Package 'thisutils'

June 17, 2025

Type Package

Title Collection of Utility Functions for Data Analysis and Computing

Version 0.0.4

Date 2025-06-14

Maintainer Meng Xu <mengxu98@qq.com>

Description Provides a collection of utility functions commonly used in data analysis and scientific computing. Includes functions for parallel processing, matrix operations, correlation calculations, and other computational tasks to streamline R workflows.

License MIT + file LICENSE

URL https://github.com/mengxu98/thisutils

BugReports https://github.com/mengxu98/thisutils/issues

Depends R (>= 4.1.0)

Imports cli, doParallel, foreach, Matrix, methods, parallel, purrr, Rcpp, RcppArmadillo, RcppParallel, stats

LinkingTo Rcpp, RcppArmadillo, RcppParallel

Config/Needs/website mengxu98/mxtemplate

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

Language en-US

NeedsCompilation yes

Author Meng Xu [aut, cre] (ORCID: <https://orcid.org/0000-0002-8300-1054>)

Repository CRAN

Date/Publication 2025-06-17 06:20:02 UTC

Contents

`thisutils-package` *thisutils: An R package for utility functions.*

Description

An R package for utility functions.

Author(s)

Meng Xu (Maintainer), <mengxu98@qq.com>

Source

https://mengxu98.github.io/thisutils/

See Also

Useful links:

- https://github.com/mengxu98/thisutils
- Report bugs at https://github.com/mengxu98/thisutils/issues

add_pkg_file *Add package file*

Description

Automatically generate a file containing functions and related code for R package development.

Usage

```
add_pkg_file(
  desc_file,
  pkg_name = NULL,
  pkg_description = NULL,
  author_name = NULL,
  author_email = NULL,
  github_url = NULL,
  output_dir = NULL,
  use_figlet = TRUE,
  figlet_font = "Slant",
 colors = c("red", "yellow", "green", "magenta", "cyan", "yellow", "green", "white",
    "magenta", "cyan"),
  unicode = TRUE,
  verbose = TRUE
)
```

Arguments

desc_file	The DESCRIPTION file. Must be provided, it will be used to extract package information. Using `add_pkg_file("DESCRIPTION")`, will be created <pkg_name>-package.R based on the DESCRIPTION file. If you want to use some specific information, such as `author_name` or `author_email`, you can provide them manually.
pkg_name	Character string, the name of the package. Default is NULL, which will be read from DESCRIPTION file.
pkg_description	Character string, short description of the package. Default is NULL, which will be read from DESCRIPTION file.
author_name	Character string, name of the package author. Default is NULL, which will be read from DESCRIPTION file.
author_email	Character string, email of the package author. Default is NULL, which will be read from DESCRIPTION file.
github_url	Character string, GitHub URL of the package. Default is NULL, which will be read from DESCRIPTION file or constructed based on package name.
output_dir	Character string, directory where to save the package file. Default is NULL, you should specify it, such as 'R/'.

use_figlet	Logical, whether to use figlet for ASCII art generation. Default is TRUE.
figlet_font	Character string, figlet font to use. Default is "Slant".
colors	Character vector, colors to use for the logo elements.
unicode	Logical, whether to use Unicode symbols. Default is TRUE.
verbose	Logical, whether to print progress messages. Default is TRUE.

Value

Creates a file in specified output directory

---

as_matrix *Convert sparse matrix into dense matrix*

Description

Convert sparse matrix into dense matrix

Usage

```
as_matrix(x, parallel = FALSE, sparse = FALSE)
```

Arguments

x	A matrix.
parallel	Logical value, default is FALSE. Setting to parallelize the computation with RcppParallel::setThreadOptions.
sparse	Logical value, default is FALSE, whether to output a sparse matrix.

Value

A dense or sparse matrix

Examples

```
m <- simulate_sparse_matrix(
  1000, 1000,
  decimal = 3
)

system.time(
  a <- as.matrix(m)
)
system.time(
  b <- as_matrix(m)
)
system.time(
  c <- as_matrix(m, parallel = TRUE)
```

```
)
system.time(
  d <- as_matrix(m, sparse = TRUE)
)

m[1:5, 1:5]
a[1:5, 1:5]
b[1:5, 1:5]
c[1:5, 1:5]

identical(a, b)
identical(a, c)
identical(b, c)
identical(a, d)
identical(b, d)
```

`check_sparsity` *Check sparsity of matrix*

Description

Check sparsity of matrix

Usage

```
check_sparsity(x)
```

Arguments

`x`	A matrix.

Value

Sparsity of matrix

`figlet` *figlet*

Description

Create ASCII art text using figlet fonts.

Usage

```
figlet(
  text,
  font = "Slant",
  width = getOption("width", 80),
  justify = "left",
  absolute = FALSE,
  strip = TRUE
)
```

Arguments

text	Text to make bigger
font	Name of font, path to font, or 'figlet_font' object
width	Width to use when justifying and breaking lines
justify	Text justification to use in rendering ("left", "centre", "right")
absolute	Logical, indicating if alignment is absolute
strip	Logical, indicating if whitespace should be removed

Value

An object of class 'figlet_text' which is a character vector with a handy print method

References

http://www.figlet.org/ https://github.com/richfitz/rfiglet https://github.com/jbkunst/figletr

Examples

```
figlet("thisutils")
```

---

figlet_font *Get a figlet font*

Description

Get a figlet font

Usage

```
figlet_font(font)
```

Arguments

font	Path or name of the font to load

Value

A ‘figlet_font‘ object for use with [figlet]

`list_figlet_fonts` *List available figlet fonts*

Description

List all figlet font files available in the package or system.

Usage

```
list_figlet_fonts()
```

Value

Character vector of available font names.

Examples

```
list_figlet_fonts()
```

`log_message` *Print diagnostic message*

Description

Integrate the message printing function with the `cli` package, and the `message` function. The message could be suppressed by `suppressMessages`.

Usage

```
log_message(
  ...,
  verbose = TRUE,
  message_type = c("info", "success", "warning", "error"),
  cli_model = TRUE,
  timestamp = TRUE,
  level = 1,
  level_symbol = "  "
)
```

Arguments

...	Text to print.
verbose	Logical value, default is TRUE. Whether to print the message.
message_type	Type of message, default is info. Could be choose one of info, success, warning, and error.
cli_model	Logical value, default is TRUE. Whether to use the cli package to print the message.
timestamp	Logical value, default is TRUE. Whether to show the current time in the message.
level	Integer value, default is 1. The level of the message, which affects the indentation. Level 1 has no indentation, higher levels add more indentation.
level_symbol	Character value, default is " " (two spaces). The symbol used for indentation at each level.

Value

Formated message printed to the console.

Examples

```
log_message("Hello, ", "world!")
log_message("Hello, world!", timestamp = FALSE)
log_message("Hello, ", "world!", message_type = "success")
log_message("Hello, world!", message_type = "warning")
suppressMessages(log_message("Hello, ", "world!"))
log_message("Hello, world!", verbose = FALSE)
log_message("Hello, world!", level = 2)
log_message("Hello, world!", level = 3, level_symbol = "->")
```

---

normalization *Normalize numeric vector*

Description

Normalize numeric vector

Usage

```
normalization(x, method = "max_min", na_rm = TRUE, ...)
```

Arguments

x	Input numeric vector.
method	Method used for normalization.
na_rm	Whether to remove NA values, and if setting TRUE, using 0 instead.
...	Parameters for other methods.

Value

Normalized numeric vector

Examples

```
nums <- c(runif(2), NA, -runif(2))
nums
normalization(nums, method = "max_min")
normalization(nums, method = "maximum")
normalization(nums, method = "sum")
normalization(nums, method = "softmax")
normalization(nums, method = "z_score")
normalization(nums, method = "mad")
normalization(nums, method = "unit_vector")
normalization(nums, method = "unit_vector", na_rm = FALSE)
```

---

parallelize_fun *Parallelize a function*

Description

Parallelize a function

Usage

```
parallelize_fun(x, fun, cores = 1, export_fun = NULL, verbose = TRUE)
```

Arguments

x	A vector or list to apply over.
fun	The function to be applied to each element.
cores	The number of cores to use for parallelization with foreach. Default is 1.
export_fun	The functions to export the function to workers.
verbose	Logical value, default is TRUE. Whether to print progress messages.

Value

A list of computed results

Examples

```
parallelize_fun(1:3, function(x) x^2)
parallelize_fun(list(1, 2, 3), function(x) x^2)
```

pearson_correlation *Correlation and covariance calculation for sparse matrix*

Description

Correlation and covariance calculation for sparse matrix

Usage

```
pearson_correlation(x, y = NULL)
```

Arguments

x	Sparse matrix or character vector.
y	Sparse matrix or character vector.

Value

A list with covariance and correlation matrices.

Examples

```
m1 <- simulate_sparse_matrix(
  100, 100
)
m2 <- simulate_sparse_matrix(
  100, 100,
  sparsity = 0.05
)
a <- pearson_correlation(m1, m2)
a$cov[1:5, 1:5]
a$cor[1:5, 1:5]
```

print.logo *print logo*

Description

print logo

Usage

```
## S3 method for class 'logo'
print(x, ...)
```

Arguments

x	Input information.
...	Other parameters.

Value

Print the ASCII logo

`r_square` *coefficient of determination (R^2)*

Description

coefficient of determination (R^2)

Usage

```
r_square(y_true, y_pred)
```

Arguments

y_true	A numeric vector with ground truth values.
y_pred	A numeric vector with predicted values.

Value

R^2 value

Examples

```
y <- rnorm(100)
y_pred <- y + rnorm(100, sd = 0.5)
r_square(y, y_pred)
```

`simulate_sparse_matrix` *Generate a simulated sparse matrix*

Description

This function generates a sparse matrix with a specified number of rows and columns, a given sparsity level, and a distribution function for the non-zero values.

Usage

```
simulate_sparse_matrix(
  nrow,
  ncol,
  sparsity = 0.95,
  distribution_fun = function(n) stats::rpois(n, lambda = 0.5) + 1,
  decimal = 0,
  seed = 1
)
```

Arguments

`nrow`	Number of rows in the matrix.
`ncol`	Number of columns in the matrix.
`sparsity`	Proportion of zero elements (sparsity level). Default is 0.95, meaning 95% of elements are zero (5% are non-zero).
`distribution_fun`	Function to generate non-zero values.
`decimal`	Numeric value, default is `0`. Controls the number of decimal places in the generated values. If set to `0`, values will be integers. When decimal > 0, random decimal parts are uniformly distributed across the full range.
`seed`	Random seed for reproducibility.

Value

A sparse matrix of class "dgCMatrix"

Examples

```
simulate_sparse_matrix(1000, 500) |>
  check_sparsity()

simulate_sparse_matrix(10, 10, decimal = 1)
simulate_sparse_matrix(10, 10, decimal = 5)
```

`sparse_cor` *A sparse correlation function*

Description

Safe correlation function which returns a sparse matrix without missing values

Usage

```
sparse_cor(
  x,
  y = NULL,
  method = "pearson",
  allow_neg = TRUE,
  remove_na = TRUE,
  remove_inf = TRUE,
  ...
)
```

Arguments

`x`	Sparse matrix or character vector.
`y`	Sparse matrix or character vector.
`method`	Method to use for calculating the correlation coefficient.
`allow_neg`	Logical. Whether to allow negative values or set them to 0.
`remove_na`	Logical. Whether to replace NA values with 0.
`remove_inf`	Logical. Whether to replace infinite values with 1.
`...`	Other arguments passed to stats::cor function.

Value

A correlation matrix.

Examples

```
m1 <- simulate_sparse_matrix(
  500, 100
)
m2 <- simulate_sparse_matrix(
  500, 100,
  seed = 2025
)
a <- sparse_cor(m1)
b <- sparse_cor(m1, m2)
c <- as_matrix(
  cor(as_matrix(m1)),
```

```
  sparse = TRUE
)
d <- as_matrix(
  cor(as_matrix(m1), as_matrix(m2)),
  sparse = TRUE
)

a[1:5, 1:5]
c[1:5, 1:5]
all.equal(a, c)

b[1:5, 1:5]
d[1:5, 1:5]
all.equal(b, d)

m1[sample(1:500, 10)] <- NA
m2[sample(1:500, 10)] <- NA

sparse_cor(m1, m2)[1:5, 1:5]

system.time(
  sparse_cor(m1)
)
system.time(
  cor(as_matrix(m1))
)

system.time(
  sparse_cor(m1, m2)
)
system.time(
  cor(as_matrix(m1), as_matrix(m2))
)
```

`split_indices` *Split indices.*

Description

An optimised version of split for the special case of splitting row indices into groups.

Usage

```
split_indices(group, n = 0L)
```

Arguments

`group`	Integer indices
`n`	The largest integer (may not appear in index). This is hint: if the largest value of `group` is bigger than `n`, the output will silently expand.

Value

A list of vectors of indices.

References

https://github.com/hadley/plyr/blob/d57f9377eb5d56107ba3136775f2f0f005f33aa3/src/split-numeric.cpp#L20

Examples

```
split_indices(sample(10, 100, rep = TRUE))
split_indices(sample(10, 100, rep = TRUE), 10)
```

---

thisutils_logo *thisutils logo*

Description

The thisutils logo, using ASCII or Unicode characters Use cli::ansi_strip to get rid of the colors.

Usage

```
thisutils_logo(unicode = cli::is_utf8_output())
```

Arguments

unicode	Unicode symbols. Default is TRUE on UTF-8 platforms.

Value

A logo with ASCII formatted text

References

https://github.com/tidyverse/tidyverse/blob/main/R/logo.R

Examples

```
thisutils_logo()
```

`%ss%` *Value selection operator*

Description

This operator returns the left side if it's not NULL, otherwise it returns the right side.

Usage

```
a %ss% b
```

Arguments

a	The left side value to check
b	The right side value to use if a is NULL

Value

a if it is not NULL, otherwise b

Examples

```
NULL %ss% 10
5 %ss% 10
```

Index