Package 'mcmcr'

September 6, 2021

```
Title Manipulate MCMC Samples
Version 0.6.1
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 (>= 0.7.0),
     coda,
     extras,
     generics,
     lifecycle,
     nlist,
     purrr,
     stats,
     term,
     universals,
     utils
Suggests covr,
     graphics,
     readr,
     rlang,
     testthat,
     tibble
RdMacros lifecycle
Config/testthat/edition 3
Encoding UTF-8
Language en-US
LazyData true
Roxygen list(markdown = TRUE)
RoxygenNote 7.1.1
```

R topics documented:

as.mcarray	4
as.mcmc.mcarray	4
as.meme.meme	5
as.meme.memearray	6
as.meme.memer	6
as.mcmcarray	7
as.memer	8
as.memers	9
as_nlist.memer	10
as_nlists.mcmcr	11
bind_chains.mcarray	11
bind_chains.mcmc	12
bind_chains.mcmc.list	13
bind_chains.mcmcarray	13
bind_chains.mcmcr	14
bind_dimensions	14
bind_dimensions_n	15
	15
•	16
	16
-	17
	18
	19
	19
	20
•	 21
•	 21
	 22
	 23
	24
	- · 25
	26
	20 27
	21 28
	20 29
	29
, and the second	2) 30
	30
	30 31
·	31
	31 32
· · · · · · · · · · · · · · · · · · ·	32 33
_	33
-	35 26
-	36
	37
	38
	39
	39
s memers	40

R	topics	documented:
---	--------	-------------

mcmcarray-object
mcmcr-object
memers
mcmcrs-object
mcmcr_example
mcmc_aperm
mcmc_map
nchains.mcarray
nchains.mcmcarray
nchains.mcmcr
nchains.mcmcrs
niters.mcarray
niters.mcmcarray
niters.mcmcr
niters.mcmcrs
npars.mcarray
npars.mcmcarray
npars.mcmcr
npdims.mcmcarray
npdims.memer
nterms.mcmcarray
nterms.mcmcr
nterms.mcmcrs
params
pars.mcmcr
pars.mcmcrs
pdims.mcarray
pdims.mcmcarray
pdims.mcmcr
rhat.mcarray
rhat.mcmc
rhat.mcmc.list
rhat.mcmcarray
rhat.mcmcr
rhat.mcmcrs
set_pars.mcmcr
set_pars.mcmcrs
split_chains.mcmcarray
split_chains.mcmcr
subset
vld memer

4 as.mcmc.mcarray

as.mcarray

Coerce to an mcarray object

Description

Coerces MCMC objects to an mcarray object.

Usage

```
as.mcarray(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 [mcmcarray-object]s to an mcmcr object

See Also

```
Other coerce: as.mcmcarray(), as.mcmcr(), mcmcrs()
```

Examples

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

as.mcmc.mcarray

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 meme 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.

as.memc.memc 5

Usage

```
## S3 method for class 'mcarray'
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 meme 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

```
mcmc.list, mcmcUpgrade, thin, window.mcmc, summary.mcmc, plot.mcmc.
```

6 as.mcmc.mcmcr

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 meme 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 meme 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.

as.mcmcarray 7

Usage

```
## S3 method for class 'mcmcr'
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

Coerce to an mcmcarray object

Description

Coerces MCMC objects to an mcmcarray-object().

Usage

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

Arguments

x object to coerce.

... Unused.

See Also

```
Other coerce: as.mcarray(), as.mcmcr(), mcmcrs()
```

Examples

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

8 as.memer

as.mcmcr

Convert to an mcmcr object

Description

Converts an MCMC object to an mcmcr-object().

Usage

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

Arguments

x An MCMC object. . . . Unused.

name A string specifying the parameter name.

Value

An mcmcr object.

Methods (by class)

- mcarray: Convert an mcarray object to an mcmcr object
- mcmcarray: Convert an mcmcarray-object() to an mcmcr object
- nlist: Convert an nlist::nlist-object() to an mcmcr object
- nlists: Convert an nlist::nlists-object() to an mcmcr object
- mcmc: Convert an coda::mcmc() object to an mcmcr object
- mcmc.list: Convert an coda::mcmc.list() object to an mcmcr object
- mcmcrs: Convert tan mcmcrs-object() to an mcmcr object

as.mcmcrs 9

See Also

```
Other coerce: as.mcarray(), as.mcmcarray(), mcmcrs()
```

Examples

```
mcmc.list <- coda::as.mcmc.list(mcmcr::mcmcr_example)
as.mcmcr(mcmc.list)</pre>
```

as.mcmcrs

Convert to an mcmcrs object

Description

Converts an MCMC object to an mcmcrs-object().

Usage

```
as.mcmcrs(x, ...)
## S3 method for class 'list'
as.mcmcrs(x, ...)
## S3 method for class 'mcmcr'
as.mcmcrs(x, name = "mcmcr1", ...)
```

Arguments

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

name A string specifying the element name.

Value

An mcmcrs object.

Methods (by class)

- list: Convert a list of [mcmcr-object]s to an mcmcrs object
- mcmcr: Convert an mcmcr-object() to an mcmcrs object

Examples

```
as.mcmcrs(mcmcr::mcmcr_example)
```

10 as_nlist.mcmcr

as_nlist.mcmcr

Coerce to nlist

Description

Coerce an R object to an nlist_object().

Usage

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

Arguments

x An object.

... 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

See Also

```
Other coerce: as_nlists()
```

Examples

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

as_nlists.mcmcr 11

as_nlists.mcmcr

Coerce to nlists

Description

Coerce an R object to an nlists_object().

Usage

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

Arguments

x An object. ... 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

See Also

```
Other coerce: as_nlist()
```

Examples

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

bind_chains.mcarray

Bind by Chains.

Description

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

Usage

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

12 bind_chains.mcmc

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

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.

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

bind_chains.mcmc.list 13

```
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.

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

bind_dimensions

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

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, ...)
```

Arguments

x An MCMC object.x2 a second MCMC object.

along A count (or NULL) indicating the parameter dimension to bind along.

... Unused.

```
universals::bind_chains()
Other bind: bind_dimensions_n(), bind_parameters()
```

bind_dimensions_n 15

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(...)
```

Arguments

... one or more MCMC objects

See Also

```
universals::bind_chains()
Other bind: bind_dimensions(), bind_parameters()
```

Examples

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

bind_parameters

Combine two MCMC object by parameters

Description

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

Usage

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

Arguments

```
x an MCMC objectx2 a second MCMC object.... Unused.
```

```
universals::bind_chains()
Other bind: bind_dimensions_n(), bind_dimensions()
```

16 check_mcmcr

Examples

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

check_mcmcarray

[Soft-deprecated] Check mcmcarray

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).

See Also

```
check_mcmcr()
```

Examples

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

check_mcmcr

[Soft-deprecated] Check mcmcr

Description

[Soft-deprecated] Check mcmcr

Usage

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

chk_mcmcr 17

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).

See Also

```
check_mcmcr()
```

Examples

```
check_mcmcr(mcmcr::mcmcr_example)
```

chk_mcmcr

Check MCMC objects

Description

```
Checks class and structure of MCMC objects.
```

chk_mcmcrs checks if an mcmcrs-object().

```
chk_mcmcarray checks if mcmcarray-object() object using
is.array(x) && is.numeric(x)
chk_mcmcr checks if an mcmcr-object().
```

Usage

```
chk_mcmcarray(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.

18 coef

Functions

- chk_mcmcarray: Check mcmcarray Object
- chk_mcmcr: Check mcmcr Object
- chk_mcmcrs: Check mcmcrs 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_mcmcrs
chk_mcmcrs(as.mcmcrs(as.mcmcr(list(x = 1))))
try(chk_mcmcrs(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 = TRUE, ...)
```

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, lower and upper credible intervals and svalue

Methods (by class)

• mcmc: Get coefficients for terms in mcmc object

collapse_chains.mcmcr 19

See Also

```
stats::coef
```

Examples

```
coef(mcmcr_example)
```

```
collapse_chains.mcmcr Collapse Chains
```

Description

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

Usage

```
## S3 method for class 'mcmcr'
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

Combine samples by dimensions

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.

20 combine_samples

Value

The MCMC object with reduced dimensions.

See Also

```
Other combine: combine_samples_n(), combine_samples()
```

Examples

```
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

```
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.

See Also

```
Other combine: combine_dimensions(), combine_samples_n()
```

Examples

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

combine_samples_n 21

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

```
combine\_samples\_n(x, ..., fun = mean)
```

Arguments

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

See Also

```
Other combine: combine_dimensions(), combine_samples()
```

Examples

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

converged.default

Converged

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,
    ...
)
```

22 converged.mcmcrs

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 results 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.mcmcrs

Converged

Description

Tests whether an object has converged.

Usage

```
## $3 method for class 'mcmcrs'
converged(
    x,
    rhat = 1.1,
    esr = 0.33,
    by = "all",
    as_df = FALSE,
    bound = FALSE,
    na_rm = FALSE,
    ...
)
```

esr.mcarray 23

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 results 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.mcarray

Effective Sampling Rate

Description

Calculates the effective sampling rate (esr).

Usage

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

Х	An 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 versus a named list.
na_rm	A flag specifying whether to ignore missing values.
	Other arguments passed to methods.

24 esr.mcmc

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 results 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 $\rho_{k+1}(\theta) < 0$.

esr.mcmc.list 25

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 results 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.

26 esr.mcmcarray

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 results 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.

```
Other convergence: converged_pars(), converged_terms(), converged(), esr_pars(), esr_terms(), rhat_pars(), rhat_terms(), rhat()
```

esr.mcmcr 27

esr	m	\sim	m	\sim	r

Effective Sampling Rate

Description

Calculates the effective sampling rate (esr).

Usage

```
## S3 method for class 'mcmcr'
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 results 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(mcmcr_example)
```

28 esr.mcmcrs

esr	m	CIT	10	rs

Effective Sampling Rate

Description

Calculates the effective sampling rate (esr).

Usage

```
## S3 method for class 'mcmcrs'
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 results 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(mcmcrs(mcmcr_example, mcmcr_example))
```

ess 29

ess

P-Value effective sample size

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 versus a named

list.

See Also

universals::esr

Examples

```
ess(mcmcr_example)
```

estimates.mcarray

Estimates

Description

Calculates the estimates for an MCMC object.

Usage

```
## S3 method for class 'mcarray'
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 results as a data frame versus a named

list.

... Optional arguments to fun.

Value

A named list or data frame.

30 estimates.mcmc.list

See Also

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

estimates.mcmc

Estimates

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 results 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()

```
{\tt estimates.mcmc.list} \qquad \textit{Estimates}
```

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 results as a data frame versus a named

list.

... Optional arguments to fun.

estimates.mcmcarray 31

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 results 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, ...)
```

32 fill_all.mcarray

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 results 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.mcarray 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 'mcarray'
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.

Details

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

Value

The modified object.

fill_all.mcmcarray 33

Methods (by class)

logical: Fill All for logical Objectsinteger: 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.

Details

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

fill_all.mcmcr

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)
```

 $\verb|fill_all.mcmcr|$

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, ...)
```

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.

fill_na.mcarray 35

Details

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

Value

The modified object.

Methods (by class)

- logical: Fill All for logical Objects
- integer: Fill All for integer Objects
- numeric: Fill All for numeric Objects
- character: Fill All for character Objects

See Also

```
Other fill: fill_na()
```

Examples

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

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

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

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

fill_na.mcarray

Fill Missing Values

Description

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

Usage

```
## S3 method for class 'mcarray'
fill_na(x, value = 0, ...)
```

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

36 fill_na.mcmcarray

Details

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

Value

The modified object.

Methods (by class)

- logical: Fill Missing Values for logical Objects
- integer: Fill Missing Values for integer Objects
- numeric: Fill Missing Values for numeric Objects
- character: Fill Missing Values for character Objects

See Also

```
Other fill: fill_all()
```

Examples

```
# logical
fill_na(c(TRUE, NA))

# integer
fill_na(c(1L, NA), 0)

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

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

fill_na.mcmcarray

Fill Missing Values

Description

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

Usage

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

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

fill_na.memer 37

Details

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

Value

The modified object.

Methods (by class)

- logical: Fill Missing Values for logical Objects
- integer: Fill Missing Values for integer Objects
- numeric: Fill Missing Values for numeric Objects
- character: Fill Missing Values for character Objects

See Also

```
Other fill: fill_all()
```

Examples

```
# logical
fill_na(c(TRUE, NA))

# integer
fill_na(c(1L, NA), 0)

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

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

fill_na.mcmcr

Fill Missing Values

Description

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

Usage

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

Arguments

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

is.mcarray

Details

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

Value

The modified object.

Methods (by class)

- logical: Fill Missing Values for logical Objects
- integer: Fill Missing Values for integer Objects
- numeric: Fill Missing Values for numeric Objects
- character: Fill Missing Values for character Objects

See Also

```
Other fill: fill_all()
```

Examples

```
# logical
fill_na(c(TRUE, NA))

# integer
fill_na(c(1L, NA), 0)

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

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

is.mcarray

Is mcarray object

Description

Tests whether an object is an mcarray.

Usage

```
is.mcarray(x)
```

Arguments

Х

The object to test.

Value

A flag indicating whether the test was positive.

is.mcmcarray 39

See Also

```
Other is: is.mcmcarray(), is.mcmcrs(), is.mcmcr()
```

Examples

```
is.mcarray(mcmcr_example)
```

is.mcmcarray

Is mcmcarray object

Description

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

Usage

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

Arguments

Х

The object to test.

Value

A flag indicating whether the test was positive.

See Also

```
Other is: is.mcarray(), is.mcmcrs(), is.mcmcr()
```

Examples

is.mcmcarray(mcmcr_example\$beta)

is.mcmcr

Is mcmcr object

Description

Tests whether an object is an mcmcr-object().

Usage

```
is.mcmcr(x)
```

Arguments

Х

The object to test.

is.mcmcrs

Value

A flag indicating whether the test was positive.

See Also

```
Other is: is.mcarray(), is.mcmcarray(), is.mcmcrs()
```

Examples

```
is.mcmcr(mcmcr_example)
```

is.mcmcrs

Is mcmcrs object

Description

Tests whether an object is an mcmcrs-object().

Usage

```
is.mcmcrs(x)
```

Arguments

Х

The object to test.

Value

A flag indicating whether the test was positive.

See Also

```
Other is: is.mcarray(), is.mcmcarray(), is.mcmcr()
```

Examples

```
is.mcmcrs(mcmcrs(mcmcr_example))
```

mcmcarray-object 41

mcmcarray-object

mcmcarray

Description

An mcmcarray object is an 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.

See Also

Other objects: mcmcr-object, mcmcrs-object

Examples

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.

See Also

Other objects: mcmcarray-object, mcmcrs-object

Examples

mcmcr_example

42 mcmcrs-object

mcmcrs

Create mcmcrs

Description

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

Usage

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

Arguments

.. Objects of class mcmcr.

Value

An object of class memers

See Also

```
Other coerce: as.mcarray(), as.mcmcarray(), as.mcmcr()
```

Examples

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

mcmcrs-object

memers

Description

An mcmcrs object stores multiple mcmcr-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.

See Also

Other objects: mcmcarray-object, mcmcr-object

Examples

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

mcmcr_example 43

mcmcr_example

An example mcmcr object

Description

An example mcmcr-object() derived from 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, ...)
```

Arguments

The MCMC object to transpose. Missing parameter dimensions are added on the end. If perm = NULL (the default) the parameter dimensions are reversed.

perm A integer vector of the new order for the parameter dimensions.

... Unused.

Value

The modified MCMC object

See Also

Other manipulate: mcmc_map()

44 nchains.mcarray

mcmc_map

MCMC map

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.

See Also

```
Other manipulate: mcmc_aperm()
```

Examples

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

nchains.mcarray

Number of Chains

Description

Gets the number of chains of an MCMC object.

Usage

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

Arguments

x An object.

... Other arguments passed to methods.

Value

An integer scalar of the number of chains.

nchains.mcmcarray 45

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.mcmcr

Number of Chains

Description

Gets the number of chains of an MCMC object.

Usage

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

Arguments

x An object.

... Other arguments passed to methods.

Value

An integer scalar of the number of chains.

```
Other MCMC dimensions: niters(), npars(), nsams(), nsims(), nterms()
```

46 niters.mcarray

nchains.mcmcrs

Number of Chains

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.mcarray

Number of Iterations

Description

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

Usage

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

Arguments

x An object.

... Other arguments passed to methods.

Value

An integer scalar of the number of iterations.

```
Other MCMC dimensions: nchains(), npars(), nsams(), nsims(), nterms()
```

niters.mcmcarray 47

niters.mcmcarray

Number of Iterations

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.mcmcr

Number of Iterations

Description

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

Usage

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

Arguments

x An object.

... Other arguments passed to methods.

Value

An integer scalar of the number of iterations.

```
Other MCMC dimensions: nchains(), npars(), nsams(), nsims(), nterms()
```

48 npars.mcarray

niters.mcmcrs

Number of Iterations

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.mcarray

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 'mcarray'
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.

npars.mcmcarray 49

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.

```
pars()
Other MCMC dimensions: nchains(), niters(), nsams(), nsims(), nterms()
Other parameters: pars(), set_pars()
```

50 npdims.mcmcarray

npars.mcmcr

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 'mcmcr'
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.

npdims.mcmcr 51

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

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

Number of Terms

Description

Gets the number of terms of an MCMC object.

Usage

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

Arguments

An object.

52 nterms.mcmcrs

Value

A integer scalar of the number of terms.

See Also

```
Other MCMC dimensions: nchains(), niters(), npars(), nsams(), nsims()
```

nterms.mcmcr

Number of Terms

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.mcmcrs

Number of Terms

Description

Gets the number of terms of an MCMC object.

Usage

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

Arguments

An object.

params 53

Value

A integer scalar of the number of terms.

See Also

Other MCMC dimensions: nchains(), niters(), npars(), nsams(), nsims()

params Parameter descriptions

Description

Parameter descriptions

Arguments

name

٤	Guments	
	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 results 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 mcmcrs 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.
	x2	a second MCMC object.
	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.
	sorted	A flag specifying whether the parameters must be sorted.
	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.
	perm	A integer vector of the new order for the parameter dimensions.
	pars	A character vector (or NULL) of the pars to zero.

A string specifying the parameter name.

54 pars.mcmcrs

|--|--|--|

Description

Gets the parameter names.

Usage

```
## S3 method for class 'mcmcr'
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()
```

	Parameter Names	pars.mcmcrs
--	-----------------	-------------

Description

Gets the parameter names.

Usage

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

Arguments

Χ	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.

pdims.mcarray 55

Value

A character vector of the names of the parameters.

See Also

```
Other parameters: npars(), set_pars()
```

pdims.mcarray

Parameter Dimensions

Description

Gets the dimensions of each parameter of an object.

Usage

```
## S3 method for class 'mcarray' 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.

56 rhat.mcarray

Value

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

See Also

```
Other dimensions: dims(), ndims(), npdims()
```

pdims.mcmcr

Parameter Dimensions

Description

Gets the dimensions of each parameter of an object.

Usage

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

Arguments

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.mcarray

R-hat

Description

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

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

rhat.meme 57

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 results 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()
```

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 results 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.

58 rhat.mcmc.list

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

R-hat

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 results 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.

```
Other convergence: converged_pars(), converged_terms(), converged(), esr_pars(), esr_terms(), esr(), rhat_pars(), rhat_terms()
```

rhat.mcmcarray 59

|--|

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 results 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.

```
Other convergence: converged_pars(), converged_terms(), converged(), esr_pars(), esr_terms(), esr(), rhat_pars(), rhat_terms()
```

rhat.mcmcr

Description

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

Usage

```
## S3 method for class 'mcmcr'
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 results 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()
```

Examples

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

rhat.memers 61

Description

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

Usage

```
## S3 method for class 'mcmcrs'
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 results as a data frame versus a named list.
na_rm	A flag specifying whether to ignore missing values.
bound	flag specifying whether to bind mcmcrs 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(mcmcrs(mcmcr_example, mcmcr_example))
rhat(mcmcrs(mcmcr_example, mcmcr_example), bound = TRUE)
```

62 set_pars.mcmcrs

 $\mathsf{set}_\mathsf{pars}.\mathsf{mcmcr}$

Set Parameters

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()
```

 $\verb"set_pars.mcmcrs"$

Set Parameters

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

An object.

value A character vector of the new parameter names.

split_chains.mcmcarray

63

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()
```

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.

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

64 subset

```
split_chains.mcmcr
```

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 'mcmcr'
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

Subset an MCMC object

Description

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

```
## $3 method for class 'mcmcarray'
subset(x, chains = NULL, iters = NULL, iterations = NULL, ...)
## $3 method for class 'mcmcr'
subset(
    x,
    chains = NULL,
    iters = NULL,
    iters = NULL,
    iterations = NULL,
    parameters = NULL,
    ...
)
```

vld_mcmcr 65

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

Arguments

x The MCMC object to subsetchains 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)

• mcmcarray: Subset an mcmcarray object

mcmcr: Subset an mcmcr objectmcmcrs: Subset an mcmcrs object

See Also

```
universals::split_chains()
```

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.

```
vld_mcmcarray(x)
vld_mcmcr(x)
vld_mcmcrs(x)
```

66 zero

Arguments

Χ

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_mcmcrs: Validate mcmcrs-object()

See Also

```
chk_mcmcr()
```

Examples

```
#' vld_mcmcarray
vld_mcmcarray(1)

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

# vld_mcmcrs
vld_mcmcrs(1)
```

zero

Zero MCMC sample values

Description

Zeros an MCMC object's sample values.

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

zero 67

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)

• mcarray: Zero an mcarray object

• mcmcarray: Zero an mcmcarray object

• mcmcr: Zero an mcmcr object

Examples

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

Index

* bind	hind chains mame 12
	bind_chains.mcmc, 12
bind_dimensions, 14	bind_chains.mcmc.list, 13
bind_dimensions_n, 15	bind_chains.mcmcarray, 13
bind_parameters, 15	bind_chains.mcmcr, 14
* coerce	bind_dimensions, 14, 15
as.mcarray,4	bind_dimensions_n, 14, 15, 15
as.mcmcarray,7	bind_iterations, 12-14, 19, 30-32, 63, 64
as.mcmcr, 8	bind_parameters, <i>14</i> , <i>15</i> , 15
mcmcrs, 42	shoots management 16
* combine	check_mcmcarray, 16
combine_dimensions, 19	check_mcmcr, 16
combine_samples, 20	check_mcmcr(), 16, 17
combine_samples_n, 21	chk::chk_s3_class(), 17
* datasets	chk::vld_s3_class(),66
mcmcr_example,43	chk_mcmcarray (chk_mcmcr), 17
* is	chk_mcmcr, 17
is.mcarray,38	chk_mcmcr(), 66
is.mcmcarray,39	chk_mcmcrs (chk_mcmcr), 17
is.mcmcr, 39	coda::line(), <i>43</i>
is.mcmcrs, 40	coda::mcmc(), 8
* manipulate	<pre>coda::mcmc.list(), 8</pre>
mcmc_aperm, 43	coef, 18
mcmc_map, 44	collapse_chains, <i>12–14</i> , <i>30–32</i> , <i>63</i> , <i>64</i>
* objects	collapse_chains.mcmcr, 19
mcmcarray-object,41	combine_dimensions, 19, 20, 21
mcmcr-object, 41	combine_samples, 20 , 20 , 21
mcmcrs-object, 42	combine_samples_n, 20, 21
	converged, <i>24</i> – <i>28</i> , <i>57</i> – <i>61</i>
as.mcarray, 4, 7, 9, 42	converged.default, 21
as.mcmc.mcarray,4	converged.mcmcrs, 22
as.mcmc.mcmc, 5	converged_pars, 22-28, 57-61
as.mcmc.mcmcarray, 6	converged_terms, 22-28, 57-61
as.mcmc.mcmcr, 6	
as.mcmcarray, 4 , 7 , 9 , 42	dims, 51, 55, 56
as.mcmcr, 4, 7, 8, 42	
as.mcmcr.list(as.mcarray),4	esr, 22, 23, 57–61
as.mcmcrs, 9	esr(), 29
as_nlist, <i>11</i>	esr.mcarray, 23
as_nlist.mcmcr, 10	esr.mcmc, 24
as_nlists, 10	esr.mcmc.list, 25
as_nlists.mcmcr, 11	esr.mcmcarray, 26
	esr.mcmcr, 27
bind_chains, 19, 30-32, 63, 64	esr.mcmcrs, 28
bind_chains.mcarray, 11	esr_pars, 22-28, 57-61

INDEX 69

esr_terms, <i>22-28</i> , <i>57-61</i>	npdims, <i>55</i> , <i>56</i>
ess, 29	npdims.mcmcarray, 50
estimates, <i>12-14</i> , <i>19</i> , <i>63</i> , <i>64</i>	npdims.mcmcr, 51
estimates.mcarray,29	nsams, <i>45-50</i> , <i>52</i> , <i>53</i>
estimates.mcmc, 30	nsims, <i>45–50</i> , <i>52</i> , <i>53</i>
estimates.mcmc.list,30	nterms, <i>45–50</i>
estimates.mcmcarray, 31	nterms.mcmcarray, 51
estimates.mcmcr, 31	nterms.mcmcr, 52
	nterms.mcmcrs, 52
fill_all, <i>36–38</i>	52
fill_all.mcarray,32	params, 53
fill_all.mcmcarray,33	pars, 49, 50, 62, 63
fill_all.mcmcr,34	pars(), 48–50
fill_na, <i>33–35</i>	pars.mcmcr, 54
fill_na.mcarray,35	pars.mcmcrs, 54
fill_na.mcmcarray,36	pdims, 51
fill_na.mcmcr, 37	pdims(), 50, 51
	pdims.mcarray, 55
is.mcarray, 38, <i>39</i> , <i>40</i>	pdims.mcmcarray, 55
is.mcmcarray, <i>39</i> , <i>39</i> , <i>40</i>	pdims.mcmcr, 56
is.mcmcr, 39, 39, 40	plot.mcmc, $5-7$
is.mcmcrs, <i>39</i> , <i>40</i> , 40	rhat, 22–28
	rhat.mcarray, 56
mcmc.list, 5-7	rhat.mcmc, 57
mcmc_aperm, 43, 44	rhat.mcmc.list, 58
mcmc_map, 43, 44	rhat.mcmcarray, 59
mcmcarray-object,41	rhat.mcmcr, 60
mcmcarray_object(mcmcarray-object),41	rhat.mcmcrs, 61
mcmcr-object,41	rhat_pars, 22–28, 57–61
mcmcr_example, 43	rhat_terms, 22-28, 57-61
mcmcr_object (mcmcr-object), 41	11102_1011115, 22-20, 37-01
mcmcrs, 4, 7, 9, 42	set_pars, 49, 50, 54, 55
mcmcrs-object, 42	set_pars.mcmcr, 62
mcmcrs_object(mcmcrs-object), 42	set_pars.mcmcrs,62
mcmcUpgrade, 5-7	split_chains, <i>12–14</i> , <i>19</i> , <i>30–32</i>
,	split_chains.mcmcarray, 63
nchains, 46–50, 52, 53	split_chains.mcmcr, 64
nchains.mcarray,44	stats::coef, 19
nchains.mcmcarray,45	subset, 64
nchains.mcmcr, 45	summary.mcmc, 5-7
nchains.mcmcrs, 46	
ndims, <i>51</i> , <i>55</i> , <i>56</i>	thin, <i>5–7</i>
niters, 45, 46, 49, 50, 52, 53	universals::bind_chains(), 14, 15
niters.mcarray, 46	universals::esr, 29
niters.mcmcarray, 47	universals::split_chains(), 65
niters.mcmcr, 47	universalsspire_enains(), 05
niters.mcmcrs, 48	vld_mcmcarray(vld_mcmcr),65
nlist_object(), 10	vld_mcmcr, 65
nlists_object(), 11	vld_mcmcr(), <i>18</i>
npars, 45–48, 52–55, 62, 63	vld_mcmcrs (vld_mcmcr), 65
npars.mcarray, 48	order decrease 5. 7
npars.mcmcarray, 49	window.mcmc, $5-7$
npars.mcmcr, 50	zero, 66
	· · · · · · · · · · · · · · · · · · ·