Package 'nlist'

June 25, 2020

Title Lists of Numeric Atomic Objects

Version 0.2.0

Description Create and manipulate numeric list ('nlist') objects.
An 'nlist' is an S3 list of uniquely named numeric objects.
An numeric object is an integer or double vector, matrix or array.
An 'nlists' object is a S3 class list of 'nlist' objects with the
same names, dimensionalities and typeofs. Numeric list objects are of
interest because they are the raw data inputs for analytic engines
such as 'JAGS', 'STAN' and 'TMB'. Numeric lists objects, which are
useful for storing multiple realizations of of simulated data sets,
can be converted to coda::mcmc and coda::mcmc.list objects.

License MIT + file LICENSE

URL https://github.com/poissonconsulting/nlist

BugReports https://github.com/poissonconsulting/nlist/issues

Depends R (>= 3.4)

Imports stats,
chk,
term (>= 0.2.0),
coda,
abind,
purrr,
generics,
tibble,
universals,
extras,
lifecycle

Suggests covr,
testthat,
rlang

RdMacros lifecycle

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.0.9000

R topics documented:

`aggregate.nlist` *Aggregate nlist*

Description

Aggregates an `nlist_object()` into a named list of numeric scalars.

Usage

```
## S3 method for class 'nlist'
aggregate(x, fun = mean, ...)
```

Arguments

`x`	An nlist object.
`fun`	A function that given a numeric vector returns a numeric scalar.
`...`	Additional arguments passed to fun.

Value

An named list of numeric scalars

Examples

```
aggregate(nlist(x = 1:9))
aggregate(nlist(y = 3:5, zz = matrix(1:9, 3)), fun = function(x) x[1])
```

`aggregate.nlists` *Aggregate nlists*

Description

Aggregates an `nlists_object()` into a `nlist_object()` or by_chain = TRUE an `nlists_object()` with `nchains` `nlist_object()`s.

Usage

```
## S3 method for class 'nlists'
aggregate(x, fun = mean, ..., by_chain = FALSE)
```

Arguments

`x`	An nlist object.
`fun`	A function that given a numeric vector returns a numeric scalar.
`...`	Additional arguments passed to fun.
`by_chain`	A flag specifying whether to aggregate by chains.

Value

An nlist object if `by_chain = FALSE` otherwise an nlists object.

Examples

```
aggregate(nlists(nlist(x = 1:3), nlist(x = 2:4)))
```

---

as.mcmc.list.nlist *As mcmc.list Object*

Description

Coerces an nlist object to a `coda::mcmc.list` object.

Usage

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

Arguments

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

Value

An mcmc.list object.

See Also

`nlist-object()` and `coda::mcmc()`

Examples

```
coda::as.mcmc.list(nlist(x = matrix(1:6, 2)))
```

---

as.mcmc.list.nlists *As mcmc Object*

Description

Coerces an nlists object to a `coda::mcmc` object.

Usage

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

Arguments

x	An object.
...	Unused.

Value

An mcmc object.

See Also

nlists-object() and coda::mcmc()

Examples

```
coda::as.mcmc.list(nlists(
  nlist(x = matrix(1:6, 2)),
  nlist(x = matrix(3:8, 2))
))
```

---

as.mcmc.nlist *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 'nlist'
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.

Examples

```
as.mcmc(nlist(x = matrix(1:6, 2)))
```

`as.mcmc.nlists` *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 'nlists'
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.

Examples

```
as.mcmc(nlists(
  nlist(x = matrix(1:6, 2)),
  nlist(x = matrix(3:8, 2))
))
```

`as_nlist` *Coerce to nlist*

Description

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

Usage

```
as_nlist(x, ...)

as.nlist(x, ...)

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

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

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

as.nlists(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

Examples

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

as_nlists *Coerce to nlists*

Description

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

Usage

```
as_nlists(x, ...)

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

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

Arguments

x	An object.
...	Unused.

Value

An nlists object.

Methods (by class)

- `list`: Coerce 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))))
```

as_term.nlist *Coerce to a Term Vector*

Description

Coerce to a Term Vector

Usage

```
## S3 method for class 'nlist'
as_term(x, ...)
```

Arguments

x	An object.
...	Unused.

Examples

```
as_term(nlist(x = matrix(1:4, ncol = 2)))
```

---

as_term.nlists *Coerce to a Term Vector*

Description

Coerce to a Term Vector

Usage

```
## S3 method for class 'nlists'
as_term(x, ...)
```

Arguments

x	An object.
...	Unused.

Examples

```
as_term(nlists(nlist(x = matrix(1:4, ncol = 2))))
```

---

as_term_frame *Coerce to a Term Frame*

Description

A term frame is a tibble with the first column a term vector called and a numeric column called value and in the case of an nlists object an integer vector called samples. It includes the original nlist or nlists object.

Usage

```
as_term_frame(x, ...)
```

Arguments

x	An object.
...	Unused.

Value

An term_frame object.

as_term_frame.nlist *Coerce nlist Object to Data Frame*

Description

Coerces an nlist object to a data.frame with an term column and a value column.

Usage

```
## S3 method for class 'nlist'
as_term_frame(x, ...)
```

Arguments

x	An nlist object.
...	Unused.

Value

A data.frame.

Examples

```
as_term_frame(nlist(x = 1, y = 4:6))
```

as_term_frame.nlists *Coerce nlists Object to Data Frame*

Description

Coerces an nlists object to a data.frame with a term, sample and value column.

Usage

```
## S3 method for class 'nlists'
as_term_frame(x, ...)
```

Arguments

x	An nlists object.
...	Unused.

Value

A data.frame.

Examples

```
as_term_frame(nlists(
  nlist(x = 1, y = 4:6),
  nlist(x = 3, y = 1:3)
))
```

chk_nlist *Check nlist Object or nlists Object*

Description

chk_nlist checks if an nlist-object().

Usage

```
chk_nlist(x, x_name = NULL)

chk_nlists(x, x_name = NULL)
```

Arguments

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

Value

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

Functions

- chk_nlists: Check nlists Object
  chk_nlists checks if an nlists-object().

Examples

```
# chk_nlist
chk_nlist(nlist(x = 1))
try(chk_nlist(list(x = 1)))

# chk_nlists
chk_nlists(nlists(nlist(x = 1)))
```

collapse_chains.nlist *Collapse Chains*

Description

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

Usage

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

Arguments

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

Details

As nlist objects can only have 1 chain the object is unchanged.

Value

The modified object with one chain.

See Also

Other MCMC manipulations: `bind_chains()`, `estimates()`, `split_chains()`

Examples

```
collapse_chains(nlist(x = 2))
```

---

collapse_chains.nlists

Collapse Chains

Description

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

Usage

```
## S3 method for class 'nlists'
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()`, `estimates()`, `split_chains()`

Examples

```
collapse_chains(nlist(x = 2))
```

estimates.nlist *Estimates*

Description

Calculates the estimates for an MCMC object.

Usage

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

Arguments

x	An object.
fun	A function that given a numeric vector returns a numeric scalar.
...	Additional arguments passed to fun.

Value

A list of uniquely named numeric objects.

See Also

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

Examples

```
estimates(nlist(x = 1:9))
estimates(nlist(y = 3:5, zz = matrix(1:9, 3)))
```

estimates.nlists *Estimates*

Description

Calculates the estimates for an MCMC object.

Usage

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

Arguments

x	An object.
fun	A function that given a numeric vector returns a numeric scalar.
...	Additional arguments passed to fun.

Value

A list of uniquely named numeric objects.

See Also

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

Examples

```
estimates(nlists(nlist(x = 1:3), nlist(x = 2:4)), fun = mean)
```

`fill_all.nlist` *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 'nlist'
fill_all(x, value = 0L, 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

```
fill_all(nlist(x = c(2, NA), y = matrix(c(1:3, NA), nrow = 2)))
fill_all(nlist(x = c(2, NA), y = matrix(c(1:3, NA), nrow = 2)), nas = FALSE)
```

`fill_all.nlists` *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 'nlists'
fill_all(x, value = 0L, 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

```
fill_all(nlists(nlist(x = c(2, NA)), nlist(x = c(NA_real_, NA))))
fill_all(nlists(nlist(x = c(2, NA)), nlist(x = c(NA_real_, NA))), nas = FALSE)
```

`fill_na.nlist` *Fill Missing Values*

Description

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

Usage

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

Arguments

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

Value

The modified object.

Methods (by class)

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

See Also

Other fill: `fill_all()`

Examples

```
fill_na(nlist(x = c(2, NA), y = matrix(c(1:3, NA), nrow = 2)))
fill_na(nlists(nlist(x = c(2, NA)), nlist(x = c(NA_real_, NA))))
```

fill_na.nlists *Fill Missing Values*

Description

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

Usage

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

Arguments

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

Value

The modified object.

Methods (by class)

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

See Also

Other fill: fill_all()

Examples

```
fill_na(nlist(x = c(2, NA), y = matrix(c(1:3, NA), nrow = 2)))
```

is_numeric *Is numeric, nlist or nlists*

Description

Test whether x is a numeric object, nlist_object() or nlists_object().

Usage

```
is_numeric(x)

is_nlist(x)

is_nlists(x)
```

Arguments

x	An object.

Value

A flag indicating whether x is a numeric object or inherits from S3 class nlist or nlists.

Functions

- `is_nlist`: Is nlist
- `is_nlists`: Is nlists

Examples

```
# is_numeric
is_numeric(list(x = 1))
is_numeric(1)

# is_nlist
is_nlist(1)
is_nlist(list(x = 1))
is_nlist(nlist(x = 1))

# is_nlists
is_nlists(nlist(x = 1))
is_nlists(nlists(nlist(x = 2), nlist(x = 3.5)))
```

`nchains.nlist` *Number of Terms*

Description

Gets the number of terms of an MCMC object.

Usage

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

Arguments

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

Details

Always 1L.

Value

A integer scalar of the number of terms.

See Also

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

Examples

```
nchains(nlist(x = 1:2))
```

`nchains.nlists` *Number of Terms*

Description

Gets the number of terms of an MCMC object.

Usage

```
## S3 method for class 'nlists'
nchains(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()

Examples

```
nchains(nlists(nlist(x = c(2, 9)), nlist(x = c(1, 7))))
nchains(split_chains(nlists(nlist(x = c(2, 9)), nlist(x = c(1, 7)))))
```

`niters.nlist` *Number of Iterations*

Description

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

Usage

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

Arguments

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

Details

Always 1.

Value

An integer scalar of the number of iterations.

See Also

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

Examples

```
niters(nlist(x = 1:2))
```

---

niters.nlists *Number of Iterations*

Description

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

Usage

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

Examples

```
niters(nlists(nlist(x = c(2, 9)), nlist(x = c(1, 7))))
```

nlist *Create nlist Object*

Description

Creates a `nlist_object()` from one of more uniquely named numeric arguments.

Usage

```
nlist(...)
```

Arguments

`...`	Uniquely named numeric objects.

Details

An nlist object is an S3 class list of uniquely named numeric elements.

nlist objects are the raw data inputs for analytic engines such as JAGS, STAN and TMB.

Value

An nlist object.

Examples

```
nlist()
nlist(x = 1)
nlist(y = 1:4, zz = matrix(1:9, 3))
```

nlists *Create nlists Object*

Description

Creates an `nlists_object()` from one of more `nlist_object()`s.

Usage

```
nlists(...)
```

Arguments

`...`	nlist objects.

Details

An nlists object is a S3 class list of `nlist_object()` elements with the same names, dimensionalities and typeofs.

nlists objects are useful for storing individual realizations of a simulated data set.

Value

An nlists object.

Examples

```
nlists()
nlists(nlist())
nlists(nlist(x = 1))
nlists(nlist(x = 1), nlist(x = -3))
```

`npdims.nlist` *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 'nlist'
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()`

Examples

```
npdims(nlist(x = 1:3))
npdims(nlist(y = 3, zz = matrix(2:5, 2)))
```

npdims.nlists *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 'nlists'
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()`

Examples

```
npdims(nlists(nlist(x = 1:3)))
npdims(nlists(
  nlist(y = 3, zz = matrix(2:5, 2)),
  nlist(y = 5, zz = matrix(1:4, 2))
))
```

nsims.nlist *Number of Simulations*

Description

Gets the number of simulations (iterations * chains) of an MCMC object.

The default methods returns the product of `nchains()` and `niters()`.

Usage

```
## S3 method for class 'nlist'
nsims(x, ...)
```

Arguments

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

Details

Always 1L.

Value

An integer scalar of the number of simulations.

See Also

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

Examples

```
nsims(nlist(x = 1:2))
```

---

nsims.nlists *Number of Simulations*

Description

Gets the number of simulations (iterations * chains) of an MCMC object.

The default methods returns the product of nchains() and niters().

Usage

```
## S3 method for class 'nlists'
nsims(x, ...)
```

Arguments

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

Value

An integer scalar of the number of simulations.

See Also

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

Examples

```
nsims(nlists(nlist(x = c(2, 9)), nlist(x = c(1, 7))))
nsims(split_chains(nlists(nlist(x = c(2, 9)), nlist(x = c(1, 7)))))
```

nterms.nlist *Number of Terms*

Description

Gets the number of terms of an MCMC object.

Usage

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

Examples

```
nterms(nlist(x = 2))
nterms(nlist(x = NA_real_))
nterms(nlist(x = 3, zz = matrix(2:5, 2)))
```

nterms.nlists *Number of Terms*

Description

Gets the number of terms of an MCMC object.

Usage

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

Examples

```
nterms(nlists(nlist(x = 1:3)))
nterms(nlists(
  nlist(y = 3, zz = matrix(2:5, 2)),
  nlist(y = 5, zz = matrix(1:4, 2))
))
```

---

pars.nlist *Parameter Names*

Description

Gets the parameter names.

Usage

```
## S3 method for class 'nlist'
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 flag specifying whether to return the parameter name for each term element.
...	Other arguments passed to methods.

Value

A character vector of the names of the parameters.

See Also

Other parameters: npars(), set_pars()

Examples

```
pars(nlist(zz = 1, y = 3:6))
```

pars.nlists *Parameter Names*

Description

Gets the parameter names.

Usage

```
## S3 method for class 'nlists'
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 flag specifying whether to return the parameter name for each term element.
...	Other arguments passed to methods.

Value

A character vector of the names of the parameters.

See Also

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

Examples

```
pars(nlists(nlist(zz = 1, y = 3:6), nlist(zz = 4, y = 13:16)))
```

pdims.nlist *Parameter Dimensions*

Description

Gets the dimensions of each parameter of an object.

Usage

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

Examples

```
pdims(nlist(x = 1:3))
pdims(nlist(y = 3, zz = matrix(2:5, 2)))
```

`pdims.nlists` *Parameter Dimensions*

Description

Gets the dimensions of each parameter of an object.

Usage

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

Examples

```
pdims(nlists(nlist(x = 1:3)))
pdims(nlists(
  nlist(y = 3, zz = matrix(2:5, 2)),
  nlist(y = 5, zz = matrix(1:4, 2))
))
```

relist_nlist *Relists an unlist nlist Object*

Description

Relists an nlist object that has been unlisted to a named numeric vector. Ensures absent terms are included and preserves integer class.

Usage

```
relist_nlist(flesh, skeleton)
```

Arguments

flesh	An atomic vector
skeleton	An nlist object.

Value

A numeric vector of the values in x.

See Also

as_nlist.numeric() and unlist_nlist()

Examples

```
relist_nlist(c(`a[2]` = 5), nlist(a = 1:3))
```

set_pars.nlist *Set Parameter Names*

Description

Sets an object's parameter names.

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

Usage

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

Examples

```
set_pars.foobar <- function(x, ...) {
  NotYetImplemented()
  # replace with code to set_pars for an object of class 'foobar'
}
```

---

set_pars.nlists *Set Parameter Names*

Description

Sets an object's parameter names.

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

Usage

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

Examples

```
set_pars.foobar <- function(x, ...) {
  NotYetImplemented()
  # replace with code to set_pars for an object of class 'foobar'
}
```

`split_chains.nlists` *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 'nlists'
split_chains(x, ...)
```

Arguments

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

Value

The modified object.

See Also

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

Examples

```
split_chains.foobar <- function(x, ...) {
  NotYetImplemented()
  # replace with code to split_chains for an object of class 'foobar'
}
```

subset.nlist *Subset nlist Object*

Description

Subsets an nlist object by its parameters.

Usage

```
## S3 method for class 'nlist'
subset(x, pars = NULL, ...)
```

Arguments

x	An nlist object.
pars	A character vector of parameter names.
...	Unused.

Details

It can also be used to reorder the parameters.

Value

An nlist object.

Examples

```
nlist <- nlist(a = 1, y = 3, x = 1:4)
subset(nlist)
subset(nlist, "a")
subset(nlist, c("x", "a"))
```

subset.nlists *Subset nlists Object*

Description

Subsets an nlists object by its parameters, chains and iterations.

Usage

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

Arguments

x	An nlists object.
chains	An integer vector of chains.
iters	An integer vector of iterations.
pars	A character vector of parameter names.
...	Unused.

Details

It can also be used to reorder the parameters as well as duplicate chains and iterations.

Value

An nlists object.

Examples

```
nlists <- nlists(
  nlist(a = 1, y = 3, x = 1:4),
  nlist(a = 2, y = 4, x = 4:1),
  nlist(a = 3, y = 6, x = 5:2)
)
subset(nlists)
subset(nlists, pars = "a")
subset(nlists, pars = c("x", "a"))
subset(nlists, iters = 1L)
subset(nlists, iters = c(2L, 2L))
```

---

thin.default *Thin MCMC Object*

Description

Thins an MCMC object's iterations.

Usage

```
## Default S3 method:
thin(x, nthin = 1L, ...)
```

Arguments

x	An object.
nthin	A positive integer of the thinning rate.
...	Unused.

Value

The thinned MCMC object.

Examples

```
thin(nlists(nlist(x = 1), nlist(x = 2), nlist(x = 3), nlist(x = 4)), nthin = 2)
```

tidy.nlists *Turn an object into a tidy tibble*

Description

Turn an object into a tidy tibble

Usage

```
## S3 method for class 'nlists'
tidy(x, ...)
```

Arguments

x	An object to be converted into a tidy `tibble::tibble()`.
...	Additional arguments to tidying method.

Value

A `tibble::tibble()` with information about model components.

Methods

No methods found in currently loaded packages.

Examples

```
tidy(nlists(
  nlist(x = 1, y = 4:6),
  nlist(x = 3, y = 7:9)
))
```

unlist.nlist *Flatten nlist Object*

Description

Flatten nlist Object

Usage

```
## S3 method for class 'nlist'
unlist(x, recursive = TRUE, use.names = TRUE)
```

Arguments

x	An nlist object.
recursive	Ignored.
use.names	A flag specifying whether to preserve names.

Value

A named numeric vector of the values in x.

See Also

unlist_nlist()

Examples

```
unlist(nlist(y = 2, x = matrix(4:7, ncol = 2)))
```

---

unlist_nlist *Flatten nlist Object*

Description

Simplifies an nlist object to an named numeric vector where the names are the terms.

Usage

```
unlist_nlist(x)
```

Arguments

x	An nlist object.

Value

A named numeric vector of the values in x.

See Also

as_nlist.numeric() and relist_nlist()

Examples

```
unlist_nlist(nlist(y = 2, x = matrix(4:7, ncol = 2)))
```

vld_nlist *Validate nlist Object or nlists Object*

Description

Validate nlist Object or nlists Object

Usage

```
vld_nlist(x)

vld_nlists(x)
```

Arguments

x	The object to check.

Value

A flag indicating whether the object was validated.

Functions

- vld_nlists: Validate nlists Object

Examples

```
# vld_nlist
vld_nlist(nlist(x = 1))
try(vld_nlist(list(x = 1)))

# vld_nlists
vld_nlists(nlists(nlist(x = 1)))
vld_nlists(1)
```

Index