Package ‘nlist’

January 24, 2020

Title Lists of Numeric Atomic Objects

Version 0.1.0

Description Create and manipulate numeric list (nlist) objects.

An nlist is an S3 list of uniquely named numeric atomic (natomic) objects.
An nonatomic 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

Depends R (>= 3.3)

Imports stats, chk, term (>= 0.1.0), coda, abind, purrr

Suggests covr, testthat

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

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

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Language en-US

NeedsCompilation no

Author Joe Thorley [aut, cre] (<https://orcid.org/0000-0002-7683-4592>),
Poisson Consulting [cph, fnd]

Maintainer Joe Thorley <joe@poissonconsulting.ca>

Repository CRAN

Date/Publication 2020-01-24 17:00:02 UTC
**R topics documented:**

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggregate.nlist</td>
<td>3</td>
</tr>
<tr>
<td>aggregate.nlists</td>
<td>3</td>
</tr>
<tr>
<td>as.mcmc.list.nlist</td>
<td>4</td>
</tr>
<tr>
<td>as.mcmc.list.nlists</td>
<td>5</td>
</tr>
<tr>
<td>as.mcmc.nlist</td>
<td>5</td>
</tr>
<tr>
<td>as.mcmc.nlists</td>
<td>6</td>
</tr>
<tr>
<td>as.nlist</td>
<td>7</td>
</tr>
<tr>
<td>as.nlists</td>
<td>8</td>
</tr>
<tr>
<td>chk_natomic</td>
<td>8</td>
</tr>
<tr>
<td>collapse_chains</td>
<td>10</td>
</tr>
<tr>
<td>fill_na</td>
<td>10</td>
</tr>
<tr>
<td>is_natomic</td>
<td>11</td>
</tr>
<tr>
<td>is.natomic-object</td>
<td>12</td>
</tr>
<tr>
<td>nchains</td>
<td>13</td>
</tr>
<tr>
<td>niter</td>
<td>14</td>
</tr>
<tr>
<td>nlist</td>
<td>15</td>
</tr>
<tr>
<td>nlist-object</td>
<td>15</td>
</tr>
<tr>
<td>nlists</td>
<td>16</td>
</tr>
<tr>
<td>nlists-object</td>
<td>16</td>
</tr>
<tr>
<td>npdims.nlist</td>
<td>17</td>
</tr>
<tr>
<td>npdims.nlists</td>
<td>17</td>
</tr>
<tr>
<td>nsams</td>
<td>18</td>
</tr>
<tr>
<td>nsims</td>
<td>19</td>
</tr>
<tr>
<td>nterms.nlist</td>
<td>20</td>
</tr>
<tr>
<td>nterms.nlists</td>
<td>20</td>
</tr>
<tr>
<td>numericise</td>
<td>21</td>
</tr>
<tr>
<td>pars.nlist</td>
<td>23</td>
</tr>
<tr>
<td>pars.nlists</td>
<td>24</td>
</tr>
<tr>
<td>pdims.nlist</td>
<td>24</td>
</tr>
<tr>
<td>pdims.nlists</td>
<td>25</td>
</tr>
<tr>
<td>relist_nlist</td>
<td>26</td>
</tr>
<tr>
<td>split_chains</td>
<td>26</td>
</tr>
<tr>
<td>subset.nlist</td>
<td>27</td>
</tr>
<tr>
<td>subset.nlists</td>
<td>28</td>
</tr>
<tr>
<td>unlist.nlist</td>
<td>29</td>
</tr>
<tr>
<td>unlist_nlist</td>
<td>29</td>
</tr>
<tr>
<td>vld_natomic</td>
<td>30</td>
</tr>
</tbody>
</table>

**Index** 32
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

```
as.mcmc.list.nlist(x, ...)
```

Arguments

- x: A nlist 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**

```r
as.mcmc.list.nlists(x, ...)
```

**Arguments**

- `x` A nlists object.
- `...` Unused.

**Value**

An mcmc object.

**See Also**

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

**Examples**

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

---

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

**Description**

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

**Usage**

```r
as.mcmc.nlist(x, ...)
```

**Arguments**

- `x` A nlist object.
- `...` Unused.
as.mcmc.nlists

Description

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

Usage

as.mcmc.nlists(x, ...)

Arguments

x A nlists object.
... Unused.

Value

An mcmc object.

See Also

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

e.xamples

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

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

Coerce an R object to an \texttt{nlist\_object()}. 

Usage

\begin{verbatim}
as.nlist(x, ...)
\end{verbatim}

## S3 method for class \texttt{numeric}
as.nlist(x, ...)

## S3 method for class \texttt{list}
as.nlist(x, ...)

## S3 method for class \texttt{data.frame}
as.nlist(x, ...)

Arguments

\begin{itemize}
  \item \texttt{x} An R object.
  \item \texttt{...} Unused
\end{itemize}

Value

An nlist object.

Methods (by class)

\begin{itemize}
  \item \texttt{numeric}: Coerce named numeric vector to nlist
  \item \texttt{list}: Coerce list to nlist
  \item \texttt{data.frame}: Coerce data.frame to nlist
\end{itemize}

Examples

\begin{verbatim}
as.nlist(list(x = 1:4))
as.nlist(c(`a[2]` = 3, `a[1]` = 2))
\end{verbatim}
### as.nlists

Coerce to nlists

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

**Usage**

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

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

**Arguments**

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

**Value**

An nlists object.

**Methods (by class)**

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

**Examples**

```r
as.nlists(list(nlist(x = c(1, 5)), nlist(x = c(2, 3)), nlist(x = c(3, 2))))
```

### chk_natomic

Check Numeric Atomic, nlist Object or nlists Object

**Description**

`chk_natomic` checks if numeric object using `is.numeric(x) && is.atomic(x)`.

`chk_natomic` checks if an `nlist-object()` while ignoring the class.

`chk_nlists` checks if an `nlists-object()` while ignoring the class.
\textit{chk\_natomic}

\textbf{Usage}

\begin{verbatim}
chk_natomic(x, x_name = NULL) 
chk_nlist(x, x_name = NULL) 
chk_nlists(x, x_name = NULL) 
\end{verbatim}

\textbf{Arguments}

\begin{verbatim}
x \hspace{1cm} \text{The object to check.} 
x\_name \hspace{1cm} \text{A string of the name of object x or NULL.} 
\end{verbatim}

\textbf{Value}

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

\textbf{Functions}

- \texttt{chk\_natomic}: Check Numeric Atomic
- \texttt{chk\_nlist}: Check nlist Object
- \texttt{chk\_nlists}: Check nlists Object

\textbf{See Also}

\texttt{vld\_natomic()}

\textbf{Examples}

\begin{verbatim}
# chk\_natomic 
chk\_natomic(1) 
chk\_natomic(matrix(1L)) 
try(chk\_natomic(TRUE)) 

# chk\_nlist 
chk\_nlist(nlist(x = 1)) 
try(chk\_nlist(list(x = 1))) 

# chk\_nlists 
chk\_nlists(nlists(nlist(x = 1))) 
\end{verbatim}
collapse_chains  

**Collapse Chains**

**Description**

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

**Usage**

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

**Arguments**

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

**Methods (by class)**

- `nlist`: Collapse chains for an nlist object
  - As nlist objects can only have 1 chain the object is unchanged.
- `nlists`: Collapse chains for an nlists object

---

fill_na  

**Fill Missing Values**

**Description**

Fills missing values.

**Usage**

```r
fill_na(x, ...)
## S3 method for class 'logical'
fill_na(x, value = FALSE, ...)
## S3 method for class 'integer'
fill_na(x, value = 0L, ...)
```
## S3 method for class 'numeric'
fill_na(x, value = 0, ...)

## S3 method for class 'character'
fill_na(x, value = "0", ...)

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

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

Arguments

x The object.
...
value A scalar of the value to replace missing values with.

Value

The object with missing values filled.

Methods (by class)

- logical: Fill missing values of logical object
- integer: Fill missing values of integer object
- numeric: Fill missing values of numeric object
- character: Fill missing values of character object
- nlist: Fill missing values of nlist object
- nlists: Fill missing values of nlist object

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))))

is_natomic | Is nonatomic, nlist or nlists

Description

Test whether x is a nonatomic_object(), nlist_object() or nlists_object().
natomic-object

Usage

is.natomic(x)

is.nlist(x)

is.nlists(x)

Arguments

x The object to test.

Value

A flag indicating whether x is a natomic object (is atomic and numeric) or inherits from nlist or nlists.

Functions

- is.natomic: Is natomic
- is.nlist: Is nlist
- is.nlists: Is nlists

Examples

# is.natomic
is.natomic(list(x = 1))
is.natomic(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)))

natomic-object

natomic Object

Description

An natomic object is an integer or double vector, matrix or array.
Examples

```r
mat <- matrix(c("1", "2", "3", "4"), 2L)
mat
is.natomic(mat)
mat <- numericize(mat)
mat
is.natomic(mat)
```

---

### nchains

**Number of MCMC chains**

**Description**

Gets the number of chains of an MCMC object.

**Usage**

```r
nchains(x, ...)
```

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

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

**Arguments**

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

**Value**

A count indicating the number of MCMC chains

**Methods (by class)**

- `nlist`: Number of chains of an `nlist::nlist_object`
  - Always 1L.
- `nlists`: Number of chains of an `nlist::nlists_object`

**Examples**

```r
nchains(nlist(x = 1:2))
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 MCMC iterations (in a chain).

Usage

```r
niters(x, ...)  
```

```r  
## S3 method for class 'nlist'

```nlist```

```r  
## S3 method for class 'nlists'

```nlists```

Arguments

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

Value

A count indicating the number of MCMC iterations.

Methods (by class)

- `nlist`: Number of MCMC iterations for an nlist object
  
  Always 1.

- `nlists`: Number of MCMC iterations for an nlists object

Examples

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

---

Create nlist Object

### Description

Creates a `nlist_object()` from one or more uniquely named `natomic_object()` arguments.

### Usage

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

### Arguments

... 

Uniquely named atomic numeric objects.

### Value

An nlist object.

### Examples

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

---

nlist-object

---

### Description

An nlist object is an S3 class list of uniquely named `natomic_object()` elements.

### Details

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

### Examples

```r
nlist <- nlist(x = 1, zz = matrix(1:9, 3))
str(nlist)
nlist
```
nlists

Create nlists Object

Description

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

Usage

nlists(...)

Arguments

... nlist objects.

Value

An nlists object.

Examples

nlists()

nlists(nlist())

nlists(nlist(x = 1))

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

nlists-object

nlits Object

Description

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

Details

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

Examples

nlists(nlist(x = 1, zz = matrix(1:9, 3)), nlist(x = 3.5, zz = matrix(2:10, 3)))
npdims.nlist  

### Description

Gets the number of dimensions of each `natomic_object()` in an `nlist_object()`.

### Usage

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

### Arguments

- **x**
  
  The nlist object

- **...**
  
  Unused.

### Value

A named list of the number of dimensions of each numeric atomic object.

### See Also

- `pdims()`

### Examples

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

npdims.nlists  

### Description

Gets the number of dimensions of each `natomic_object()` in an `nlists_object()`.

### Usage

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

### Arguments

- **x**
  
  The nlists object

- **...**
  
  Unused.
**nsams**

### Description

A named list of the number of dimensions of each numeric atomic object.

### See Also

`pdims()`

### Examples

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

---

**nsams**  
*Number of MCMC Samples*

### Description

Gets the number of MCMC samples (simulations * terms)

### Usage

```r
nsams(x, ...)
```

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

### Arguments

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

### Value

A count of the total number of sample values.

### Methods (by class)

- default: Number of sample values of default object

### See Also

`nsims()` and `nterms()`
Examples
nsims(nlist(x = 2))
nsims(nlist(x = 1:2))
nsims(nlists(nlist(x = c(2, 9)), nlist(x = c(1, 7))))

nsims
Number of MCMC Simulations

Description
Gets the number of MCMC simulations (iterations * chains)

Usage
nsims(x, ...)

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

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

Arguments
x The object
...	Unused.

Value
A count of the number of simulations.

Methods (by class)
- nlist: Number of simulations of an nlist object
  Always 1L.
- nlists: Number of simulations of an nlists object

See Also
niters() and nchains()

Examples
nsims(nlist(x = 1:2))
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 nlist_object().

**Usage**

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

**Arguments**

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

**Value**

A count of the number of terms.

**Examples**

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

**nterms.nlists**

**Number of Terms**

**Description**

Gets the number of terms of an nlists_object().

**Usage**

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

**Arguments**

- `x`  The object.
- `...` Unused.
**numericise**

**Value**
A count of the number of terms.

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

---

**numericise**  
**Numericise (or Numericize)**

**Description**
Attempts to coerce a non-numeric R object to `natomic_object()` or list of `{natomic_object}`. If possible the dimensionality is preserved.

**Usage**
```r
numericise(x, ...)
numericize(x, ...)

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

## S3 method for class 'logical'
numericise(x, ...)

## S3 method for class 'integer'
numericise(x, ...)

## S3 method for class 'double'
numericise(x, ...)

## S3 method for class 'factor'
numericise(x, ...)

## S3 method for class 'Date'
numericise(x, ...)

## S3 method for class 'POSIXct'
numericise(x, ...)

## S3 method for class 'hms'
numericise(x, ...)
```
numericise(x, ...)

## S3 method for class 'matrix'
numericise(x, ...)

## S3 method for class 'array'
numericise(x, ...)

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

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

Arguments

x An R object.
...

Details

Date, POSIXct and hms objects are floored first.

numericize() is an alias for numericise.

Value

The modified object.

Methods (by class)

- default: Numericise default object
- logical: Numericise logical vector
- integer: Numericise integer vector
- double: Numericise double vector
- factor: Numericise factor
- Date: Numericise Date
- POSIXct: Numericise POSIXct
- hms: Numericise hms
- matrix: Numericise matrix
- array: Numericise array
- list: Numericise list
- data.frame: Numericise data.frame
Examples

numericize(TRUE)
numericize("1.9")
numericize(factor(c("beta", "alpha")))
numericize(matrix(c(TRUE, FALSE, NA, TRUE), 2))
numericize(as.Date("1970-02-03"))
numericize(as.POSIXct("1970-02-03", tz = "GMT"))

---

pars.nlist | Parameter Names
---

Description

Gets the parameter names for an nlist_object().

Usage

```r
## S3 method for class 'nlist'
pars(x, scalar = NA, terms = FALSE, ...)
```

Arguments

- `x`: An nlist object.
- `scalar`: A logical scalar specifying whether to get the names of all parameters (NA), only scalars (TRUE) or all parameters except scalars (FALSE).
- `terms`: A flag specifying whether to return the parameter name for each term.
- `...`: Unused.

Value

A character vector of the parameter names.

Examples

```r
term::pars(nlist(zz = 1, y = 3:6))
```
pars.nlists  Parameter Names

Description

Gets the parameter names for an nlists_object().

Usage

```r
## S3 method for class 'nlists'
pars(x, scalar = NA, terms = FALSE, ...)
```

Arguments

- `x`: An nlists object.
- `scalar`: A logical scalar specifying whether to get the names of all parameters (NA), only scalars (TRUE) or all parameters except scalars (FALSE).
- `terms`: A flag specifying whether to return the parameter name for each term.
- `...`: Unused.

Value

A character vector of the parameter names.

Examples

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

pdims.nlist  Parameter Dimensions of an nlist Object

Description

 Gets the dimensions of an nlist object.

Usage

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

Arguments

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

A named list of the dimensions of each `natomic_object()`.

See Also

`term::pdims()`

Examples

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

---

**pdims.nlists**  
*Dimensions of an nlists Object*

Description

Gets the dimensions of an nlists object.

Usage

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

Arguments

- `x`  
  An nlists object.

- `...`  
  Unused

Value

A named list of the dimensions of each `natomic_object()`.

See Also

`term::pdims()`

Examples

```r
term::pdims(nlists(nlist(x = 1:3)))
term::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**
An atomic 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))
```

---

split_chains  
**Split Chains**

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

**Usage**

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

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

**Arguments**
- `x`  
  An MCMC object.
- `...`  
  Unused.
subset.nlist

Methods (by class)

- nlists: Split chains for an `nlists_object`.

---

**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"))
```
Description

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

Usage

```r
## 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

```r
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))
```
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
An named atomic 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 atomic vector where the names are the terms.

Usage
```r
unlist_nlist(x)
```

Arguments
- `x` An nlist object.
Value

An named atomic 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)))

Description

Validate Numeric Atomic, nlist Object or nlists Object

Usage

vld_natomic(x)
vld_nlist(x)
vld_nlists(x)

Arguments

x The object to check.

Value

A flag indicating whether the object was validated.

Functions

• vld_natomic: Validate Numeric Atomic
• vld_nlist: Validate nlist Object
• vld_nlists: Validate nlists Object

See Also

chk_natomic()
**Examples**

```r
#' vld_natomic
vld_natomic(1)
vld_natomic(matrix(1L))
try(vld_natomic(TRUE))

#' 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.nlist, 3
aggregate.nl, 3
as.mcmc.list.nl, 4
as.mcmc.list.nl, 5
as.mcmc.nl, 5
as.mcmc.nl, 6
as.nl, 7
as.nl.numeric(), 26, 30
as.nl, 8

chk_natomic, 8
chk_natomic(), 30
chk_nl(chk_natomic), 8
chk_nl(chk_natomic), 8
coda::mcmc(), 4–6
collapse_chains, 10

fillna, 10

is.natomic(is_natomic), 11
is.nl(is_natomic), 11
is.nl(is_natomic), 11
is_natomic, 11

natomic-object, 12
natomic_object(natomic-object), 12
natomic_object(), 11, 15, 17, 21, 25
nch, 13
nch(), 19
niter, 14
niter(), 19
nl, 15
nl::nlObject(), 13
nl::nlObject(), 13
nl_object(nlObject), 15
nl_object(), 3, 7, 11, 15–17, 20, 23
nls, 16
nls-object, 16
nls_object(nls-object), 16

nlists_object(), 3, 8, 11, 16, 17, 20, 24, 27
npdms.nl, 17
npdms.nl, 17
nsams, 17
nsim, 19
nsim(), 18
nterms, 20
nterms.nl, 20
numericise, 21
numericize(numericise), 21

pars.nl, 23
pars.nl, 24
pdms(), 17, 18
pdms.nl, 24
pdms.nl, 25
relist.nl, 26
relist.nl(), 30

split_chains, 26
subset.nl, 27
subset.nl, 28

term::pdms(), 25

unlist.nl, 29
unlist.nl, 29
unlist.nl(), 26, 29

vl_natomic, 30
vl_natomic(), 9
vl_nl(vl_natomic), 30
vl_nlst(vl_natomic), 30

32