	<i>Combine two MCMC objects by dimensions</i>
-----------------	---

Description

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

Usage

```
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.

See Also

[bind_dimensions_n\(\)](#)

Examples

```
bind_dimensions(mcmcr_example, mcmcr_example)
```

bind_dimensions_n	<i>Combine multiple MCMC objects by parameter dimensions</i>
-------------------	--

Description

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

Usage

```
bind_dimensions_n(...)
```

Arguments

...	one or more MCMC objects
-----	--------------------------

See Also

[bind_dimensions\(\)](#)

Examples

```
bind_dimensions_n(mcmcr_example, mcmcr_example, mcmcr_example)
```

bind_parameters	<i>Combine two MCMC object by parameters</i>
-----------------	--

Description

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

Usage

```
bind_parameters(x, x2, ...)
```

Arguments

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

Examples

```
bind_parameters(
  subset(mcmcr_example, pars = "sigma"),
  subset(mcmcr_example, pars = "beta")
)
```

check_mcmcarray	Soft-deprecated <i>Check mcmcarray</i>
-----------------	---

Description

Soft-deprecated Check mcmcarray

Usage

```
check_mcmcarray(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.

Value

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

Examples

```
check_mcmcarray(mcmcr::mcmcr_example$beta)
```

check_mcmcr	Soft-deprecated <i>Check mcmcr</i>
-------------	---

Description**Soft-deprecated** 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)
```

chk_mcmcr	<i>Check MCMC Objects</i>
-----------	---------------------------

Description

Checks class and structure of MCMC objects.

chk_mcmarray checks if [mcmarray-object\(\)](#) object using`is.array(x) && is.numeric(x)`chk_mcmcr checks if an [mcmcr-object\(\)](#).chk_mcmcrs checks if an [mcmcrs-object\(\)](#).**Usage**

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

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

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


Arguments

- x The object to check.
- x_name A string of the name of object x or NULL.

Details

To just check class use `chk::chk_s3_class()`.

Value

NULL, invisibly. Called for the side effect of throwing an error if the condition is not met.

Functions

- `chk_mcmcarray`: Check mcmcarray Object
- `chk_mcmcr`: Check mcmcr Object
- `chk_mmcrcs`: Check mcrcs Object

See Also

[vld_mcmcr\(\)](#)

Examples

```
# chk_mcmcarray
try(chk_mcmcarray(1))

# chk_mcmcr
chk_mcmcr(as.mcmcr(list(x = 1)))
try(chk_mcmcr(1))

# chk_mmcrcs
chk_mmcrcs(as.mmcrcs(as.mcmcr(list(x = 1))))
try(chk_mmcrcs(1))
```

coef

Term Coefficients

Description

Gets coefficients for all the terms in an MCMC object.

Usage

```
## S3 method for class 'mcmc'
coef(object, conf_level = 0.95, estimate = median, simplify = FALSE, ...)
```

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.
simplify	A flag specifying whether to return just the estimate, lower, upper and svalue.
...	Unused

Value

An data frame of the coefficients with the columns indicating the term, estimate, the standard deviation (sd) and zscore (if `simplify = FALSE`), lower and upper credible intervals and pvalue or svalue (if `simplify = TRUE`).

Methods (by class)

- `mcmc`: Get coefficients for terms in `mcmc` object

See Also

`stats::[coef][stats::coef]`

Examples

```
coef(mcmc_example)
```

collapse_chains.mcmc *Collapse Chains*

Description

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

Usage

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

Arguments

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

Value

The modified object with one chain.

See Also

Other MCMC manipulations: [bind_chains\(\)](#), [bind_iterations\(\)](#), [estimates\(\)](#), [split_chains\(\)](#)

combine_dimensions	<i>Combine Samples by Dimensions</i>
--------------------	--------------------------------------

Description

Combines MCMC object samples by dimensions along along using fun.

Usage

```
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.

Examples

```
combine_dimensions(mcmcr_example$alpha)
```

combine_samples	<i>Combine MCMC Samples of Two Objects</i>
-----------------	--

Description

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

Usage

```
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.

Examples

```
combine_samples(mcmcr_example, mcmcr_example, fun = sum)
```

combine_samples_n	<i>Combine MCMC Samples of multiple objects</i>
-------------------	---

Description

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

Usage

```
combine_samples_n(x, ..., fun = mean)
```

Arguments

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

Examples

```
combine_samples_n(mcmcr_example, mcmcr_example, mcmcr_example, fun = sum)
```

converged.default	<i>Converged</i>
-------------------	------------------

Description

Tests whether an object has converged.

Usage

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

Arguments

x	An object.
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.
na_rm	A flag specifying whether to ignore missing values.
...	Other arguments passed to methods.

Value

A logical scalar indicating whether the object has converged.

See Also

Other convergence: [converged_pars\(\)](#), [converged_terms\(\)](#), [esr_pars\(\)](#), [esr_terms\(\)](#), [esr\(\)](#), [rhat_pars\(\)](#), [rhat_terms\(\)](#), [rhat\(\)](#)

Examples

```
converged(mcmcr_example)
```

converged.mcmcrcs	<i>Converged</i>
-------------------	------------------

Description

Tests whether an object has converged.

Usage

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

Arguments

x	An object.
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.
na_rm	A flag specifying whether to ignore missing values.
...	Other arguments passed to methods.

Value

A logical scalar indicating whether the object has converged.

See Also

Other convergence: [converged_pars\(\)](#), [converged_terms\(\)](#), [esr_pars\(\)](#), [esr_terms\(\)](#), [esr\(\)](#), [rhat_pars\(\)](#), [rhat_terms\(\)](#), [rhat\(\)](#)

Examples

```
converged(mcmcrs(mcmcr_example, mcmcr_example))
converged(mcmcrs(mcmcr_example, mcmcr_example), bound = TRUE)
```

 esr.marray

Effective Sampling Rate

Description

Calculates the effective sampling rate (esr).

Usage

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

Arguments

x	An object.
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.
na_rm	A flag specifying whether to ignore missing values.
...	Other arguments passed to methods.

Details

By default

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

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

Value

A number between 0 and 1 indicating the esr value.

References

Brooks, S., Gelman, A., Jones, G.L., and Meng, X.-L. (Editors). 2011. Handbook for Markov Chain Monte Carlo. Taylor & Francis, Boca Raton.

See Also

Other convergence: [converged_pars\(\)](#), [converged_terms\(\)](#), [converged\(\)](#), [esr_pars\(\)](#), [esr_terms\(\)](#), [rhat_pars\(\)](#), [rhat_terms\(\)](#), [rhat\(\)](#)

esr.mcmc

Effective Sampling Rate

Description

Calculates the effective sampling rate (esr).

Usage

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

Arguments

x	An object.
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.
na_rm	A flag specifying whether to ignore missing values.
...	Other arguments passed to methods.

Details

By default

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

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

Value

A number between 0 and 1 indicating the esr value.

References

Brooks, S., Gelman, A., Jones, G.L., and Meng, X.-L. (Editors). 2011. Handbook for Markov Chain Monte Carlo. Taylor & Francis, Boca Raton.

See Also

Other convergence: [converged_pars\(\)](#), [converged_terms\(\)](#), [converged\(\)](#), [esr_pars\(\)](#), [esr_terms\(\)](#), [rhat_pars\(\)](#), [rhat_terms\(\)](#), [rhat\(\)](#)

esr.mcmc.list

Effective Sampling Rate

Description

Calculates the effective sampling rate (esr).

Usage

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

Arguments

x	An object.
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.
na_rm	A flag specifying whether to ignore missing values.
...	Other arguments passed to methods.

Details

By default

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

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

Value

A number between 0 and 1 indicating the esr value.

References

Brooks, S., Gelman, A., Jones, G.L., and Meng, X.-L. (Editors). 2011. Handbook for Markov Chain Monte Carlo. Taylor & Francis, Boca Raton.

See Also

Other convergence: [converged_pars\(\)](#), [converged_terms\(\)](#), [converged\(\)](#), [esr_pars\(\)](#), [esr_terms\(\)](#), [rhat_pars\(\)](#), [rhat_terms\(\)](#), [rhat\(\)](#)

 esr.mcmcarray

Effective Sampling Rate

Description

Calculates the effective sampling rate (esr).

Usage

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

Arguments

x	An object.
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.
na_rm	A flag specifying whether to ignore missing values.
...	Other arguments passed to methods.

Details

By default

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

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

Value

A number between 0 and 1 indicating the esr value.

References

Brooks, S., Gelman, A., Jones, G.L., and Meng, X.-L. (Editors). 2011. Handbook for Markov Chain Monte Carlo. Taylor & Francis, Boca Raton.

See Also

Other convergence: [converged_pars\(\)](#), [converged_terms\(\)](#), [converged\(\)](#), [esr_pars\(\)](#), [esr_terms\(\)](#), [rhat_pars\(\)](#), [rhat_terms\(\)](#), [rhat\(\)](#)

 esr.mcmc

Effective Sampling Rate

Description

Calculates the effective sampling rate (esr).

Usage

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

Arguments

x	An object.
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.
na_rm	A flag specifying whether to ignore missing values.
...	Other arguments passed to methods.

Details

By default

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

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

Value

A number between 0 and 1 indicating the esr value.

References

Brooks, S., Gelman, A., Jones, G.L., and Meng, X.-L. (Editors). 2011. Handbook for Markov Chain Monte Carlo. Taylor & Francis, Boca Raton.

See Also

Other convergence: [converged_pars\(\)](#), [converged_terms\(\)](#), [converged\(\)](#), [esr_pars\(\)](#), [esr_terms\(\)](#), [rhat_pars\(\)](#), [rhat_terms\(\)](#), [rhat\(\)](#)

Examples

```
esr(mcmc_example)
```

esr.mcmcrcs *Effective Sampling Rate*

Description

Calculates the effective sampling rate (esr).

Usage

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

Arguments

x	An object.
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.
na_rm	A flag specifying whether to ignore missing values.
...	Other arguments passed to methods.

Details

By default

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

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

Value

A number between 0 and 1 indicating the esr value.

References

Brooks, S., Gelman, A., Jones, G.L., and Meng, X.-L. (Editors). 2011. Handbook for Markov Chain Monte Carlo. Taylor & Francis, Boca Raton.

See Also

Other convergence: [converged_pars\(\)](#), [converged_terms\(\)](#), [converged\(\)](#), [esr_pars\(\)](#), [esr_terms\(\)](#), [rhat_pars\(\)](#), [rhat_terms\(\)](#), [rhat\(\)](#)

Examples

```
esr(mcmcrcs(mcmcr_example, mcmcr_example))
```

ess	<i>P-Value Effective Sample Size</i>
-----	--------------------------------------

Description

Calculates the effective sample size based on `esr()`.

Usage

```
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

```
ess(mcmcr_example)
```

estimates.marray	<i>Estimates</i>
------------------	------------------

Description

Calculates the estimates for an MCMC object.

Usage

```
## S3 method for class 'marray'
estimates(x, fun = median, as_df = FALSE, ...)
```

Arguments

x	An object.
fun	A function that given a numeric vector returns a numeric scalar.
as_df	A flag indicating whether to return the values as a data frame versus a named list.
...	Optional arguments to fun.

Value

A named list or data frame.

See Also

Other MCMC manipulations: `bind_chains()`, `bind_iterations()`, `collapse_chains()`, `split_chains()`

estimates.mcmc	<i>Estimates</i>
----------------	------------------

Description

Calculates the estimates for an MCMC object.

Usage

```
## S3 method for class 'mcmc'
estimates(x, fun = median, as_df = FALSE, ...)
```

Arguments

x	An object.
fun	A function that given a numeric vector returns a numeric scalar.
as_df	A flag indicating whether to return the values as a data frame versus a named list.
...	Optional arguments to fun.

Value

A named list or data frame.

See Also

Other MCMC manipulations: [bind_chains\(\)](#), [bind_iterations\(\)](#), [collapse_chains\(\)](#), [split_chains\(\)](#)

estimates.mcmc.list	<i>Estimates</i>
---------------------	------------------

Description

Calculates the estimates for an MCMC object.

Usage

```
## S3 method for class 'mcmc.list'
estimates(x, fun = median, as_df = FALSE, ...)
```

Arguments

x	An object.
fun	A function that given a numeric vector returns a numeric scalar.
as_df	A flag indicating whether to return the values as a data frame versus a named list.
...	Optional arguments to fun.

Value

A named list or data frame.

See Also

Other MCMC manipulations: [bind_chains\(\)](#), [bind_iterations\(\)](#), [collapse_chains\(\)](#), [split_chains\(\)](#)

estimates.mcmcarray *Estimates*

Description

Calculates the estimates for an MCMC object.

Usage

```
## S3 method for class 'mcmcarray'
estimates(x, fun = median, as_df = FALSE, ...)
```

Arguments

x	An object.
fun	A function that given a numeric vector returns a numeric scalar.
as_df	A flag indicating whether to return the values as a data frame versus a named list.
...	Optional arguments to fun.

Value

A named list or data frame.

See Also

Other MCMC manipulations: [bind_chains\(\)](#), [bind_iterations\(\)](#), [collapse_chains\(\)](#), [split_chains\(\)](#)

estimates.mcmcr *Estimates*

Description

Calculates the estimates for an MCMC object.

Usage

```
## S3 method for class 'mcmcr'
estimates(x, fun = median, as_df = FALSE, ...)
```

Arguments

x	An object.
fun	A function that given a numeric vector returns a numeric scalar.
as_df	A flag indicating whether to return the values as a data frame versus a named list.
...	Optional arguments to fun.

Value

A named list or data frame.

See Also

Other MCMC manipulations: [bind_chains\(\)](#), [bind_iterations\(\)](#), [collapse_chains\(\)](#), [split_chains\(\)](#)

Examples

```
estimates(mcmcr_example)
```

fill_all.marray	<i>Fill All Values</i>
-----------------	------------------------

Description

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

Usage

```
## S3 method for class 'marray'
fill_all(x, value = 0, nas = TRUE, ...)
```

Arguments

x	An object.
value	A scalar of the value to replace values with.
nas	A flag specifying whether to also fill missing values.
...	Other arguments passed to methods.

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_all.mcmcarray *Fill All Values*

Description

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

Usage

```
## S3 method for class 'mcmcarray'
fill_all(x, value = 0, nas = TRUE, ...)
```

Arguments

x	An object.
value	A scalar of the value to replace values with.
nas	A flag specifying whether to also fill missing values.
...	Other arguments passed to methods.

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_all.mcmcr

Fill All Values

Description

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

Usage

```
## S3 method for class 'mcmcr'
fill_all(x, value = 0, nas = TRUE, ...)
```

Arguments

x	An object.
value	A scalar of the value to replace values with.
nas	A flag specifying whether to also fill missing values.
...	Other arguments passed to methods.

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)
```

is.marray

Is marray Object

Description

Tests whether an object is an marray.

Usage

```
is.marray(x)
```

Arguments

x The object to test.

Value

A flag indicating whether the test was positive.

Examples

```
is.marray(mcmcr_example)
```

is.mcmcarray	<i>Is mcmcarray Object</i>
--------------	----------------------------

Description

Tests whether an object is an `mcmcarray-object()`.

Usage

```
is.mcmcarray(x)
```

Arguments

x The object to test.

Value

A flag indicating whether the test was positive.

Examples

```
is.mcmcarray(mcmcr_example$beta)
```

is.mcmcr	<i>Is mcmcr Object</i>
----------	------------------------

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)
```

is.mcmcrs *Is mcmcrs Object*

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.mcmcrs(mcmcrs(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

```
mcmcr_example$beta
```

mcmcr-object	<i>mcmcr</i>
--------------	--------------

Description

An mcmcr object stores multiple uniquely named `mcmcrarray-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

```
mcmcr_example
```

mcmcrs	<i>Create mcmcrs</i>
--------	----------------------

Description

Creates an `mcmcrs-object()` from multiple `link{mcmcr-object}s`.

Usage

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

Arguments

... Objects of class mcmcr.

Value

An object of class mcmcrs

Examples

```
mcmcrs(mcmcr_example, mcmcr_example)
```

mcmcrs-object	<i>mcmcrs</i>
---------------	---------------

Description

An `mcmcrs` object stores multiple `mcmc_r-object()`s 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

```
mcmcrs(mcmc_r_example, mcmc_r_example)
```

mcmc_r_example	<i>An Example mcmc_r Object</i>
----------------	---------------------------------

Description

An example `mcmc_r-object()` derived from `coda::[line][coda::line]`.

Usage

```
mcmc_r_example
```

Format

An object of class `mcmc_r` of length 3.

Examples

```
mcmc_r_example
```

mcmc_aperm	<i>MCMC Object Transposition</i>
------------	----------------------------------

Description

Transpose an MCMC object by permuting its parameter dimensions.

Usage

```
mcmc_aperm(x, perm, ...)
```

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

mcmc_map	<i>MCMC Map</i>
----------	-----------------

Description

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

Usage

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

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.

Examples

```
mcmc_map(mcmcr_example$beta, exp)
```

nchains.marray *Number of Chains*

Description

Gets the number of chains of an MCMC object.

Usage

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

Arguments

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

Value

An integer scalar of the number of chains.

See Also

Other MCMC dimensions: [niters\(\)](#), [npars\(\)](#), [nsams\(\)](#), [nsims\(\)](#), [nterms\(\)](#)

nchains.mcmcarray *Number of Chains*

Description

Gets the number of chains of an MCMC object.

Usage

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

Arguments

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

Value

An integer scalar of the number of chains.

See Also

Other MCMC dimensions: [niters\(\)](#), [npars\(\)](#), [nsams\(\)](#), [nsims\(\)](#), [nterms\(\)](#)

nchains.mcmc	<i>Number of Chains</i>
--------------	-------------------------

Description

Gets the number of chains of an MCMC object.

Usage

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

Arguments

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

Value

An integer scalar of the number of chains.

See Also

Other MCMC dimensions: [niters\(\)](#), [npars\(\)](#), [nsams\(\)](#), [nsims\(\)](#), [nterms\(\)](#)

nchains.mcmcrs	<i>Number of Chains</i>
----------------	-------------------------

Description

Gets the number of chains of an MCMC object.

Usage

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

Arguments

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

Value

An integer scalar of the number of chains.

See Also

Other MCMC dimensions: [niters\(\)](#), [npars\(\)](#), [nsams\(\)](#), [nsims\(\)](#), [nterms\(\)](#)

niters.marray	<i>Number of Iterations</i>
---------------	-----------------------------

Description

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

Usage

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

Arguments

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

Value

An integer scalar of the number of iterations.

See Also

Other MCMC dimensions: [nchains\(\)](#), [npars\(\)](#), [nsams\(\)](#), [nsims\(\)](#), [nterms\(\)](#)

niters.mcmcarray	<i>Number of Iterations</i>
------------------	-----------------------------

Description

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

Usage

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

Arguments

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

Value

An integer scalar of the number of iterations.

See Also

Other MCMC dimensions: [nchains\(\)](#), [npars\(\)](#), [nsams\(\)](#), [nsims\(\)](#), [nterms\(\)](#)

niters.mcmc	<i>Number of Iterations</i>
-------------	-----------------------------

Description

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

Usage

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

Arguments

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

Value

An integer scalar of the number of iterations.

See Also

Other MCMC dimensions: [nchains\(\)](#), [npars\(\)](#), [nsams\(\)](#), [nsims\(\)](#), [nterms\(\)](#)

niters.mcmcrs	<i>Number of Iterations</i>
---------------	-----------------------------

Description

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

Usage

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

Arguments

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

Value

An integer scalar of the number of iterations.

See Also

Other MCMC dimensions: [nchains\(\)](#), [npars\(\)](#), [nsams\(\)](#), [nsims\(\)](#), [nterms\(\)](#)

npars.marray *Number of Parameters*

Description

Gets the number of parameters of an object.

The default methods returns the length of `pars()` if none are NA, otherwise it returns NA.

Usage

```
## S3 method for class 'marray'
npars(x, scalar = NULL, ...)
```

Arguments

x	An object.
scalar	A logical scalar specifying whether to include all parameters (NULL), only scalars (TRUE) or all parameters except scalars (FALSE).
...	Other arguments passed to methods.

Value

An integer scalar of the number of parameters.

See Also

[pars\(\)](#)

Other MCMC dimensions: [nchains\(\)](#), [niters\(\)](#), [nsams\(\)](#), [nsims\(\)](#), [nterms\(\)](#)

Other parameters: [pars\(\)](#), [set_pars\(\)](#)

npars.mcmcarray *Number of Parameters*

Description

Gets the number of parameters of an object.

The default methods returns the length of `pars()` if none are NA, otherwise it returns NA.

Usage

```
## S3 method for class 'mcmcarray'
npars(x, scalar = NULL, ...)
```

Arguments

x	An object.
scalar	A logical scalar specifying whether to include all parameters (NULL), only scalars (TRUE) or all parameters except scalars (FALSE).
...	Other arguments passed to methods.

Value

An integer scalar of the number of parameters.

See Also

[pars\(\)](#)

Other MCMC dimensions: [nchains\(\)](#), [niters\(\)](#), [nsams\(\)](#), [nsims\(\)](#), [nterms\(\)](#)

Other parameters: [pars\(\)](#), [set_pars\(\)](#)

npars.mcmc

Number of Parameters

Description

Gets the number of parameters of an object.

The default methods returns the length of [pars\(\)](#) if none are NA, otherwise it returns NA.

Usage

```
## S3 method for class 'mcmc'
npars(x, scalar = NULL, ...)
```

Arguments

x	An object.
scalar	A logical scalar specifying whether to include all parameters (NULL), only scalars (TRUE) or all parameters except scalars (FALSE).
...	Other arguments passed to methods.

Value

An integer scalar of the number of parameters.

See Also

[pars\(\)](#)

Other MCMC dimensions: [nchains\(\)](#), [niters\(\)](#), [nsams\(\)](#), [nsims\(\)](#), [nterms\(\)](#)

Other parameters: [pars\(\)](#), [set_pars\(\)](#)

npdims.mcmcarray *Number of Parameter Dimensions*

Description

Gets the number of the dimensions of each parameter of an object.

The default methods returns the length of each element of [pdims\(\)](#) as an integer vector.

Usage

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

Arguments

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

Value

A named integer vector of the number of dimensions of each parameter.

See Also

Other dimensions: [dims\(\)](#), [ndims\(\)](#), [pdims\(\)](#)

npdims.mcmcr *Number of Parameter Dimensions*

Description

Gets the number of the dimensions of each parameter of an object.

The default methods returns the length of each element of [pdims\(\)](#) as an integer vector.

Usage

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

Arguments

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

Value

A named integer vector of the number of dimensions of each parameter.

See Also

Other dimensions: [dims\(\)](#), [ndims\(\)](#), [pdims\(\)](#)

nterms.mcmcarray	<i>Number of Terms</i>
------------------	------------------------

Description

Gets the number of terms of an MCMC object.

Usage

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

Arguments

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

Value

A integer scalar of the number of terms.

See Also

Other MCMC dimensions: [nchains\(\)](#), [niters\(\)](#), [npars\(\)](#), [nsams\(\)](#), [nsims\(\)](#)

nterms.mcmcr	<i>Number of Terms</i>
--------------	------------------------

Description

Gets the number of terms of an MCMC object.

Usage

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

Arguments

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

Value

A integer scalar of the number of terms.

See Also

Other MCMC dimensions: [nchains\(\)](#), [niters\(\)](#), [npars\(\)](#), [nsams\(\)](#), [nsims\(\)](#)

nterms.mcmcrcs	<i>Number of Terms</i>
----------------	------------------------

Description

Gets the number of terms of an MCMC object.

Usage

```
## S3 method for class 'mcmcrcs'
nterms(x, ...)
```

Arguments

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

Value

A integer scalar of the number of terms.

See Also

Other MCMC dimensions: [nchains\(\)](#), [niters\(\)](#), [npars\(\)](#), [nsams\(\)](#), [nsims\(\)](#)

params	<i>Parameter Descriptions</i>
--------	-------------------------------

Description

Parameter Descriptions

Arguments

x	An object.
scalar	A logical scalar specifying whether to include all parameters (NULL), only scalars (TRUE) or all parameters except scalars (FALSE).
terms	A logical scalar specifying whether to provide the parameters for each term.
nas	A flag specifying whether to also fill missing values.
nthin	A positive integer of the thinning 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.
fun	A function that given a numeric vector returns a numeric scalar.
bound	flag specifying whether to bind mcmcrcs objects by their chains before calculating rhat.
rhat	The maximum rhat value.

esr	The minimum effective sampling rate.
na_rm	A flag specifying whether to ignore missing values.
parameters	A character vector (or NULL) of the parameters to subset by.
iterations	An integer vector (or NULL) of the iterations to subset by.
...	Unused.

pars.mcmc *Parameter Names*

Description

Gets the parameter names.

Usage

```
## S3 method for class 'mcmc'
pars(x, scalar = NULL, terms = FALSE, ...)
```

Arguments

x	An object.
scalar	A logical scalar specifying whether to include all parameters (NULL), only scalars (TRUE) or all parameters except scalars (FALSE).
terms	A logical scalar specifying whether to provide the parameters for each term.
...	Other arguments passed to methods.

Value

A character vector of the names of the parameters.

See Also

Other parameters: [npars\(\)](#), [set_pars\(\)](#)

pars.mmcrcs *Parameter Names*

Description

Gets the parameter names.

Usage

```
## S3 method for class 'mcrcs'
pars(x, scalar = NULL, terms = FALSE, ...)
```

Arguments

x	An object.
scalar	A logical scalar specifying whether to include all parameters (NULL), only scalars (TRUE) or all parameters except scalars (FALSE).
terms	A logical scalar specifying whether to provide the parameters for each term.
...	Other arguments passed to methods.

Value

A character vector of the names of the parameters.

See Also

Other parameters: [npars\(\)](#), [set_pars\(\)](#)

pdims.marray

Parameter Dimensions

Description

Gets the dimensions of each parameter of an object.

Usage

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

Arguments

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

Value

A named list of integer vectors of the dimensions of each parameter.

See Also

Other dimensions: [dims\(\)](#), [ndims\(\)](#), [npdims\(\)](#)

pdims.mcmcarray *Parameter Dimensions*

Description

Gets the dimensions of each parameter of an object.

Usage

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

Arguments

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

Value

A named list of integer vectors of the dimensions of each parameter.

See Also

Other dimensions: [dims\(\)](#), [ndims\(\)](#), [npdims\(\)](#)

pdims.mcmc *Parameter Dimensions*

Description

Gets the dimensions of each parameter of an object.

Usage

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

Arguments

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

Value

A named list of integer vectors of the dimensions of each parameter.

See Also

Other dimensions: [dims\(\)](#), [ndims\(\)](#), [npdims\(\)](#)

rhat.marray	<i>R-hat</i>
-------------	--------------

Description

Calculates an R-hat (potential scale reduction factor) value.

Usage

```
## S3 method for class 'marray'  
rhat(x, by = "all", as_df = FALSE, na_rm = FALSE, ...)
```

Arguments

x	An object.
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.
na_rm	A flag specifying whether to ignore missing values.
...	Other arguments passed to methods.

Details

By default the uncorrected, unfolded, univariate, split R-hat value.

Value

A number ≥ 1 indicating the rhat value.

References

Gelman, A., and Rubin, D.B. 1992. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science* 7(4): 457–472.

See Also

Other convergence: [converged_pars\(\)](#), [converged_terms\(\)](#), [converged\(\)](#), [esr_pars\(\)](#), [esr_terms\(\)](#), [esr\(\)](#), [rhat_pars\(\)](#), [rhat_terms\(\)](#)

rhat.mcmc	<i>R-hat</i>
-----------	--------------

Description

Calculates an R-hat (potential scale reduction factor) value.

Usage

```
## S3 method for class 'mcmc'  
rhat(x, by = "all", as_df = FALSE, na_rm = FALSE, ...)
```

Arguments

x	An object.
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.
na_rm	A flag specifying whether to ignore missing values.
...	Other arguments passed to methods.

Details

By default the uncorrected, unfolded, univariate, split R-hat value.

Value

A number ≥ 1 indicating the rhat value.

References

Gelman, A., and Rubin, D.B. 1992. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science* 7(4): 457–472.

See Also

Other convergence: [converged_pars\(\)](#), [converged_terms\(\)](#), [converged\(\)](#), [esr_pars\(\)](#), [esr_terms\(\)](#), [esr\(\)](#), [rhat_pars\(\)](#), [rhat_terms\(\)](#)

rhat.mcmc.list	<i>R-hat</i>
----------------	--------------

Description

Calculates an R-hat (potential scale reduction factor) value.

Usage

```
## S3 method for class 'mcmc.list'  
rhat(x, by = "all", as_df = FALSE, na_rm = FALSE, ...)
```

Arguments

x	An object.
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.
na_rm	A flag specifying whether to ignore missing values.
...	Other arguments passed to methods.

Details

By default the uncorrected, unfolded, univariate, split R-hat value.

Value

A number ≥ 1 indicating the rhat value.

References

Gelman, A., and Rubin, D.B. 1992. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science* 7(4): 457–472.

See Also

Other convergence: [converged_pars\(\)](#), [converged_terms\(\)](#), [converged\(\)](#), [esr_pars\(\)](#), [esr_terms\(\)](#), [esr\(\)](#), [rhat_pars\(\)](#), [rhat_terms\(\)](#)

rhat.mcmcarray	<i>R-hat</i>
----------------	--------------

Description

Calculates an R-hat (potential scale reduction factor) value.

Usage

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

Arguments

x	An object.
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.
na_rm	A flag specifying whether to ignore missing values.
...	Other arguments passed to methods.

Details

By default the uncorrected, unfolded, univariate, split R-hat value.

Value

A number ≥ 1 indicating the rhat value.

References

Gelman, A., and Rubin, D.B. 1992. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science* 7(4): 457–472.

See Also

Other convergence: [converged_pars\(\)](#), [converged_terms\(\)](#), [converged\(\)](#), [esr_pars\(\)](#), [esr_terms\(\)](#), [esr\(\)](#), [rhat_pars\(\)](#), [rhat_terms\(\)](#)

`rhat.mcmc`*R-hat*

Description

Calculates an R-hat (potential scale reduction factor) value.

Usage

```
## S3 method for class 'mcmc'  
rhat(x, by = "all", as_df = FALSE, na_rm = FALSE, ...)
```

Arguments

<code>x</code>	An object.
<code>by</code>	A string indicating whether to determine by "term", "parameter" or "all".
<code>as_df</code>	A flag indicating whether to return the values as a data frame versus a named list.
<code>na_rm</code>	A flag specifying whether to ignore missing values.
<code>...</code>	Other arguments passed to methods.

Details

By default the uncorrected, unfolded, univariate, split R-hat value.

Value

A number ≥ 1 indicating the rhat value.

References

Gelman, A., and Rubin, D.B. 1992. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science* 7(4): 457–472.

See Also

Other convergence: [converged_pars\(\)](#), [converged_terms\(\)](#), [converged\(\)](#), [esr_pars\(\)](#), [esr_terms\(\)](#), [esr\(\)](#), [rhat_pars\(\)](#), [rhat_terms\(\)](#)

Examples

```
rhat(mcmc_example)  
rhat(mcmc_example, by = "parameter")  
rhat(mcmc_example, by = "term")  
rhat(mcmc_example, by = "term", as_df = TRUE)
```

rhat.mcmcrcs	<i>R-hat</i>
--------------	--------------

Description

Calculates an R-hat (potential scale reduction factor) value.

Usage

```
## S3 method for class 'mcmcrcs'  
rhat(x, by = "all", as_df = FALSE, na_rm = FALSE, bound = FALSE, ...)
```

Arguments

x	An object.
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.
na_rm	A flag specifying whether to ignore missing values.
bound	flag specifying whether to bind mcmcrcs objects by their chains before calculating rhat.
...	Other arguments passed to methods.

Details

By default the uncorrected, unfolded, univariate, split R-hat value.

Value

A number ≥ 1 indicating the rhat value.

References

Gelman, A., and Rubin, D.B. 1992. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science* 7(4): 457–472.

See Also

Other convergence: [converged_pars\(\)](#), [converged_terms\(\)](#), [converged\(\)](#), [esr_pars\(\)](#), [esr_terms\(\)](#), [esr\(\)](#), [rhat_pars\(\)](#), [rhat_terms\(\)](#)

Examples

```
rhat(mcmcrcs(mcmcr_example, mcmcr_example))  
rhat(mcmcrcs(mcmcr_example, mcmcr_example), bound = TRUE)
```

set_pars.mcmcr	<i>Set Parameters</i>
----------------	-----------------------

Description

Sets an object's parameter names.

The assignment version `pars<-()` forwards to `set_pars()`.

Usage

```
## S3 method for class 'mcmcr'
set_pars(x, value, ...)
```

Arguments

x	An object.
value	A character vector of the new parameter names.
...	Other arguments passed to methods.

Details

value must be a unique character vector of the same length as the object's parameters.

Value

The modified object.

See Also

Other parameters: [npars\(\)](#), [pars\(\)](#)

set_pars.mcmcrs	<i>Set Parameters</i>
-----------------	-----------------------

Description

Sets an object's parameter names.

The assignment version `pars<-()` forwards to `set_pars()`.

Usage

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

Arguments

x	An object.
value	A character vector of the new parameter names.
...	Other arguments passed to methods.

Details

value must be a unique character vector of the same length as the object's parameters.

Value

The modified object.

See Also

Other parameters: [npars\(\)](#), [pars\(\)](#)

split_chains.mcmcarray

Split Chains

Description

Splits each of an MCMC object's chains in half to double the number of chains and halve the number of iterations.

Usage

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

Arguments

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

Value

The modified object.

See Also

Other MCMC manipulations: [bind_chains\(\)](#), [bind_iterations\(\)](#), [collapse_chains\(\)](#), [estimates\(\)](#)

split_chains.mcmc	<i>Split Chains</i>
-------------------	---------------------

Description

Splits each of an MCMC object's chains in half to double the number of chains and halve the number of iterations.

Usage

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

Arguments

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

Value

The modified object.

See Also

Other MCMC manipulations: [bind_chains\(\)](#), [bind_iterations\(\)](#), [collapse_chains\(\)](#), [estimates\(\)](#)

subset	<i>Subset an MCMC Object</i>
--------	------------------------------

Description

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

Usage

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

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

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

Arguments

x	The MCMC object to subset
chains	An integer vector of chains.
iters	An integer vector of iterations.
iterations	An integer vector (or NULL) of the iterations to subset by.
...	Unused.
pars	A character vector of parameter names.
parameters	A character vector (or NULL) of the parameters to subset by.

Methods (by class)

- mcmcrarray: Subset an mcmcrarray object
- mcmcr: Subset an mcmcr object
- mcmcrs: Subset an mcmcrs object

Examples

```
subset(mcmcr_example,
  chains = 2L, iters = 1:100,
  pars = c("beta", "alpha")
)
```

vld_mcmcr

Validate MCMC Objects

Description

Validates class and structure of MCMC objects.

Usage

```
vld_mcmcrarray(x)
```

```
vld_mcmcr(x)
```

```
vld_mcmcrs(x)
```

Arguments

x The object to check.

Details

To just validate class use `chk::vld_s3_class()`.

Value

A flag indicating whether the object was validated.

Functions

- `vld_mcmcarray`: Validate `mcmcarray-object()`
- `vld_mcmcr`: Validate `mcmcr-object()`
- `vld_mmcrcs`: Validate `mcrcs-object()`

See Also

[chk_mcmcr\(\)](#)

Examples

```
#' vld_mcmcarray
vld_mcmcarray(1)

# vld_mcmcr
vld_mcmcr(1)
vld_mcmcr(mcmcr::mcmcr_example)

# vld_mmcrcs
vld_mmcrcs(1)
```

zero

Zero MCMC Sample Values

Description

Zeros an MCMC object's sample values.

Usage

```
zero(x, ...)
```

```
## S3 method for class 'mccarray'
zero(x, ...)
```

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

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

Arguments

x	The MCMC object.
...	Unused
pars	A character vector (or NULL) of the pars to zero.

Details

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

Value

The MCMC

Methods (by class)

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

Examples

```
zero(mcmcr_example, pars = "beta")
```

Index

* datasets

- mcmcr_example, 38

- as.mccarray, 3
- as.mcmc.mccarray, 4
- as.mcmc.mcmc, 5
- as.mcmc.mcmcarray, 6
- as.mcmc.mcmcr, 6
- as.mcmcarray, 7
- as.mcmcr, 8
- as.mcmcr.list(as.mccarray), 3
- as.mcmcrs, 9
- as_nlist.mcmcr, 10
- as_nlists.mcmcr, 10

- bind_chains, 18, 28–31, 59, 60
- bind_chains.mccarray, 11
- bind_chains.mcmc, 12
- bind_chains.mcmc.list, 12
- bind_chains.mcmcarray, 13
- bind_chains.mcmcr, 13
- bind_dimensions, 14
- bind_dimensions(), 14
- bind_dimensions_n, 14
- bind_dimensions_n(), 14
- bind_iterations, 11–13, 18, 28–31, 59, 60
- bind_parameters, 15

- check_mcmcarray, 15
- check_mcmcr, 16
- chk::chk_s3_class(), 17
- chk::vld_s3_class(), 62
- chk_mcmcarray(chk_mcmcr), 16
- chk_mcmcr, 16
- chk_mcmcr(), 62
- chk_mcmcrs(chk_mcmcr), 16
- coda::mcmc(), 8
- coda::mcmc.list(), 8
- coef, 17
- collapse_chains, 11–13, 28–31, 59, 60
- collapse_chains.mcmcr, 18
- combine_dimensions, 19
- combine_samples, 19
- combine_samples_n, 20

- converged, 23–27, 52–57
- converged.default, 20
- converged.mcmcrs, 21
- converged_pars, 21–27, 52–57
- converged_terms, 21–27, 52–57

- dims, 46, 50, 51

- esr, 21, 22, 52–57
- esr(), 28
- esr.mccarray, 22
- esr.mcmc, 23
- esr.mcmc.list, 24
- esr.mcmcarray, 25
- esr.mcmcr, 26
- esr.mcmcrs, 27
- esr_pars, 21–27, 52–57
- esr_terms, 21–27, 52–57
- ess, 28
- estimates, 11–13, 18, 59, 60
- estimates.mccarray, 28
- estimates.mcmc, 29
- estimates.mcmc.list, 29
- estimates.mcmcarray, 30
- estimates.mcmcr, 30

- fill_all.mccarray, 31
- fill_all.mcmcarray, 32
- fill_all.mcmcr, 33
- fill_na, 32–34

- is.mccarray, 34
- is.mcmcarray, 35
- is.mcmcr, 35
- is.mcmcrs, 36

- mcmc.list, 5–7
- mcmc_aperm, 39
- mcmc_map, 39
- mcmcarray-object, 36
- mcmcarray_object(mcmcarray-object), 36
- mcmcr-object, 37
- mcmcr_example, 38
- mcmcr_object(mcmcr-object), 37
- mcmcrs, 37

- mcmcrs-object, 38
- mcmcrs_object (mcmcrs-object), 38
- mcmcUpgrade, 5–7
- nchains, 42–45, 47, 48
- nchains.marray, 40
- nchains.mmcarray, 40
- nchains.mmcrcr, 41
- nchains.mmcrcrs, 41
- ndims, 46, 50, 51
- niters, 40, 41, 44, 45, 47, 48
- niters.marray, 42
- niters.mmcarray, 42
- niters.mmcrcr, 43
- niters.mmcrcrs, 43
- nlist_object(), 10
- nlists_object(), 10
- npars, 40–43, 47–50, 58, 59
- npars.marray, 44
- npars.mmcarray, 44
- npars.mmcrcr, 45
- npdims, 50, 51
- npdims.mmcarray, 46
- npdims.mmcrcr, 46
- nsams, 40–45, 47, 48
- nsims, 40–45, 47, 48
- nterms, 40–45
- nterms.mmcarray, 47
- nterms.mmcrcr, 47
- nterms.mmcrcrs, 48
- params, 48
- pars, 44, 45, 58, 59
- pars(), 44, 45
- pars.mmcrcr, 49
- pars.mmcrcrs, 49
- pdims, 46
- pdims(), 46
- pdims.marray, 50
- pdims.mmcarray, 51
- pdims.mmcrcr, 51
- plot.mcmc, 5–7
- rhat, 21–27
- rhat.marray, 52
- rhat.mcmc, 53
- rhat.mcmc.list, 54
- rhat.mmcarray, 55
- rhat.mmcrcr, 56
- rhat.mmcrcrs, 57
- rhat_pars, 21–27, 52–57
- rhat_terms, 21–27, 52–57
- set_pars, 44, 45, 49, 50
- set_pars.mmcrcr, 58
- set_pars.mmcrcrs, 58
- split_chains, 11–13, 18, 28–31
- split_chains.mmcarray, 59
- split_chains.mmcrcr, 60
- subset, 60
- summary.mcmc, 5–7
- thin, 5–7
- vld_mmcarray (vld_mmcrcr), 61
- vld_mmcrcr, 61
- vld_mmcrcr(), 17
- vld_mmcrcrs (vld_mmcrcr), 61
- window.mcmc, 5–7
- zero, 62