

Package ‘kimisc’

July 2, 2014

Encoding UTF-8

Type Package

Title Kirill's miscellaneous functions

Version 0.2-1

Date 2014-04-02

Author Kirill Mueller

Maintainer Kirill Mueller <mail@kirill-mueller.de>

Description A collection of useful functions not found anywhere else, mainly for programming: Generalized lagged differences, checking containment in an interval, creating a factor where the levels maintain the order of appearance, sampling rows from a data frame, converting seconds from midnight to and from H:M:S format, choosing the first non-NA value, transposing lists of lists, returning the name of the file currently sourced, smart named lists and vectors, and an alternative interface to assign().

License GPL-3

Imports plyr

Suggests testthat

URL <https://github.com/kr1mlr/kimisc>

BugReports <https://github.com/kr1mlr/kimisc/issues>

Collate 'coalesce_na.R' 'export.R' 'gdiff.R' 'hms_to_seconds.R'
'in_interval.R' 'kimisc-package.R' 'nlist.R' 'ofactor.R'
'sample_data_frame.R' 'seconds_to_hms.R' 'thisfile.R' 'tll.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2014-04-02 11:55:48

R topics documented:

kimisc-package	2
coalesce.na	3
export	3
export.list	4
gdiff	5
hms.to.seconds	6
in.interval.lo	7
in.interval.ro	7
nc	8
nin.interval.lo	9
nin.interval.ro	10
nlist	10
ofactor	11
sample.rows	12
seconds.to.hms	12
setMissingNames	13
thisfile	14
tll	15
Index	16

kimisc-package	<i>Kirill's miscellaneous functions</i>
----------------	-----------------------------------------

Description

A collection of useful functions not found anywhere else, mainly for programming: Generalized lagged differences, checking containment in an interval, creating a factor where the levels maintain the order of appearance, sampling rows from a data frame, converting seconds from midnight to and from H:M:S format, choosing the first non-NA value, transposing lists of lists, returning the name of the file currently sourced, smart named lists and vectors, and an alternative interface to assign().

Details

Package: kimisc
 Type: Package
 Version: 0.2-1
 Date: 2014-04-02
 License: GPL v3

Author(s)

Kirill Müller

coalesce.na	<i>Replaces NA values</i>
-------------	---------------------------

Description

This (vectorized) function returns the first non-NA argument, similar to the SQL function COALESCE. If a vector or matrix is passed as first argument, the remaining arguments are recycled to generate a vector/matrix of the same dimension, and coalescing is done element by element.

Usage

```
coalesce.na(x, ...)
```

Arguments

x	The first value to coalesce.
...	Other values to coalesce.

Value

A vector of the same length as x.

Examples

```
coalesce.na(NA, -1)
coalesce.na(5, 3)
coalesce.na(c(1,NA,NA), c(NA,2))
coalesce.na(matrix(c(NA, 1:3), nrow=2))
coalesce.na(NA)
```

export	<i>Exports to an environment</i>
--------	----------------------------------

Description

This function is a wrapper around `export.list` that exports variables by their name to another environment.

Usage

```
export(..., target.env = .GlobalEnv)
```

Arguments

... variables to be exported.
target.env The target environment. Use the global environment by default.

Value

Invisible NULL.

Author(s)

Roland

References

<http://stackoverflow.com/a/17484932/946850>

See Also

[export.list](#), [assign](#)

Examples

```
local({
  newly.created.var <- 5
  export(newly.created.var)
})
newly.created.var
rm(newly.created.var)
```

export.list

Exports to an environment

Description

This function is a wrapper around [assign](#) that exports the contents of a named list to an environment. The variable names in the target environment are constructed from the names of the list items or taken from a separate argument.

Usage

```
export.list(arg.list, arg.names = names(arg.list), target.env = .GlobalEnv)
```

Arguments

arg.list list of objects, possibly named.
arg.names names to use for the items in the target environment. Use the names of arg.list by default.
target.env The target environment. Use the global environment by default.

Value

Invisible NULL.

Author(s)

Roland

References

<http://stackoverflow.com/a/17484932/946850>

See Also

[export](#), [assign](#)

Examples

```
export.list(list(newly.created.var=5))
newly.created.var
rm(newly.created.var)
```

gdiff

Generalized lagged differences

Description

Returns suitably lagged and iterated differences using arbitrary difference functions.

Usage

```
gdiff(x, lag = 1L, differences = 1L, FUN = `-`, ...)
```

Arguments

FUN	A distance function that accepts two parameters
...	further arguments to be passed to or from methods.
x	a numeric vector or matrix containing the values to be differenced.
lag	an integer indicating which lag to use.
differences	an integer indicating the order of the difference.

Value

If x is a vector of length n and $differences = 1$, then the computed result is equal to the successive differences $FUN(x[(1+lag):n], x[1:(n-lag)])$.

If difference is larger than one this algorithm is applied recursively to x . Note that the returned value is a vector which is shorter than x .

If x is a matrix then the difference operations are carried out on each column separately.

See Also[diff](#)**Examples**

```
gdiff(1:4)
gdiff(1:4, FUN = `/\`)
```

hms.to.seconds	<i>Converts a time value given in H:M:S format to the number of seconds since midnight</i>
----------------	--------------------------------------------------------------------------------------------

Description

This function is very similar to `strptime` with the `%X` conversion specification. Anything with three numbers between two colons is interpreted as a time, no consistency check is performed on the actual hour, minute and second values. Thus, strings like `25:15:00` and `23:78:101` also will be converted. Incorrectly formatted strings are converted to `NA` with a warning.

Usage

```
hms.to.seconds(x)
```

Arguments

`x` A (vector of) strings in H:M:S format.

Value

A (vector of) integer values of the same length as `x`.

See Also[strptime](#)**Examples**

```
hms.to.seconds(c("00:00:01", "00:01:00", "01:00:00"))
hms.to.seconds(c("25:15:00", "23:78:101"))
hms.to.seconds("invalid")
```

`in.interval.lo` *Checks if values are contained in an interval (open on the left)*

Description

This function checks if the values in the `x` parameter are contained in the interval $(lo, hi]$. NA values are treated as "not in the interval".

Usage

```
in.interval.lo(x, lo, hi)
```

Arguments

<code>x</code>	A vector of values. (Lists will be coerced to a numeric vector.)
<code>lo</code>	Left end of the interval.
<code>hi</code>	Right end of the interval.

Value

A boolean vector of the same length as `x`.

See Also

[in.interval.ro](#), [nin.interval.lo](#), [nin.interval.ro](#)

Examples

```
in.interval.lo(c(-1, 0, 1, 2), 0, 1)
in.interval.lo(NA, 1, 3)
```

`in.interval.ro` *Checks if values are contained in an interval (open on the right)*

Description

This function checks if the values in the `x` parameter are contained in the interval $[lo, hi)$. NA values are treated as "not in the interval".

Usage

```
in.interval.ro(x, lo, hi)
```

Arguments

x	A vector of values. (Lists will be coerced to a numeric vector.)
lo	Left end of the interval.
hi	Right end of the interval.

Value

A boolean vector of the same length as x.

See Also

[in.interval.lo](#), [nin.interval.lo](#), [nin.interval.ro](#)

Examples

```
in.interval.ro(c(-1, 0, 1, 2), 0, 1)
in.interval.ro(NA, 1, 3)
```

nc

Smart named vector

Description

This function is a wrapper around `c` that assigns names to unnamed arguments based on the unevaluated expression used in the call.

Usage

```
nc(...)
```

Arguments

... Vector elements, possibly named

Value

A named vector.

Author(s)

Hadley Wickham

References

<http://stackoverflow.com/a/5043280/946850>, <http://tolstoy.newcastle.edu.au/R/e9/help/10/03/8392.html>

See Also[c](#), [nlist](#)**Examples**

```
a <- 1; b <- 2; c <- 3
nc(a, b, d=c)
nc(mean(c(a, b, c)))
```

nin.interval.lo	<i>Checks if values are outside of an interval (open on the left)</i>
-----------------	-----------------------------------------------------------------------

Description

This function checks if the values in the `x` parameter are contained in the interval `(lo, hi]`. NA values are treated as "not in the interval".

Usage

```
nin.interval.lo(x, lo, hi)
```

Arguments

<code>x</code>	A vector of values. (Lists will be coerced to a numeric vector.)
<code>lo</code>	Left end of the interval.
<code>hi</code>	Right end of the interval.

Value

A boolean vector of the same length as `x`.

See Also[in.interval.lo](#), [in.interval.ro](#), [nin.interval.ro](#)**Examples**

```
nin.interval.lo(c(-1, 0, 1, 2), 0, 1)
nin.interval.lo(NA, 1, 3)
```

nin.interval.ro	<i>Checks if values are outside of an interval (open on the right)</i>
-----------------	------------------------------------------------------------------------

Description

This function checks if the values in the `x` parameter are contained in the interval `[lo, hi)`. NA values are treated as "not in the interval".

Usage

```
nin.interval.ro(x, lo, hi)
```

Arguments

<code>x</code>	A vector of values. (Lists will be coerced to a numeric vector.)
<code>lo</code>	Left end of the interval.
<code>hi</code>	Right end of the interval.

Value

A boolean vector of the same length as `x`.

See Also

[in.interval.lo](#), [in.interval.ro](#), [nin.interval.lo](#)

Examples

```
nin.interval.ro(c(-1, 0, 1, 2), 0, 1)
nin.interval.ro(NA, 1, 3)
```

nlist	<i>Smart named list</i>
-------	-------------------------

Description

This function is a wrapper around [list](#) that assigns names to unnamed arguments based on the unevaluated expression used in the call.

Usage

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

Arguments

<code>...</code>	List items, possibly named
------------------	----------------------------

Value

A named list.

Author(s)

Hadley Wickham

References

<http://stackoverflow.com/a/5043280/946850>, <http://tolstoy.newcastle.edu.au/R/e9/help/10/03/8392.html>

See Also

[list](#)

Examples

```
a <- 1; b <- 2; c <- 3
nlist(a, b, d=c)
nlist(mean(c(a, b, c)))
```

ofactor

Order-preserving factors

Description

The function `ofactor` is a convenience wrapper for `factor` that orders the levels as they appear in the data if the `levels` argument is not specified.

Usage

```
ofactor(x = character(), ...)
```

Arguments

`x` A vector of data, usually taking a small number of distinct values.
`...` Other arguments passed on to [factor](#).

Value

A factor. See [factor](#) for details.

Examples

```
ofactor(3:1)
ofactor(9:12, exclude=11)
identical(ofactor(3:1, levels=1:3), factor(3:1))
```

sample.rows	<i>Random Samples and Permutations for Data Frames</i>
-------------	--------------------------------------------------------

Description

This function takes a sample of the specified size from the rows of `x` using either with or without replacement.

Usage

```
sample.rows(x, size, replace = FALSE, prob = NULL)
```

Arguments

<code>x</code>	A data frame.
<code>size</code>	A non-negative integer giving the number of items to choose.
<code>replace</code>	Should sampling be with replacement?
<code>prob</code>	A vector of probability weights for obtaining the rows of the data frame being sampled.

Details

This function internally calls [sample.int](#).

Value

A data frame of the same shape as `x`.

Examples

```
set.seed(42)
sample.rows(data.frame(a=c(1,2,3), b=c(4,5,6), row.names=c('a', 'b', 'c')), 10, replace=TRUE)
```

seconds.to.hms	<i>Converts a time value given as number of seconds since midnight to the H:M:S format</i>
----------------	--------------------------------------------------------------------------------------------

Description

This function is very similar to `strftime` with the `%X` conversion specification. Hour values larger than 24 are permitted. Fractions will be rounded down to the next integer. Non-numeric values are coerced to NA with a warning.

Usage

```
seconds.to.hms(x)
```

Arguments

x A (vector of) numbers.

Value

A (vector of) character values of the same length as x.

See Also

[strftime](#)

Examples

```
seconds.to.hms(c(1, 60, 3600.5))
seconds.to.hms(c(100000, -4000.5))
seconds.to.hms("invalid")
```

setMissingNames

Set the Missing Names in an Object

Description

This function is an enhanced version of [setNames](#) in the sense that the elements that already have names are not renamed.

Usage

```
setMissingNames(object, nm)
```

Arguments

object an object for which a names attribute will be meaningful
nm a character vector of names to assign to the object

Value

An object of the same sort as object with the new names assigned to the unnamed elements.

Author(s)

Hadley Wickham, Kirill Müller

References

<http://stackoverflow.com/a/5043280/946850>

See Also

[setNames](#)

Examples

```
setMissingNames(c(a=1, b=2, 3), letters[2:4])
```

thisfile	<i>Determines the path of the currently running script</i>
----------	------------------------------------------------------------

Description

R does not store nor export the path of the currently running script. This is an attempt to circumvent this limitation by applying heuristics (such as call stack and argument inspection) that work in many cases.

Usage

```
thisfile()
```

Details

This function currently works only if the script was sourced or run with `Rscript` or using the `--file` parameter to the R executable. For code run with `Rscript`, the exact value of the parameter passed to `Rscript` is returned.

Value

The path of the currently running script, NULL if it cannot be determined.

Author(s)

Kirill Müller, Hadley Wickham, Michael R. Head

References

<http://stackoverflow.com/q/1815606/946850>

See Also

[source](#), [Rscript](#), [getwd](#)

Examples

```
## Not run: thisfile()
```

tll	<i>Transposes a list of lists</i>
-----	-----------------------------------

Description

The argument is assumed to be a list of n (named) lists with length m each. It is converted to a (named) list of m elements with length n each.

Usage

```
tll(1)
```

Arguments

1 List of lists, possibly named.

Value

A list of lists corresponding to a transposition of the argument.

See Also

[t](#)

Examples

```
tll(list(list(1, 2), list(3, 4)))  
tll(list(list(a=1, b=2), list(a=3, b=4)))  
tll(list(x=list(a=1, b=2), y=list(a=3, b=4)))
```

Index

*Topic **package**
 kimisc-package, 2

assign, 4, 5

c, 8, 9
coalesce.na, 3

diff, 6

export, 3, 5
export.list, 3, 4, 4

factor, 11

gdiff, 5
getwd, 14

hms.to.seconds, 6

in.interval.lo, 7, 8–10
in.interval.ro, 7, 7, 9, 10

kimisc (kimisc-package), 2
kimisc-package, 2

list, 10, 11

nc, 8
nin.interval.lo, 7, 8, 9, 10
nin.interval.ro, 7–9, 10
nlist, 9, 10

ofactor, 11

Rscript, 14

sample.int, 12
sample.rows, 12
seconds.to.hms, 12
setMissingNames, 13
setNames, 13

source, 14
strftime, 13
strptime, 6

t, 15
thisfile, 14
tll, 15