Package ‘Nmisc’

October 12, 2022

Type Package

Title Miscellaneous Functions Used at 'Numeract LLC'

Version 0.3.7

Description Contains functions useful for debugging, set operations on vectors, and 'UTC' date and time functionality. It adds a few vector manipulation verbs to 'purrr' and 'dplyr' packages. It can also generate an R file to install and update packages to simplify deployment into production. The functions were developed at the data science firm 'Numeract LLC' and are used in several packages and projects.

URL `https://github.com/numeract/Nmisc`

BugReports `https://github.com/numeract/Nmisc/issues`

License MIT + file LICENCE

Encoding UTF-8

RoxygenNote 7.1.1

Language en-US

Depends R (>= 3.4)

Imports dplyr, magrittr, purrr, rappdirs, rlang, tibble, tidyselect, stringr

Suggests lubridate, testthat, covr

NeedsCompilation no

Author Mike Badescu [aut, cre], Ana-Maria Niculescu [aut], Teodor Ciuraru [ctb], Numeract LLC [cph]

Maintainer Mike Badescu <`mike.badescu@numeract.com`>

Repository CRAN

Date/Publication 2021-04-28 13:50:02 UTC

R topics documented:

`catn` *Concatenate with new line*

Description

Wrapper around cat which appends new line to output.

Usage

```
catn(...)
```

Arguments

`...`	Arguments to be passed to `cat` function.

Value

None

See Also

`cat`

`clear_warnings` *Avoid repeated warnings*

Description

Clear warnings for production code.

Usage

```
clear_warnings()
```

See Also

`warnings`

`format_utc` *Format Date and POSIXct*

Description

Converts Date and POSIXct objects to the format given as input.

Usage

```
format_utc(x, format = NULL, usetz = TRUE)
```

Arguments

`x`	A Date or POSIXct object to be converted.
`format`	A character string. The default format is "%Y-%m-%d" for Date and "%Y-%m-%d %H:%M:%S" for POSIXct.
`usetz`	Logical. If TRUE, the time zone abbreviation is appended to the output. Applicable only if an POSIXct object.

Value

A character string representing the formatted date.

See Also

`format.Date`, `format.POSIXct`

Examples

```
format_utc(Sys.time(), format = "%Y-%m-%d", usetz = FALSE)
```

`generate_install_file` *Generates an R file to install packages used by the project.*

Description

The function takes the output of `get_packages` and writes in a file the commands needed to install and update package used throughout the project.

Usage

```
generate_install_file(
  file,
  package_df = get_packages(),
  include_core_package = FALSE
)
```

Arguments

`file`	The name of the file to be created.
`package_df`	A data frame obtained with `get_packages` that contains information regarding the name, version and source of the package.
`include_core_package`	Logical, whether to include in the generated install file package which come with R by default

Value

Nothing

See Also

`get_packages`

Examples

```
## Not run:
package_df <- get_packages(package_options = c("library"))
generate_install_file("install_packages.R", package_df)

## End(Not run)
```

get_os	*Returns the name of the Operating System*

Description

A simple wrapper around `rappdirs:::get_os`, allowing it to be exported.

Usage

```
get_os()
```

Value

One of `"win"`, `"mac"`, `"unix"`, `"Unknown OS"`.

get_packages	*Get information about the package used in the project*

Description

The function returns a data frame containing information about packages that are loaded with `library()`, `require()`, used with `::` operator, listed in the DESCRIPTION file, and/or already loaded.

Usage

```
get_packages(
  project_path = ".",
  include_pattern = "\\.R(md)?$",
  exclude_pattern = "tests/",
  package_options = c("referenced", "library", "description")
)
```

Arguments

`project_path`	A string representing the path of the project root in which the function will look recursively in order to find files that fit `include_pattern`
`include_pattern`	A string representing a regex that matches project files in which to look for packages. By default, `get_packages` looks for all .R and .Rmd files in the current project.
`exclude_pattern`	A string representing a regex that matches project files to exclude. By default, `get_packages` excludes all files found in "tests" folder.

`package_options`	A character vector that represents the method through which packages are loaded or referenced. The options are: `referenced` for packages referenced by the `::` operator, `library` for packages loaded using `library()` or `require()`, `description` for packages mentioned in `DESCRIPTION` file, and `loaded` for packages already loaded in the current session.

Value

A data frame containing package information:

`package_name`	The name of the package
`requested_by`	The context in which the package was used
`is_base`	Whether package is part of the core R packages
`source`	The source from which the package was installed
`version`	The version of the package, if installed locally
`is_installed`	Whether the package is installed locally

See Also

`generate_install_file`

Examples

```
## Not run:
package_df <- get_packages(
    project_path = '.',
    include_pattern = '\\.R$',
    exclude_pattern = '',
    package_options = c('referenced'))

## End(Not run)
```

`is.POSIXct` *Is it a POSIXct object?*

Description

Is it a POSIXct object?

Usage

```
is.POSIXct(x)
```

Arguments

`x`	An R object.

See Also

`lubridate::is.POSIXct`

`keep_at` *Keep or discard elements*

Description

`keep_at()` keeps only the elements from specific positions while `discard_at()` does the opposite. The functions are wrappers around `purrr::keep` and `purrr::discard`, respectively.

Usage

```
keep_at(.x, .at)

discard_at(.x, .at)
```

Arguments

`.x`	A list or a vector.
`.at`	A character vector (names), a numeric vector (positions), a symbol or or a list generated by `tidyselect` select helpers.

Value

A list or a vector.

See Also

`purrr::keep`

Examples

```
x <- c("First" = 1, "Second" = 2, "Last" = 3)
keep_at(x, "Second")
keep_at(x, Second)
keep_at(x, 2)
keep_at(x, starts_with("Sec"))
#> Second
#>      2

keep_at(x, ends_with("t"))
#> First  Last
#>     1     3

x <- c(1, 2, 3)
discard_at(x, 1)
#> Second   Last
#>      2      3
```

keep_if_in *Keep elements present in x and not contained in y*

Description

Unlike intersect, it does not remove duplicates in x and keeps its order.

Usage

```
keep_if_in(x, y)

x %if_in% y
```

Arguments

x	Source vector.
y	Destination vector (of the same mode as x).

Value

A filtered version of x.

See Also

keep_if_not_in

Examples

```
keep_if_in(1:5, 3:6)
# returns [3, 4, 5]

keep_if_in(c(4, 3, 4, 3, 1), 3:6)
# returns [4 3 4 3]
```

keep_if_not_in *Discard elements present in x and not contained in y*

Description

Unlike setdiff, it does not remove duplicates in x and keeps its order.

Usage

```
keep_if_not_in(x, y)

x %if_not_in% y
```

Arguments

`x`	Source vector.
`y`	Destination vector (of the same mode as x).

Value

A filtered version of x.

See Also

`keep_if_in`

Examples

```
keep_if_not_in(1:5, 3:6)
# returns [1 2]

keep_if_not_in(c(4, 3, 4, 3, 1), 3:6)
# returns [1]
```

`now_utc` *Current time in UTC time zone*

Description

Returns a vector with the current date and time in the UTC time zone.

Usage

```
now_utc(length = 1L)
```

Arguments

`length`	Positive integer (scalar) indicating the length of the returned vector. If length is a vector of multiple elements, only the first element is taken into account.

Value

A POSIXct vector of size `length` with the `tzone` attribute set to "UTC".

See Also

`Sys.time`, `lubridate::now`

Examples

```
now_utc(0)
# returns "POSIXct of length 0"
```

`pull_with_names` *Pull out a single column*

Description

Pull out a single column by using its name or its position and name the obtained vector using values from another column.

Usage

```
pull_with_names(.data, var = -1, name_col)
```

Arguments

`.data`	A data frame
`var`	The name of the column of interest, or a positive integer, giving the position counting from the left, or a negative integer, giving the position counting from the right. This argument supports tidyeval.
`name_col`	The column whose values will be used to name the pulled column. This argument supports tidyeval.

Value

A named vector.

Examples

```
head(pull_with_names(iris, 4, "Species"))
```

`seq_nrow` *Creates a sequence based on the number of rows or columns*

Description

Creates a sequence from 1 to the number of row or columns, respectively.

Usage

```
seq_nrow(x)

seq_ncol(x)
```

Arguments

`x`	a data frame or a matrix

Value

a vector of integers

See Also

`seq`

`setequal_na` *Check if two vectors have the same elements*

Description

Wrapper around `setequal` that adds extra parameter `na.rm`.

Usage

```
setequal_na(x, y, na.rm = FALSE)
```

Arguments

`x, y`	Vectors (of the same mode) containing a sequence of items.
`na.rm`	Boolean value indicating whether `NA` should be omitted or not.

Value

A logical scalar that states the result.

Examples

```
setequal_na(c(2, 1, 3), c(1, 2, 3))
# returns TRUE

setequal_na(c(1, NA, 3), c(3, NA, 1), na.rm = TRUE)
# returns TRUE

setequal_na(c(NA, NA), c(NA), na.rm = TRUE)
# returns TRUE

setequal_na(c(NA, NA), c(NA))
# returns TRUE

setequal_na(c(1, 2, 3), c(1, 2, 3, NA))
# returns FALSE
```

str1 — *High level overview of the structure of an R object*

Description

`str1()` is a wrapper around `str` which sets maximal level of nesting to 1, while `str2()` sets maximal level of nesting to 2.

Usage

```
str1(x)

str2(x)
```

Arguments

`x`	An R object

Value

Does not return anything.

See Also

`str`

Index