

Package ‘edm1’

June 20, 2024

Title Simplify Complex Data Manipulation
Version 2.0.0.0
Description Provides complex sorting algorithms. Provides date manipulation algorithms. In addition to providing handy functions to discretize variables, an SQL joins alternatives, a set of function to work with geographical coordinates, and other functions to work with text mining.
License GPL (==3)
Encoding UTF-8
Roxygen list(markdown = TRUE)
RoxygenNote 7.3.1
Imports stringr,
 stringi,
 dplyr,
 openxlsx

Contents

fold_rec	1
fold_rec2	2
get_rec	2
list_files	3
see_file	3
see_inside	4
Index	5

fold_rec	<i>fold_rec</i>
----------	-----------------

Description

Allow to get all the files recursively from a path according to an end and start depth value. If you want to have an other version of this function that uses a more sophisticated algorithm (which can be faster), check file_rec2. Depth example: if i have dir/dir2/dir3, dir/dir2b/dir3b, i have a depth equal to 3

Usage

```
fold_rec(xmax, xmin = 1, pathc = ".")
```

Arguments

xmax	is the end depth value
xmin	is the start depth value
pathc	is the reference path

fold_rec2

fold_rec2

Description

Allow to find the directories and the subdirectories with a specified end and start depth value from a path. This function might be more powerfull than file_rec because it uses a custom algorythm that does not nee to perform a full recursive search before tuning it to only find the directories with a good value of depth. Depth example: if i have dir/dir2/dir3, dir/dir2b/dir3b, i have a depth equal to 3

Usage

```
fold_rec2(xmax, xmin = 1, pathc = ".")
```

Arguments

xmax	is the depth value
xmin	is the minimum value of depth
pathc	is the reference path, from which depth value is equal to 1

get_rec

get_rec

Description

Allow to get the value of directorie depth from a path.

Usage

```
get_rec(pathc = ".")
```

Arguments

pathc	is the reference path example: if i have dir/dir2/dir3, dir/dir2b/dir3b, i have a depth equal to 3
-------	--

list_files	<i>list_files</i>
------------	-------------------

Description

A list.files() based function addressing the need of listing the files with extension a or or extension b ...

Usage

```
list_files(patternc, pathc = ".")
```

Arguments

patternc	is a vector containing all the extensions you want
pathc	is the path, can be a vector of multiple path because list.files() supports it.

see_file	<i>see_file</i>
----------	-----------------

Description

Allow to get the filename or its extension

Usage

```
see_file(string_, index_ext = 1, ext = TRUE)
```

Arguments

string_	is the input string
index_ext	is the occurrence of the dot that separates the filename and its extension
ext	is a boolean that if set to TRUE, will return the file extension and if set to FALSE, will return filename

Examples

```
print(see_file(string_="file.abc.xyz"))  
#[1] ".abc.xyz"  
print(see_file(string_="file.abc.xyz", ext=FALSE))  
#[1] "file"  
print(see_file(string_="file.abc.xyz", index_ext=2))  
#[1] ".xyz"
```

see_inside

see_inside

Description

Return a list containing all the column of the files in the current directory with a chosen file extension and its associated file and sheet if `xlsx`. For example if i have 2 files "out.csv" with 2 columns and "out.xlsx" with 1 column for its first sheet and 2 for its second one, the return will look like this: `c(column_1, column_2, column_3, column_4, column_5, unique_separator, "1-2-out.csv", "3-3-sheet_1-out.xlsx", 4-5-sheet_2-out.xlsx)`

Usage

```
see_inside(
  pattern_,
  path_ = ".",
  sep_ = c(", "),
  unique_sep = "####",
  rec = FALSE
)
```

Arguments

<code>pattern_</code>	is a vector containin the file extension of the spreadsheets (" <code>xlsx</code> ", " <code>csv</code> "...)
<code>path_</code>	is the path where are located the files
<code>sep_</code>	is a vector containing the separator for each csv type file in order following the operating system file order, if the vector does not match the number of the csv files found, it will assume the separator for the rest of the files is the same as the last csv file found. It means that if you know the separator is the same for all the csv type files, you just have to put the separator once in the vector.
<code>unique_sep</code>	is a pattern that you know will never be in your input files
<code>rec</code>	is a boolean allows to get files recursively if set to <code>TRUE</code> , defaults to <code>TRUE</code> If <code>x</code> is the return value, to see all the files name, position of the columns and possible sheet name associated with, do the following:

Index

fold_rec, [1](#)
fold_rec2, [2](#)

get_rec, [2](#)

list_files, [3](#)

see_file, [3](#)
see_inside, [4](#)