Package 'dsmisc'

October 13, 2022

Type Package

Title Data Science Box of Pandora Miscellaneous

Version 0.3.3

Date 2020-09-11

Description Tool collection for common and not so common data science use cases. This includes custom made algorithms for data management as well as value calculations that are hard to find elsewhere because of their specificity but would be a waste to get lost nonetheless. Currently available functionality: find sub-graphs in an edge list data.frame, find mode or modes in a vector of values, extract (a) specific regular expression group(s), generate ISO time stamps that play well with file names, or generate URL parameter lists by expanding value combinations.

License GPL (>= 2)

Imports Rcpp (>= 1.0.1), stringr

LinkingTo Rcpp

RoxygenNote 7.1.1

Encoding UTF-8

SystemRequirements C++11

Suggests covr, testthat, spelling

Language en-US

NeedsCompilation yes

Author Peter Meissner [aut, cre]

Maintainer Peter Meissner <retep.meissner@gmail.com>

Repository CRAN

Date/Publication 2020-09-12 05:40:13 UTC

R topics documented:

`df_defactorize` *df_defactorize*

Description

df_defactorize

Usage

```
df_defactorize(df)
```

Arguments

`df`	a data.frame like object

Value

returns the same data.frame except that factor columns have been transformed into character columns

Examples

```
df <-
  data.frame(
    a = 1:2,
    b = factor(c("a", "b")),
    c = as.character(letters[3:4]),
    stringsAsFactors = FALSE
 )
vapply(df, class, "")

df_df <- df_defactorize(df)
vapply(df_df, class, "")
```

`graphs_find_subgraphs` *Subgraphs in Undirected Graphs/Networks*

Description

Finding and indexing subgraphs in undirected graph.

Usage

```
graphs_find_subgraphs(id_1, id_2, verbose = 1L)
```

Arguments

`id_1`	vector of integers indicating ids
`id_2`	vector of integers indicating ids
`verbose`	in integer indicating the amount of verbosity; good for long running tasks or to get more information about the workings of the algorithm; currently accepted values: 0, 1, 2

Details

Input is given as two vectors where each pair of node ids ‘id_1[i]‘ - ‘id_2[i]‘ indicates an edge between two nodes.

Value

An integer vector with subgraph ids such that each distinct subgraph - i.e. all nodes are reachable within the graph and no node outside the subgraph is reachable - gets a distinct integer value. Integer values are assigned via

Examples

```
graphs_find_subgraphs(c(1,2,1,5,6,6), c(2,3,3,4,5,4), verbose = 0)
graphs_find_subgraphs(c(1,2,1,5,6,6), c(2,3,3,4,5,4), verbose = 2)
```

stats_mode *Mode*

Description

Function calculating the mode.

Usage

```
stats_mode(x, multimodal = FALSE, warn = TRUE)
```

Arguments

x	vector to get mode for
multimodal	wether or not all modes should be returned in case of more than one
warn	should the function warn about multimodal outcomes?

Value

vector of mode or modes

stats_mode_multi *Mode Allowing for Multi Modal Mode*

Description

Function calculating the mode, allowing for multiple modes in case of equal frequencies.

Usage

```
stats_mode_multi(x)
```

Arguments

x	vector to get mode for

Value

vector with all modes

str_group_extract *Extract Regular Expression Groups*

Description

Extract Regular Expression Groups

Usage

```
str_group_extract(string, pattern, group = NULL, nas = TRUE)
```

Arguments

string	string to extract from
pattern	pattern with groups to match
group	groups to extract
nas	return NA values (TRUE) or filter them out (FALSE)

Value

string vector or string matrix

Examples

```
strings <- paste(LETTERS, seq_along(LETTERS), sep = "_")
str_group_extract(strings, "([\\w])_(\\d+)")
str_group_extract(strings, "([\\w])_(\\d+)", 1)
str_group_extract(strings, "([\\w])_(\\d+)", 2)
```

time_stamp *Time Stamps for File Names*

Description

Generating file name ready iso time stamps.

Usage

```
time_stamp(ts = Sys.time(), sep = c("-", "_", "_"))
```

Arguments

ts	one or more POSIX time stamp
sep	separators to be used for formatting

Value

Returns timestamp string in format yyyy-mm-dd_HH_MM_SS ready to be used safely in file names on various operating systems.

Examples

```
time_stamp()
time_stamp( Sys.time() - 10000 )
```

`web_gen_param_list_expand` *URL Parameter Combinations*

Description

Generate URL parameter combinations from sets of parameter values.

Usage

```
web_gen_param_list_expand(..., sep_1 = "=", sep_2 = "&")
```

Arguments

`...`	multiple vectors passed on as named arguments or a single list or a data.frame
`sep_1`	first separator to use between key and value
`sep_2`	second separator to use between key-value pairs

Value

string vector with assembled query string parameter combinations

Examples

```
web_gen_param_list_expand(q = "beluga", lang = c("de", "en"))
```

Index