

Package ‘nlist’

September 25, 2020

Title Lists of Numeric Atomic Objects

Version 0.3.0

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,
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 7.1.1

R topics documented:

aggregate.nlist	3
aggregate.nlists	4
as_mcmc	4
as_mcmc_list	5
as_nlist	6
as_nlists	7
as_term.mcmc	8
as_term.nlist	8
as_term.nlists	9
as_term_frame	9
as_term_frame.nlist	10
as_term_frame.nlists	10
bind_iterations.mcmc	11
bind_iterations.mcmc.list	11
chk_nlist	12
collapse_chains.mcmc	13
collapse_chains.mcmc.list	13
collapse_chains.nlist	14
collapse_chains.nlists	15
complete_terms.mcmc	15
estimates.nlist	16
estimates.nlists	17
fill_all.nlist	17
fill_all.nlists	18
fill_na.nlist	19
fill_na.nlists	20
is_numeric	20
nchains.mcmc	21
nchains.mcmc.list	22
nchains.nlist	22
nchains.nlists	23
niters.mcmc	23
niters.mcmc.list	24
niters.nlist	25
niters.nlists	25
nlist	26
nlists	27
npdims.mcmc.list	27
npdims.nlist	28
npdims.nlists	29
nsims.nlist	29
nsims.nlists	30
nterms.mcmc	31
nterms.mcmc.list	31
nterms.nlist	32
nterms.nlists	32
pars.mcmc	33
pars.mcmc.list	34
pars.nlist	34
pars.nlists	35

pdims.mcmc	36
pdims.mcmc.list	36
pdims.nlist	37
pdims.nlists	37
relist_nlist	38
set_pars.mcmc	39
set_pars.mcmc.list	39
set_pars.nlist	40
set_pars.nlists	41
split_chains.nlists	41
subset.mcmc	42
subset.mcmc.list	43
subset.nlist	44
subset.nlists	44
thin.default	45
tidy.mcmc	46
tidy.mcmc.list	46
tidy.nlists	47
unlist.nlist	47
unlist_nlist	48
vld_nlist	49

Index 50

aggregate.nlist	<i>Aggregate nlist</i>
-----------------	------------------------

Description

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

Usage

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

Arguments

<code>x</code>	An nlist object.
<code>fun</code>	A function that given a numeric vector returns a numeric scalar.
<code>...</code>	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	<i>Aggregate nlists</i>
------------------	-------------------------

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

<code>x</code>	An nlist object.
<code>fun</code>	A function that given a numeric vector returns a numeric scalar.
<code>...</code>	Additional arguments passed to <code>fun</code> .
<code>by_chain</code>	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	<i>Coerce to mcmc Object</i>
---------	------------------------------

Description

Coerce an R object to an mcmc object.

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.

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

```
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.

Examples

```
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	<i>Coerce to nlist</i>
----------	------------------------

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, ...)

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

## S3 method for class 'mcmc.list'
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`
- `mcmc`: Coerce `mcmc` (with one iteration) to `nlist`
- `mcmc.list`: Coerce `mcmc.list` (with one iteration) to `nlist`

Examples

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

`as_nlists`*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 'nlist'
as_nlists(x, ...)
```

Arguments

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

Value

An `nlists` object.

Methods (by class)

- `list`: Coerce list to `nlists`
- `mcmc`: Coerce `mcmc` to `nlists`
- `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.mcmc *Coerce to a Term Vector*

Description

Coerce to a Term Vector

Usage

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

Arguments

x An object.
... Unused.

Examples

```
as_term(as_mcmc(nlist(x = matrix(1:4, ncol = 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	<i>Coerce to a Term Vector</i>
----------------	--------------------------------

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	<i>Coerce to a Term Frame</i>
---------------	-------------------------------

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)
))
```

`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

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

Value

The combined object.

See Also

Other MCMC manipulations: [bind_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

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

Value

The combined object.

See Also

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

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

```
## 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

```
## 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\(\)](#), [bind_iterations\(\)](#), [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\(\)](#), [bind_iterations\(\)](#), [estimates\(\)](#), [split_chains\(\)](#)

Examples

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

complete_terms.mcmc *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.nlist	<i>Estimates</i>
-----------------	------------------

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\(\)](#), [bind_iterations\(\)](#), [collapse_chains\(\)](#), [split_chains\(\)](#)

Examples

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

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

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	<i>Fill All Values</i>
----------------	------------------------

Description

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

Usage

```
## S3 method for class '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	<i>Fill All Values</i>
-----------------	------------------------

Description

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

Usage

```
## S3 method for class '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	<i>Fill Missing Values</i>
----------------	----------------------------

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	<i>Is numeric, nlist or nlists</i>
------------	------------------------------------

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

Description

Gets the number of chains of an MCMC object.

Usage

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

Arguments

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

Value

An integer scalar of the number of chains.

See Also

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

nchains.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\(\)](#)

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	<i>Number of Terms</i>
----------------	------------------------

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

Description

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

Usage

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

Arguments

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

Value

An integer scalar of the number of iterations.

See Also

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

`niters.mcmc.list` *Number of Iterations*

Description

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

Usage

```
## 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.

See Also

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

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

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	<i>Number of Iterations</i>
---------------	-----------------------------

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))
```

<code>nlists</code>	<i>Create nlists Object</i>
---------------------	-----------------------------

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))
```

<code>npdims.mcmc.list</code>	<i>Number of Parameter Dimensions</i>
-------------------------------	---------------------------------------

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 'mcmc.list'
npdims(x, ...)
```

Arguments

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

Value

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

See Also

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

npdims.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	<i>Number of Parameter Dimensions</i>
---------------	---------------------------------------

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	<i>Number of Simulations</i>
-------------	------------------------------

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	<i>Number of Simulations</i>
--------------	------------------------------

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.mcmc	<i>Number of Terms</i>
-------------	------------------------

Description

Gets the number of terms of an MCMC object.

Usage

```
## 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\(\)](#), [niters\(\)](#), [npars\(\)](#), [nsams\(\)](#), [nsims\(\)](#)

nterms.mcmc.list	<i>Number of Terms</i>
------------------	------------------------

Description

Gets the number of terms of an MCMC object.

Usage

```
## 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\(\)](#), [niters\(\)](#), [npars\(\)](#), [nsams\(\)](#), [nsims\(\)](#)

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

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	<i>Number of Terms</i>
---------------	------------------------

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.mcmc

Parameter Names

Description

Gets the parameter names.

Usage

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

Arguments

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

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

Arguments

<code>x</code>	An object.
<code>scalar</code>	A logical scalar specifying whether to include all parameters (NULL), only scalars (TRUE) or all parameters except scalars (FALSE).
<code>terms</code>	A flag specifying whether to return the parameter name for each term element.
<code>...</code>	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

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

Arguments

<code>x</code>	An object.
<code>scalar</code>	A logical scalar specifying whether to include all parameters (NULL), only scalars (TRUE) or all parameters except scalars (FALSE).
<code>terms</code>	A flag specifying whether to return the parameter name for each term element.
<code>...</code>	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	<i>Parameter Names</i>
-------------	------------------------

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.mcmc`*Parameter Dimensions*

Description

Gets the dimensions of each parameter of an object.

Usage

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

Arguments

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

Value

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

See Also

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

`pdims.mcmc.list`*Parameter Dimensions*

Description

Gets the dimensions of each parameter of an object.

Usage

```
## S3 method for class 'mcmc.list'  
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.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.mcmc	<i>Set Parameters</i>
---------------	-----------------------

Description

Sets an object's parameter names.

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

Usage

```
## 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	<i>Set Parameters</i>
--------------------	-----------------------

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	<i>Set Parameter Names</i>
----------------	----------------------------

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", "c1"))
```

set_pars.nlists	<i>Set Parameter Names</i>
-----------------	----------------------------

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")
```

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

Description

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

Usage

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

subset.mcmc	<i>Subset mcmc Object</i>
-------------	---------------------------

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, ...)
```

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	<i>Subset mcmc.list Object</i>
------------------	--------------------------------

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

```
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	<i>Subset nlist Object</i>
--------------	----------------------------

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	<i>Subset nlists Object</i>
---------------	-----------------------------

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

This is 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.mcmc	<i>Turn an object into a tidy tibble</i>
-----------	--

Description

Turn an object into a tidy tibble

Usage

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

Arguments

x	An object to be converted into a tidy <code>tibble::tibble()</code> .
...	Additional arguments to tidying method.

Value

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

Methods

No methods found in currently loaded packages.

tidy.mcmc.list	<i>Turn an object into a tidy tibble</i>
----------------	--

Description

Turn an object into a tidy tibble

Usage

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

Arguments

x	An object to be converted into a tidy <code>tibble::tibble()</code> .
...	Additional arguments to tidying method.

Value

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

Methods

No methods found in currently loaded packages.

tidy.nlists	<i>Turn an object into a tidy tibble</i>
-------------	--

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 <code>tibble::tibble()</code> .
...	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	<i>Flatten nlist Object</i>
--------------	-----------------------------

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	<i>Validate nlist Object or nlists Object</i>
-----------	---

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.nlist, 3
aggregate.nlists, 4
as.nlist (as_nlist), 6
as.nlists (as_nlist), 6
as_mcmc, 4
as_mcmc_list, 5
as_nlist, 6
as_nlist.numeric(), 38, 48
as_nlists, 7
as_term.mcmc, 8
as_term.nlist, 8
as_term.nlists, 9
as_term_frame, 9
as_term_frame.nlist, 10
as_term_frame.nlists, 10

bind_chains, 11–17, 42
bind_iterations, 13–17, 42
bind_iterations.mcmc, 11
bind_iterations.mcmc.list, 11

chk_nlist, 12
chk_nlists (chk_nlist), 12
collapse_chains, 11, 12, 16, 17, 42
collapse_chains.mcmc, 13
collapse_chains.mcmc.list, 13
collapse_chains.nlist, 14
collapse_chains.nlists, 15
complete_terms.mcmc, 15

dims, 28, 29, 36–38

estimates, 11–15, 42
estimates.nlist, 16
estimates.nlists, 17

fill_all, 19, 20
fill_all.nlist, 17
fill_all.nlists, 18
fill_na, 18, 19
fill_na.nlist, 19
fill_na.nlists, 20

is_nlist (is_numeric), 20
is_nlists (is_numeric), 20

is_numeric, 20

nchains, 23–26, 30–33
nchains(), 29, 30
nchains.mcmc, 21
nchains.mcmc.list, 22
nchains.nlist, 22
nchains.nlists, 23
ndims, 28, 29, 36–38
niters, 21–23, 30–33
niters(), 29, 30
niters.mcmc, 23
niters.mcmc.list, 24
niters.nlist, 25
niters.nlists, 25
nlist, 26
nlist-object (nlist), 26
nlist_object (nlist), 26
nlist_object(), 3, 4, 6, 20, 26, 27
nlists, 27
nlists-object (nlists), 27
nlists_object (nlists), 27
nlists_object(), 4, 7, 20, 27
npars, 21–26, 30–35, 39–41
npdims, 36–38
npdims.mcmc.list, 27
npdims.nlist, 28
npdims.nlists, 29
nsams, 21–26, 30–33
nsims, 21–26, 31–33
nsims.nlist, 29
nsims.nlists, 30
nterms, 21, 22, 24–26, 30
nterms.mcmc, 31
nterms.mcmc.list, 31
nterms.nlist, 32
nterms.nlists, 32

pars, 39–41
pars.mcmc, 33
pars.mcmc.list, 34
pars.nlist, 34
pars.nlists, 35
pdims, 28, 29

`pdims()`, 27–29
`pdims.mcmc`, 36
`pdims.mcmc.list`, 36
`pdims.nlist`, 37
`pdims.nlists`, 37

`relist_nlist`, 38
`relist_nlist()`, 48

`set_pars`, 33–35
`set_pars.mcmc`, 39
`set_pars.mcmc.list`, 39
`set_pars.nlist`, 40
`set_pars.nlists`, 41
`split_chains`, 11–17
`split_chains.nlists`, 41
`subset.mcmc`, 42
`subset.mcmc.list`, 43
`subset.nlist`, 44
`subset.nlists`, 44

`thin.default`, 45
`tibble::tibble()`, 46, 47
`tidy.mcmc`, 46
`tidy.mcmc.list`, 46
`tidy.nlists`, 47

`unlist.nlist`, 47
`unlist_nlist`, 48
`unlist_nlist()`, 38, 48

`vld_nlist`, 49
`vld_nlists(vld_nlist)`, 49