Package 'kimisc'

February 14, 2016

Encoding UTF-8

Type Package

Title Kirill's Miscellaneous Functions

Version 0.3

Date 2016-02-14

Description A collection of useful functions not found anywhere else, mainly for programming: Pretty intervals, 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, pryr

Suggests testthat (>= 0.10.0)

Enhances knitr

URL http://krlmlr.github.io/kimisc

URLNote https://github.com/krlmlr/kimisc

BugReports https://github.com/krlmlr/kimisc/issues

RoxygenNote 5.0.1

NeedsCompilation no

Author Kirill Müller [aut, cre]

Maintainer Kirill Müller <krlmlr+r@mailbox.org>

Repository CRAN

Date/Publication 2016-02-14 17:30:21

R topics documented:

`kimisc-package` *Kirill's Miscellaneous Functions*

Description

A collection of useful functions not found anywhere else, mainly for programming: Pretty intervals, 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().

Author(s)

Kirill Müller

coalesce.na *Replaces NA values*

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)
```

cut_format *Convert Numeric to Factor, with custom formatting*

Description

This is an enhanced version of cut that allows a custom formatting to be applied to the values.

Usage

```
cut_format(x, breaks, include.lowest = FALSE, right = TRUE,
  ordered_result = FALSE, ..., format_fun = format, sep = ", ",
  paren = c("(", "[", ")", "]"))
```

Arguments

x	a numeric vector which is to be converted to a factor by cutting.
breaks	A numeric vector of two or more unique cut points
include.lowest	logical, indicating if an 'x[i]' equal to the lowest (or highest, for `right = FALSE`) 'breaks' value should be included.
right	logical, indicating if the intervals should be closed on the right (and open on the left) or vice versa.
ordered_result	logical: should the result be an ordered factor?
...	Passed to `cut`
format_fun	`[function(x): character]` A vectorized function that performs the desired formatting. Default: `format`
sep	`[character(1)]` The separator between lower and upper end of the interval. Default: `", "`
paren	`[character(4)]` Opening and closing parentheses in two variants. Default: `c("(", "[", ")", "]")`

See Also

http://stackoverflow.com/q/14456371/946850

Examples

```
cut_format(runif(10), seq(0, 1, by = 0.25), format_fun = function(x) paste(x * 100, "%"))
cut_format(runif(10), seq(0, 1, by = 0.25), paren = c("<", "{", ">", "}"))
```

---

df_to_list *Converts a name-value data frame to a named list*

Description

This function converts a data frame back to a list. It is the reverse operation to list_to_df.

Usage

```
df_to_list(df_for_list)
```

Arguments

df_for_list	The data frame to be converted to a list

Details

In a data frame with more than two columns, heuristics are applied to detect the name and value column.

export *Exports to an environment*

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

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.
FUN	A distance function that accepts two parameters
...	further arguments to be passed to or from methods.

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 *Converts a time value given in H:M:S format to the number of seconds since midnight*

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

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.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)
```

list_to_df *Converts a list to a name-value data frame*

Description

This function coerces its input to a list and returns a data frame with as many rows as there are list items in the input, and two columns (one for the names, one for the values). If the list is not named, the natural sequence will be used as item names.

Usage

```
list_to_df(list_for_df)
```

Arguments

list_for_df	The object to be converted to a data frame

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` *Checks if values are outside of 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

```
nin.interval.lo(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, 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 *Checks if values are outside of 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

```
nin.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, in.interval.ro, nin.interval.lo

Examples

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

nlist *Smart named list*

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

`...`	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 *Random Samples and Permutations for Data Frames*

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

`x`	A data frame.
`size`	A non-negative integer giving the number of items to choose.
`replace`	Should sampling be with replacement?
`prob`	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 *Converts a time value given as number of seconds since midnight to the H:M:S format*

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` *Determines the path of the currently running script*

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()

thisfile_source()

thisfile_rscript()

thisfile_knit()
```

Details

This functions currently work only if the script was `source`d, processed with `knitr`, 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 *Transposes a list of lists*

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(l)
```

Arguments

l	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)))
```

vswitch *Vectorized switch*

Description

The function vswitch is a vectorized version of switch optimized for performance.

Usage

```
vswitch(EXPR, ...)
```

Arguments

EXPR	an expression evaluating to a number or a character string.
...	the list of alternatives. If it is intended that EXPR has a character-string value these will be named, perhaps except for one alternative to be used as a 'default' value.

Details

Only the EXPR argument is treated as a vector. In particular, if any of the alternatives (or the default alternative) is a vector, the result will be a list of vectors.

Value

The value of one of the elements of . . ., or NA whenever no element is selected. Contrary to switch the result is always visible.

Index