

# Package ‘Nmisc’

May 7, 2026

**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

## Contents

catn . . . . .	2
clear_warnings . . . . .	3
format_utc . . . . .	3
generate_install_file . . . . .	4
get_os . . . . .	5
get_packages . . . . .	5
is.POSIXct . . . . .	6
keep_at . . . . .	7
keep_if_in . . . . .	8
keep_if_not_in . . . . .	8
now_utc . . . . .	9
pull_with_names . . . . .	10
seq_nrow . . . . .	10
setequal_na . . . . .	11
str1 . . . . .	12
<b>Index</b>	<b>13</b>

---

catn	<i>Concatenate with new line</i>
------	----------------------------------

---

### 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	<i>Avoid repeated warnings</i>
----------------	--------------------------------

---

**Description**

Clear warnings for production code.

**Usage**

```
clear_warnings()
```

**See Also**

[warnings](#)

---

format_utc	<i>Format Date and POSIXct</i>
------------	--------------------------------

---

**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

<code>file</code>	The name of the file to be created.
<code>package_df</code>	A data frame obtained with <code>get_packages</code> that contains information regarding the name, version and source of the package.
<code>include_core_package</code>	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	<i>Returns the name of the Operating System</i>
--------	---

---

**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	<i>Get information about the package used in the project</i>
--------------	--

---

**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 <code>include_pattern</code>
include_pattern	A string representing a regex that matches project files in which to look for packages. By default, <code>get_packages</code> looks for all <code>.R</code> and <code>.Rmd</code> files in the current project.
exclude_pattern	A string representing a regex that matches project files to exclude. By default, <code>get_packages</code> 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:

<code>package_name</code>	The name of the package
<code>requested_by</code>	The context in which the package was used
<code>is_base</code>	Whether package is part of the core R packages
<code>source</code>	The source from which the package was installed
<code>version</code>	The version of the package, if installed locally
<code>is_installed</code>	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	<i>Keep or discard elements</i>
---------	---------------------------------

---

**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 <a href="#">tidyselect</a> 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	<i>Keep elements present in x and not contained in y</i>
------------	--

---

**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	<i>Discard elements present in x and not contained in y</i>
----------------	---

---

**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	<i>Current time in UTC time zone</i>
---------	--------------------------------------

---

**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	<i>Pull out a single column</i>
-----------------	---------------------------------

---

**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	<i>Creates a sequence based on the number of rows or columns</i>
----------	--

---

**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	<i>Check if two vectors have the same elements</i>
-------------	--

---

**Description**

Wrapper around [setequal](#) that adds extra parameter `na.rm`.

**Usage**

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

**Arguments**

<code>x, y</code>	Vectors (of the same mode) containing a sequence of items.
<code>na.rm</code>	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

`%if_in%` (`keep_if_in`), 8  
`%if_not_in%` (`keep_if_not_in`), 8

`cat`, 2  
`catn`, 2  
`clear_warnings`, 3

`discard_at` (`keep_at`), 7

`format.Date`, 3  
`format.POSIXct`, 3  
`format_utc`, 3

`generate_install_file`, 4, 6  
`get_os`, 5  
`get_packages`, 4, 5

`intersect`, 8  
`is.POSIXct`, 6

`keep_at`, 7  
`keep_if_in`, 8, 9  
`keep_if_not_in`, 8, 8

`lubridate::is.POSIXct`, 7  
`lubridate::now`, 9

`now_utc`, 9

`pull_with_names`, 10  
`purrr::keep`, 7

`seq`, 11  
`seq_ncol` (`seq_nrow`), 10  
`seq_nrow`, 10  
`setdiff`, 8  
`setequal`, 11  
`setequal_na`, 11  
`str`, 12  
`str1`, 12  
`str2` (`str1`), 12

`Sys.time`, 9  
`tidyselect`, 7  
`warnings`, 3