

R documentation

of 'all_data_to_upper.Rd' etc.

December 21, 2016

all_data_to_upper	<i>Converts column names to upper case for a given list of datasets</i>
-------------------	---

Description

Converts column names to upper case for a given list of datasets

Usage

```
all_data_to_upper(list_of_loaded_datasets)
```

Arguments

list_of_loaded_datasets
A list of loaded datasets

Value

Changes the datasets in place, with column names converted to upper case

Examples

```
## Not run:
setwd("path/to/datasets/")
list_of_datasets <- ls(pattern = "expr") # "expr" is a character string common to
every data set (for example, suppose all you loaded data sets are of the form *_entrep_*)
list_of_loaded_datasets <- read_list(list_of_datasets, read_func = read.csv)
updated_list <- all_data_to_upper(list_of_loaded_datasets)

## End(Not run)
```

%ni%	<c2><ab>Not in<c2><bb> operator. Taken from a stackoverflow answer by user 'baptiste'
------	---

Description

<c2><ab>Not in<c2><bb> operator. Taken from a stackoverflow answer by user 'baptiste'

Usage

... %ni% NA

Arguments

- x An object
- y A list of objects

Value

True or False

Examples

3 %ni% seq(5, 10)

map_filter	<i>Filter a list of conditions</i>
------------	------------------------------------

Description

Filter a list of conditions

Usage

map_filter(dataset, variable, list_values)

Arguments

- dataset A 'data.frame' object
- variable The variable to filter
- list_values An atomic vector giving the conditions we want to filter by

Value

A list containing the filtered 'data.frame's.

Examples

```
data(mtcars)
map_filter(mtcars, "cyl", c(4, 6))
```

modal_value	Returns the mode of a vector
-------------	------------------------------

Description

Returns the mode of a vector

Usage

```
modal_value(x, na.rm = FALSE)
```

Arguments

x	A list of values
na.rm	Should NAs be removed? FALSE by default

Value

Returns the mode of x, either a numeric if x is a list of numerics, or a character if x is a list of characters

Examples

```
x <- c(3,3,3,4,5,NA, NA, NA, NA, NA)
modal_value(x, na.rm = TRUE)
```

multi_join	Merges more than two datasets together using 'Reduce()'
------------	---

Description

This function is useful to merge a lot of datasets together. It uses 'dplyr' join functions internally. Inspired by: <http://novicemetrics.blogspot.lu/2011/04/merging-multiple-data-files-into-one.html>

Usage

```
multi_join(list_of_loaded_data, join_func, ...)
```

Arguments

join_func A function, the join function to use to join the data together
 ... Further parameters to pass to the join function
 list_of_loaded_datasets
 A list of loaded datasets

Value

Returns the merged datasets

Examples

```
## Not run:
setwd("path/to/datasets/")
list_of_datasets <- ls(pattern = "expr") # "expr" is a character string common to
every data set (for example, suppose all you loaded data sets are of the form *_entrep_*)
list_of_loaded_datasets <- read_list(list_of_datasets, read_func = read.csv)
merged_data <- multi_join(list_of_loaded_datasets, full_join, by = "some_var")
If you have a list of lists of datasets, you can merge the lists like this:
lapply(hu, multi_join, read_csv, full_join)

## End(Not run)
```

one_row	<i>Keep only one row per individual. Only keep the row with the least amount of NA in every column</i>
---------	--

Description

Keep only one row per individual. Only keep the row with the least amount of NA in every column

Usage

```
one_row(dataframe, list_var)
```

Arguments

dataframe A 'data.frame' object
 list_var An atomic vector of characters giving the variable combinations to keep

Value

A 'data.frame' object with only one line per individual

Examples

```
dataframe <- as.data.frame(list("id" = c(rep(1, 3), rep(2, 4)), "x1" = rnorm(7)))
new_df <- one_row(dataframe, "id")
```

read_list	<i>Reads a list of datasets</i>
-----------	---------------------------------

Description

Reads a list of datasets

Usage

```
read_list(list_of_datasets, read_func, ...)
```

Arguments

list_of_datasets	A list of datasets (names of datasets are strings)
read_func	A function, the read function to use to read the data
...	Further arguments passed to read_func

Value

Returns a list of the datasets

Examples

```
## Not run:  
setwd("path/to/datasets/")  
list_of_datasets <- list.files(pattern = "*.csv")  
list_of_loaded_datasets <- read_list(list_of_datasets, read_func = read.csv)  
  
## End(Not run)
```

to_map	<i>Generalizes functions to work on lists of data</i>
--------	---

Description

Generalizes functions to work on lists of data

Usage

```
to_map(func)
```

Arguments

func	A function
------	------------

Value

A function that works on lists of data

Examples

```
## Not run:
setwd("path/to/datasets/")
list_of_datasets <- list.files(pattern = "*.csv")
list_of_loaded_datasets <- read_list(list_of_datasets, read_func = read.csv)
# To get the summary statistics of all variables for all the datasets in the list:
summarymap <- to_map(summary)
summarymap(list_of_loaded_datasets)
# To get the summary statistics of a single variable for all the datasets in the list:
summarymap(list_of_loaded_datasets, "variable")
# This is equivalent to
purrr::map(list_of_loaded_datasets, (function(x) summary(x$variable)))

## End(Not run)
```

Index

`%ni%`, [2](#)

`all_data_to_upper`, [1](#)

`map_filter`, [2](#)

`modal_value`, [3](#)

`multi_join`, [3](#)

`one_row`, [4](#)

`read_list`, [5](#)

`to_map`, [5](#)