Package 'vvmover'

July 22, 2025

Title Read and Write Data

Version 1.6.0

Description Offers a wide range of functions for reading and writing data in various file formats, including CSV, RDS, Excel and ZIP files.
Additionally, it provides functions for retrieving metadata associated with files, such as file size and creation date, making it easy to manage and organize large data sets.
This package is designed to simplify data import and export tasks, and provide users with a comprehensive set of tools to work with different types of data files.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

URL https://vusaverse.github.io/vvmover/, https://github.com/vusaverse/vvmover

BugReports https://github.com/vusaverse/vvmover/issues

Imports dplyr, magrittr, readr, readxl, utils

NeedsCompilation no

Author Tomer Iwan [aut, cre, cph]

Maintainer Tomer Iwan <t.iwan@vu.nl>

Repository CRAN

Date/Publication 2024-02-26 13:30:02 UTC

Contents

`add_current_datetime_column` *Add current date and time to data frame.*

Description

This function adds a new column to a data frame with the current date and time. The name of the new column is a combination of the provided prefix, stage, and "Date_time". If the new column already exists, it will be overwritten.

Usage

```
add_current_datetime_column(data)
```

Arguments

`data`	Data frame.

Value

Data frame with an additional column containing the current date and time.

Examples

```
## Not run:
# Create a sample data frame
data <- data.frame(a = 1:5, b = letters[1:5])

# Add date to file name
add_current_datetime_column(data)

## End(Not run)
```

`calculate_na_percentage` *Calculate NA Percentage*

Description

This function calculates the percentage of NA values in a given vector. It also includes a margin of 10%.

Usage

```
calculate_na_percentage(x, ...)
```

Arguments

`x`	A numeric vector.
`...`	Additional arguments (not used).

Value

A numeric value representing the percentage of NA values in the vector.

`check_installed_package` *check_installed_package*

Description

Check if a package is installed. If not, throw an error message

Usage

```
check_installed_package(package_name, check = FALSE)
```

Arguments

`package_name`	the name of the package (quoted)
`check`	the function should work as a boolean operator

Value

Boolean value whether package is installed.

Examples

```
check_installed_package("dplyr")
```

get_last_modifed_date *get_last_modified_date*

Description

get_last_modified_date

Usage

```
get_last_modifed_date(file_path)
```

Arguments

file_path	Path to the file.

Value

Date that file was last modified

get_recent_file *Get recent file*

Description

Is a wrapper around get_recent_file_date_filename_ymd en get_recent_file_date_modified and retrieves the most recent version of a file based on naming or date modified.

Usage

```
get_recent_file(path, match, date_type = "modified")
```

Arguments

path	The path to search for the file
match	The search term matched in the file name
date_type	The way to find the recent file date_type = "modified" is based on customization, date_type = "filename_ymd" is based on file name.

Value

The most recent file.

See Also

Other Get recent files: `get_recent_file_date_filename_ymd()`, `get_recent_file_date_modified()`

`get_recent_file_date_filename_ymd`
Get recent file date filename ymd

Description

This function determines the path of the most recent version of a file in a folder. Sorting is determined by file name where it is a condition that the filename starts with ymd encoding.

Usage

```
get_recent_file_date_filename_ymd(path, match)
```

Arguments

`path`	The path to search for the file.
`match`	The search term to match in the file name.

Value

The most recent file

See Also

Other Get recent files: `get_recent_file_date_modified()`, `get_recent_file()`

`get_recent_file_date_modified`
Get recent file date modified This function determines the path of the most recent version of a file in a folder. The sorting is determined based on the date of the last change.

Description

Get recent file date modified This function determines the path of the most recent version of a file in a folder. The sorting is determined based on the date of the last change.

Usage

```
get_recent_file_date_modified(path, match, echo = TRUE)
```

Arguments

`path`	The path to search for the file.
`match`	The search term to match in the file name.
`echo`	Print the date the file was last modified in the console.

Value

The most recent file.

See Also

Other Get recent files: `get_recent_file_date_filename_ymd()`, `get_recent_file()`

`is_field_subset` *Check if Field is Subset*

Description

This function checks if a field is a subset. We define a subset as a character field that contains fewer than 20 unique values.

Usage

```
is_field_subset(field_name, df, column_names, column_types)
```

Arguments

`field_name`	Name of the field to check.
`df`	Dataframe to check.
`column_names`	Set that the field is part of.
`column_types`	Types that belong to the column names.

`move_files_pattern` *move files pattern*

Description

Move files in a directory based on regular expression

Usage

```
move_files_pattern(Folder_origin, Folder_dest, pattern, recursive = FALSE)
```

Arguments

`Folder_origin`	Source folder.
`Folder_dest`	Destination folder.
`pattern`	Pattern to match files in source folder on.
`recursive`	Default: FALSE. Whether to use recursive search in directory.

Value

message

print_last_modified *Print date the file was last modified*

Description

This function prints the date a file was last modified

Usage

```
print_last_modified(path)
```

Arguments

path	The file path

Value

message

Examples

```
print_last_modified(readr::readr_example("mtcars.csv"))
```

random_string_vector *Generate a random string vector*

Description

Generate a random string vector

Usage

```
random_string_vector(
  n = 500,
  length = 6,
  characters = c(letters, LETTERS, 0:9)
)
```

Arguments

n	The number of items in the vector. Default is set to 500.
length	the number of characters in a string. Default is set to 6.
characters	A vector containing the characters to include. Default is all lowercase, all, uppercase letters and all numbers.

read_excel_allsheets *Read Excel Allsheets*

Description

Read in all sheets in an Excel file.

Usage

```
read_excel_allsheets(filename)
```

Arguments

filename	Name of Excel file

Value

Dataframe

Examples

```
read_excel_allsheets(readxl::readxl_example("clippy.xls"))
```

save_session_info_to_file *Save session info to a file*

Description

Stores session info in a .txt file.

Usage

```
save_session_info_to_file(path)
```

Arguments

path	The directory path where the session info file will be saved.

Value

A .txt file containing session info, saved at the specified path.

`unzip_read_delim` *Extract a .zip archive and read in with read_delim*

Description

Extract a .zip archive and read it in with readr's read_delim function. The file is extracted to a temporary location, and then deleted after reading it.

Usage

```
unzip_read_delim(zip_path, filename = NULL, ...)
```

Arguments

`zip_path`	The file path of the .zip archive
`filename`	OPTIONAL: The file name of the file in the .zip archive to be read. This parameter can be left empty if there is only 1 file in the archive.
`...`	arguments to the readr::read_delim function. see: read_delim

Value

Dataframe

Examples

```
unzip_read_delim(readr::readr_example("mtcars.csv.zip"))
```

Index