Package ‘nlist’

September 1, 2021

**Title**  Lists of Numeric Atomic Objects

**Version**  0.3.3

**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**  abind,
chk,
coda,
extras,
generics,
lifecycle,
purrr,
stats,
term,
tibble,
universals

**Suggests**  covr,
rlang,
testthat

**RdMacros**  lifecycle

**Config/testthat/edition**  3

**Encoding**  UTF-8

**Language**  en-US

**Roxygen**  list(markdown = TRUE)

**RoxygenNote**  7.1.1
**R topics documented:**

<table>
<thead>
<tr>
<th>R topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggregate.nlist</td>
<td>3</td>
</tr>
<tr>
<td>aggregate.nlists</td>
<td>4</td>
</tr>
<tr>
<td>as_mcmc</td>
<td>4</td>
</tr>
<tr>
<td>as_mcmc_list</td>
<td>5</td>
</tr>
<tr>
<td>as_nlist</td>
<td>6</td>
</tr>
<tr>
<td>as_nlists</td>
<td>8</td>
</tr>
<tr>
<td>as_term.mcmc</td>
<td>9</td>
</tr>
<tr>
<td>as_term.nlist</td>
<td>9</td>
</tr>
<tr>
<td>as_term.nlists</td>
<td>10</td>
</tr>
<tr>
<td>as_term_frame</td>
<td>10</td>
</tr>
<tr>
<td>as_term_frame.nlist</td>
<td>11</td>
</tr>
<tr>
<td>as_term_frame.nlists</td>
<td>11</td>
</tr>
<tr>
<td>bind_iterations.mcmc</td>
<td>12</td>
</tr>
<tr>
<td>bind_iterations.mcmc.list</td>
<td>13</td>
</tr>
<tr>
<td>chk_nlist</td>
<td>13</td>
</tr>
<tr>
<td>collapse Chains.mcmc</td>
<td>14</td>
</tr>
<tr>
<td>collapse Chains.mcmc.list</td>
<td>15</td>
</tr>
<tr>
<td>collapse Chains.nlist</td>
<td>15</td>
</tr>
<tr>
<td>collapse Chains.nlists</td>
<td>16</td>
</tr>
<tr>
<td>complete_terms.mcmc</td>
<td>17</td>
</tr>
<tr>
<td>estimates.nlist</td>
<td>17</td>
</tr>
<tr>
<td>estimates.nlists</td>
<td>18</td>
</tr>
<tr>
<td>fill_all.nlist</td>
<td>19</td>
</tr>
<tr>
<td>fill_all.nlists</td>
<td>20</td>
</tr>
<tr>
<td>fill_na.nlist</td>
<td>21</td>
</tr>
<tr>
<td>fill_na.nlists</td>
<td>22</td>
</tr>
<tr>
<td>is_numeric</td>
<td>23</td>
</tr>
<tr>
<td>nchains.mcmc</td>
<td>24</td>
</tr>
<tr>
<td>nchains.mcmc.list</td>
<td>24</td>
</tr>
<tr>
<td>nchains.nlist</td>
<td>25</td>
</tr>
<tr>
<td>nchains.nlists</td>
<td>25</td>
</tr>
<tr>
<td>nterms.mcmc</td>
<td>26</td>
</tr>
<tr>
<td>nterms.mcmc.list</td>
<td>27</td>
</tr>
<tr>
<td>nterms.nlist</td>
<td>27</td>
</tr>
<tr>
<td>nterms.nlists</td>
<td>28</td>
</tr>
<tr>
<td>nlist</td>
<td>28</td>
</tr>
<tr>
<td>nlists</td>
<td>29</td>
</tr>
<tr>
<td>npdims.mcmc.list</td>
<td>30</td>
</tr>
<tr>
<td>npdims.nlist</td>
<td>30</td>
</tr>
<tr>
<td>npdims.nlists</td>
<td>31</td>
</tr>
<tr>
<td>nsims.nlist</td>
<td>32</td>
</tr>
<tr>
<td>nsims.nlists</td>
<td>32</td>
</tr>
<tr>
<td>nterms.mcmc</td>
<td>33</td>
</tr>
<tr>
<td>nterms.mcmc.list</td>
<td>34</td>
</tr>
<tr>
<td>nterms.nlist</td>
<td>34</td>
</tr>
<tr>
<td>nterms.nlists</td>
<td>35</td>
</tr>
<tr>
<td>pars.mcmc</td>
<td>35</td>
</tr>
<tr>
<td>pars.mcmc.list</td>
<td>36</td>
</tr>
<tr>
<td>pars.nlist</td>
<td>37</td>
</tr>
<tr>
<td>pars.nlists</td>
<td>37</td>
</tr>
</tbody>
</table>
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

See Also

Other aggregate: aggregate.nlists()
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 object.

fun  A function that given a numeric vector returns a numeric scalar.

...  Unused.

by_chain  A flag specifying whether to aggregate by chains.

Value

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

See Also

Other aggregate: aggregate.nlist()

Examples

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

Coerce to mcmc Object

Description

Coerce an R object to an mcmc object.
as_mcmc_list

Usage

as_mcmc(x, ...)

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

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

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

Arguments

x    An object.
...	Unused.

Value

An mcmc object.

Methods (by class)

• mcmc.list: Coerce an mcmc.list object to an mcmc object.
• nlist: Coerce an nlist object to an mcmc object.
• nlists: Coerce an nlists object to an mcmc object.

See Also

coda::as.mcmc()

Other mcmc: as_mcmc_list()

Examples

as_mcmc(as_mcmc_list(nlists(nlist(x = 2), nlist(x = 3))))
as_mcmc(nlist(x = matrix(1:6, 2)))
as_mcmc(nlists(
  nlist(x = matrix(1:6, 2)),
  nlist(x = matrix(3:8, 2))
))

---

as_mcmc_list Coerce to an mcmc.list Object

Description

Coerce an R object to an mcmc.list object.
Usage

```r
as_mcmc_list(x, ...)
## S3 method for class 'mcmc'
as_mcmc_list(x, ...)
## S3 method for class 'nlist'
as_mcmc_list(x, ...)
## S3 method for class 'nlists'
as_mcmc_list(x, ...)
```

Arguments

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

Value

An mcmc.list object.

Methods (by class)

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

See Also

Other mcmc: `as_mcmc()`

Examples

```r
as_mcmc_list(nlist(x = matrix(1:6, 2)))
as_mcmc_list(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()`.
as_nlist

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, ...)

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

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

as.nlists(x, ...)

Arguments

x An object.
...

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  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 'mcmc'
as_nlists(x, ...)

## S3 method for class 'mcmc.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
- 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))))
as_term.mcmc

Description
Coerce to a Term Vector

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

Arguments
- `x`: An object.
- `...`: Unused.

See Also
Other coerce term: `as_term.nlists()`, `as_term.nlist()`, `as_term.frame.nlists()`, `as_term.frame.nlist()`, `as_term_frame()`

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

as_term.nlist

Description
Coerce to a Term Vector

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

Arguments
- `x`: An object.
- `...`: Unused.

See Also
Other coerce term: `as_term.mcmc()`, `as_term.nlists()`, `as_term.frame.nlists()`, `as_term.frame.nlist()`, `as_term_frame()`

Examples
```r
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.

See Also
Other coerce term: as_term.mcmc(), as_term.nlist(), as_term_frame.nlists(), as_term_frame.nlist(), as_term_frame()

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.

See Also
Other coerce term: as_term.mcmc(), as_term.nlist(), as_term.nlist(), as_term_frame.nlists(), as_term_frame.nlist()
**Coerce nlist Object to Data Frame**

**Description**

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

**Usage**

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

**Arguments**

- `x`: An nlist object.
- `...`: Unused.

**Value**

A data.frame.

**See Also**

Other coerce term: `as_term.mcmc()`, `as_term.nlists()`, `as_term.nlist()`, `as_term_frame.nlists()`, `as_term_frame()`

**Examples**

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

---

**Coerce nlists Object to Data Frame**

**Description**

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

**Usage**

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

**Arguments**

- `x`: An nlists object.
- `...`: Unused.

**Value**

A data.frame.
bind_iterations.mcmc

See Also
Other coerce term: as_term.mcmc(), as_term.nlists(), as_term.nlist(), as_term_frame.nlist(), as_term_frame()

Examples

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

bind_iterations.mcmc  Bind Iterations

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

Usage

## S3 method for class 'mcmc'
bind_iterations(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_chains(), collapse_chains(), estimates(), split_chains()

Examples

bind_iterations(as_mcmc(nlist(x = 1)), as_mcmc(nlist(x = 3)))
bind_iterations.mcmc.list

Bind Iterations

Description

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

Usage

## S3 method for class 'mcmc.list'
bind_iterations(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_chains(), collapse_chains(), estimates(), split_chains()

Examples

bind_iterations(as_mcmc_list(nlist(x = 1)), as_mcmc_list(nlist(x = 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.mcmc  Collapse Chains

Description

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

Usage

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

Arguments

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

Details

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

Value

The modified object with one chain.

See Also

Other collapse: collapse_chains.nlists(), collapse_chains.nlist()

Examples

collapse_chains(as_mcmc(nlist(x = 2)))
collapse_chains.mcmc.list

*Collapse Chains*

**Description**

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

**Usage**

```r
## S3 method for class 'mcmc.list'
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()`

---

collapse_chains.nlist

*Collapse Chains*

**Description**

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

**Usage**

```r
## 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.
collapse_chains.nlists

**Collapse Chains**

**Description**

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

**Usage**

```r
## 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 collapse: `collapse_chains.mcmc()`, `collapse_chains.nlist()`

**Examples**

```r
collapse_chains(nlist(x = 2))
```
Complete Terms

Description

Adds any absent elements to an mcmc object.

Usage

## S3 method for class 'mcmc'
complete_terms(x, silent = FALSE, ...)

Arguments

x An mcmc object.
silent A flag specifying whether to suppress warning messages.
... Unused.

Details

The terms are repaired before being completed. Missing or invalid or inconsistent terms are dropped with a warning.

Value

The repaired and complete mcmc object.

Examples

mcmc <- as_mcmc(nlist(beta = matrix(1:4, nrow = 2)))
mcmc <- mcmc[, -4, drop = FALSE]
complete_terms(mcmc)

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.
estimates.nlists

Value

A list of uniquely named numeric objects.

See Also

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

```r
## 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.

### 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

```r
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)
```
Description
Fills all of an object’s (missing and non-missing) values while preserving the object’s dimensionality and class.

Usage

```r
## 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.

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

```r
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)
```
**Description**
Fills all of an object’s missing values while preserving the object’s dimensionality and class.

**Usage**
```r
## 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.

**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**
```r
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 all of an object’s missing values while preserving the object’s dimensionality and class.

Usage

```r
### 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.

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

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

Description

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

Usage

```r
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

```r
# 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.mcmc  Number of Chains

Description
Gets the number of chains of an MCMC object.

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

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

Value
An integer scalar of the number of chains.

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

nchains.mcmc.list  Number of Chains

Description
Gets the number of chains of an MCMC object.

Usage
## S3 method for class 'mcmc.list'
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()
### Description

Gets the number of terms of an MCMC object.

### Usage

```r
## 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

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

---

### Description

Gets the number of terms of an MCMC object.

### Usage

```r
## 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(), niter(), 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)))))

Description

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

Usage

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

Arguments

x  An object.

...  Other arguments passed to methods.

Value

An integer scalar of the number of iterations.

See Also

Other MCMC dimensions: nchains(), npars(), nsams(), nsims(), nterms()
## S3 method for class `mcmc.list`
niters(x, ...)

### Arguments

- **x**: An object.
- **...**: Other arguments passed to methods.

### Value

An integer scalar of the number of iterations.

### Description

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

### Usage

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

### See Also

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

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

### Arguments

- **x**: An object.
- **...**: Other arguments passed to methods.

### Value

An integer scalar of the number of iterations.

### Description

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

### Usage

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

### See Also

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

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

Examples

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

---

nlist.nlists  
*Number of Iterations*

Description

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

Usage

```r
## 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

```r
 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

```r
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.

See Also
nlists()

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.

See Also
nlist()

Examples
nlists()
nlists(nlist())
nlists(nlist(x = 1))
nlists(nlist(x = 1), nlist(x = -3))
n(pdims.mcmc.list) \hspace{1cm} \textit{Number of Parameter Dimensions}

\textbf{Description}

Gets the number of the dimensions of each parameter of an object. The default methods returns the length of each element of \texttt{pdims()} as an integer vector.

\textbf{Usage}

\begin{verbatim}
## S3 method for class 'mcmc.list'
npdims(x, ...)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{x} \hspace{1cm} An object.
  \item \texttt{...} \hspace{1cm} Other arguments passed to methods.
\end{itemize}

\textbf{Value}

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

\textbf{See Also}

Other dimensions: \texttt{dims()}, \texttt{ndims()}, \texttt{pdims()}

\hline

n(pdims.nlist) \hspace{1cm} \textit{Number of Parameter Dimensions}

\textbf{Description}

Gets the number of the dimensions of each parameter of an object. The default methods returns the length of each element of \texttt{pdims()} as an integer vector.

\textbf{Usage}

\begin{verbatim}
## S3 method for class 'nlist'
npdims(x, ...)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{x} \hspace{1cm} An object.
  \item \texttt{...} \hspace{1cm} Other arguments passed to methods.
\end{itemize}

\textbf{Value}

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

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

Examples

```r
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

```r
## 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

```r
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**

```r
## 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**

```r
nsims(nlist(x = 1:2))
```
## nterms.mcmc

**Arguments**

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

**Value**

An integer scalar of the number of simulations.

**See Also**

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

**Examples**

```r
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)))))
```

---

### Description

Gets the number of terms of an MCMC object.

### Usage

```r
## S3 method for class 'mcmc'
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()`, `niter()`, `npars()`, `nsams()`, `nsims()`
nterms.mcmc.list  Number of Terms

Description

Gets the number of terms of an MCMC object.

Usage

```r
## S3 method for class 'mcmc.list'
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()`, `niter()`, `npars()`, `nsams()`, `nsims()`

nterms.nlist  Number of Terms

Description

Gets the number of terms of an MCMC object.

Usage

```r
## 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()`, `niter()`, `npars()`, `nsams()`, `nsims()`
Examples

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

<table>
<thead>
<tr>
<th>nterms.nlists</th>
<th>Number of Terms</th>
</tr>
</thead>
</table>

Description

Gets the number of terms of an MCMC object.

Usage

```r
## 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()`, `niter()`, `npar()`, `nsample()`, `nsim()`

Examples

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

<table>
<thead>
<tr>
<th>pars.mcmc</th>
<th>Parameter Names</th>
</tr>
</thead>
</table>

Description

Gets the parameter names.

Usage

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

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()

### pars.mcmc.list  Parameter Names

Description

Gets the parameter names.

Usage

```r
## S3 method for class 'mcmc.list'
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()
pars.nlist

Parameter Names

### Description

Gets the parameter names.

### Usage

```r
## 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

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

---

pars.nlists

Parameter Names

### Description

Gets the parameter names.

### Usage

```r
## 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.
### pdims.mcmc

**Parameter Dimensions**

**Description**

Gets the dimensions of each parameter of an object.

**Usage**

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

**Arguments**

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

**Value**

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

**See Also**

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

---

### pdims.mcmc.list

**Parameter Dimensions**

**Description**

Gets the dimensions of each parameter of an object.

**Usage**

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

**Value**

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

**See Also**

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

---

**Value**

A character vector of the names of the parameters.

**See Also**

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

**Examples**

```r
dpars(nlists(nlist(zz = 1, y = 3:6), nlist(zz = 4, y = 13:16)))
```
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: \texttt{dims()}, \texttt{ndims()}, \texttt{npdims()}

Examples

\begin{verbatim}
pdims(nlist(x = 1:3))
pdims(nlist(y = 3, zz = matrix(2:5, 2)))
\end{verbatim}
**pdims.nlists**  
*Parameter Dimensions*

**Description**

Gets the dimensions of each parameter of an object.

**Usage**

```r
## 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**

```r
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**

```r
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

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

Description

Sets an object’s parameter names.

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

Usage

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

Arguments

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

Details

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

Value

The modified object.

See Also

Other parameters: `npars()`, `pars()`
set_pars.mcmc.list  Set Parameters

Description
Sets an object’s parameter names.
The assignment version pars<-() forwards to set_pars().

Usage
## S3 method for class 'mcmc.list'
set_pars(x, value, ...)

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

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

Value
The modified object.

See Also
Other parameters: npars(), pars()

set_pars.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

nlist <- nlist(x = 1, y = 3:4)
pars(nlist) <- c("a", "b")
nlist
set_pars(nlist, c("z", "c!"))

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

nlists <- nlists(nlist(x = 2), nlist(x = 3))
pars(nlists) <- "a"
nlists
set_pars(nlists, "zz")

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(), bind_iterations(), collapse_chains(), estimates()

Examples

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

Description

Subsets an mcmc object by its parameters and/or iterations.

Usage

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

subset.mcmc.list

Arguments

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

Details

Future versions should allow it to be reordered by its parameters.

Value

An mcmc object.

Examples

mcmc <- as_mcmc(nlist(beta = 1:2, theta = 1))
subset(mcmc, pars = "beta")
subset(mcmc, iters = c(1L,1L))

subset.mcmc.list  Subset mcmc.list Object

Description

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

Usage

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

Arguments

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

Future versions should allow it to be reordered by its parameters.

Value

An mcmc.list object.

Examples

```r
mcmc.list <- as_mcmc_list(nlists(nlist(beta = 1:2, theta = 1),
                                 nlist(beta = 3:4, theta = -1)))
subset(mcmc.list, pars = "beta")
subset(mcmc.list, iters = c(1L,1L))
```

### subset.nlist

#### Subset nlist Object

Description

Subsets an nlist object by its parameters.

Usage

```r
## 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

```r
nlist <- nlist(a = 1, y = 3, x = 1:4)
subset(nlist)
subset(nlist, "a")
subset(nlist, c("x", "a"))
```
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))
**tidy.mcmc**

**thin.default**

*Thin MCMC Object*

**Description**

Thins an MCMC object’s iterations.

**Usage**

```r
## 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**

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

**tidy.mcmc**

*Turn an object into a tidy tibble*

**Description**

Turn an object into a tidy tibble

**Usage**

```r
## S3 method for class 'mcmc'
tidy(x, simplify = FALSE, ...)
```

**Arguments**

- `x`: An object.
- `simplify`: A flag specifying whether to drop sd and zscore columns.
- `...`: Unused.

**Value**

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

**Methods**

No methods found in currently loaded packages.
**tidy.mcmc.list**  
*Turn an object into a tidy tibble*

**Description**  
Turn an object into a tidy tibble

**Usage**  
```r  
## S3 method for class 'mcmc.list'  
tidy(x, simplify = FALSE, ...)  
```

**Arguments**  
- `x`: An object.  
- `simplify`: A flag specifying whether to drop sd and zscore columns.  
- `...`: Unused.

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

**Methods**  
No methods found in currently loaded packages.

---

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

**Description**  
Turn an object into a tidy tibble

**Usage**  
```r  
## S3 method for class 'nlists'  
tidy(x, simplify = FALSE, ...)  
```

**Arguments**  
- `x`: An object.  
- `simplify`: A flag specifying whether to drop sd and zscore columns.  
- `...`: Unused.

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

No methods found in currently loaded packages.

Examples

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

---

**unlist.nlist**

**Flatten nlist Object**

Description

Flatten nlist Object

Usage

```r
## 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

```r
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

* aggregate
  aggregate.nlist, 3
  aggregate.nlists, 4

* coerce term
  as_term.mcmc, 9
  as_term.nlist, 9
  as_term.nlists, 10
  as_term_frame, 10
  as_term_frame.nlist, 11
  as_term_frame.nlists, 11

* coerce
  as_nlist, 6
  as_nlists, 8

* collapse
  collapse_chains.mcmc, 14
  collapse_chains.nlist, 15
  collapse_chains.nlists, 16

* mcmc
  as_mcmc, 4
  as_mcmc_list, 5

aggregate.nlist, 3, 4
aggregate.nlists, 3, 4
as.nlist (as_nlist), 6
as.nlists (as_nlist), 6
as_mcmc, 4, 6
as_mcmc_list, 5, 5
as_nlist, 6, 8
as_nlist.numeric(), 41, 51
as_nlists, 7, 8
as_term.mcmc, 9, 9, 10–12
as_term.nlist, 9, 9, 10–12
as_term.nlists, 9, 10, 10, 11, 12
as_term_frame, 9, 10, 10, 11, 12
as_term_frame.nlist, 9, 10, 11, 12
as_term_frame.nlists, 9–11, 11

bind_chains, 12, 13, 15, 18, 44
bind_iterations, 15, 18, 44
bind_iterations.mcmc, 12
bind_iterations.mcmc.list, 13

chk_nlist, 13
chk_nlists (chk_nlist), 13

coda::as.mcmc(), 5
collapse_chains, 12, 13, 18, 44
collapse_chains.mcmc, 14, 16
collapse_chains.mcmc.list, 15
collapse_chains.nlist, 14, 15, 16
collapse_chains.nlists, 14, 16, 16

complete_terms.mcmc, 17
dims, 30, 31, 38–40

estimates, 12, 13, 15, 44

estimates.nlist, 17
estimates.nlists, 18

fill_all, 21, 22
fill_all.nlist, 19
fill_all.nlists, 20
fill_na, 19, 20
fill_na.nlist, 21
fill_na.nlists, 22

is_nlist (is_numeric), 23
is_nlists (is_numeric), 23
is_numeric, 23

nchains, 25–28, 32–35

nchains(), 32
nchains.mcmc, 24
nchains.mcmc.list, 24
nchains.nlist, 25
nchains.nlists, 25

ndims, 30, 31, 38–40

niters, 24–26, 32–35

niters(), 32

niters.mcmc, 26

niters.mcmc.list, 27

niters.nlist, 27

niters.nlists, 28

nlist, 28

nlist(), 29

nlist-object (nlist), 28

nlist_object (nlist), 28

nlist_object(), 3, 4, 6, 23, 28, 29

nlists, 29

nlists(), 29
nlists-object (nlists), 29
nlists_object (nlists), 29
nlists_object(), 4, 8, 23, 29
npars, 24–28, 32–38, 41–43
npdms, 38–40
npdms.mcmc.list, 30
npdms.nlist, 30
npdms.nlists, 31
nsams, 24–28, 32–35
nsims, 24–28, 33–35
nsims.nlist, 32
nsims.nlists, 32
nterms, 24, 26–28, 32, 33
nterms.mcmc, 33
nterms.mcmc.list, 34
nterms.nlist, 34
nterms.nlists, 35
pars, 41–43
pars.mcmc, 35
pars.mcmc.list, 36
pars.nlist, 37
pars.nlists, 37
pdims, 30, 31
pdims(), 30, 31
pdims.mcmc, 38
pdims.mcmc.list, 38
pdims.nlist, 39
pdims.nlists, 40
relist_nlist, 40
relist_nlist(), 51
set_pars, 36–38
set_pars.mcmc, 41
set_pars.mcmc.list, 42
set_pars.nlist, 42
set_pars.nlists, 43
split_chains, 12, 13, 15, 18
split_chains.nlists, 44
subset.mcmc, 44
subset.mcmc.list, 45
subset.nlist, 46
subset.nlists, 47
thin.default, 48
tibble::tibble(), 48, 49
tidy.mcmc, 48
tidy.mcmc.list, 49
tidy.nlists, 49
unlist.nlist, 50
unlist_nlist, 51
vld_nlist, 51
vld_nlists(vld_nlist), 51