

Package: cpmtools (via r-universe)

August 17, 2024

Title Miscellaneous Functions by 'Colin Millar'

Version 0.0.0.9002

Description Miscellaneous functions.

Imports utils, TAF, formatR, sinew, git2r, jsonlite, glue, graphics,
stats

License GPL-3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.0

Repository <https://ices-tools-prod.r-universe.dev>

RemoteUrl <https://github.com/colinpmillar/cpmtools>

RemoteRef HEAD

RemoteSha 4a6b211299169d92c4f612a7921349631a30cec3

Contents

add.roxy	2
canvas	2
git_init	4
installed_packages	5
open_dir	6
quick_help	7
sourceTAF2	8
taf_gitignore	8
taf_roxy	9
update_r_win	10
write.taf.json	10
write_sam_upload	11

Index	12
--------------	-----------

add.roxy

add.roxy

Description

adds roxygen documentation headers to the top on R function and optionally write this to a file in the R folder.

Usage

```
add.roxy(obj, file = FALSE)
```

Arguments

obj	the function to create documentation for
file	if TRUE write to file (R/function name.R), Default: FALSE

Value

TRUE if successful, FALSE if not

See Also

[makeOxygen tidy_source](#)

Examples

```
## Not run:  
if(interactive()){  
  #EXAMPLE1  
}  
  
## End(Not run)
```

canvas*Paradox*

Description

Functions to make pleasing doodles

Usage

```

canvas()

paradox(
  shape,
  dist = 0.05,
  reverse = FALSE,
  cols = colorRampPalette(c("red", "blue"))(40),
  debug = FALSE
)

paradoxes(shapes, dist = 0.05, cols = colorRampPalette(c("red", "blue"))(40))

scale_shape(shape, scale = 1, shift = 0)

```

Arguments

shape	a matrix with columns x and y defining a shape to doodle within, can be any closed polygon, but some shapes work better than others.
dist	the distance up the adjacent edge where each new line will be drawn, default 0.05
reverse	should the shape be drawn in reverse order (see examples)
cols	a palate of colours as a vector of strings interpretable as colours
debug	should debug messages be produces (default FALSE)
shapes	a list of shapes (polygons) to doodle within
scale	if scaling, how much to scale by
shift	if scaling, should the shape also be shifted and by how much

Value

displays a plot

Examples

```

if (interactive()) {

  triangle <- cbind(x = c(0, 1, .7), y = c(0, 0.1, 1))
  triangle2 <- cbind(x = c(0, 0, .7), y = c(0, 1, 1))

  square <- cbind(x = c(0, 1, 1, 0), y = c(0, 0, 1, 1))

  canvas()
  paradox(square, 0.03)

  canvas()
  paradox(triangle, 0.042)
  paradox(triangle2, 0.042)
}

```

```

canvas()
paradox(scale_shape(square, c(0.5, 1)), 0.03)
paradox(scale_shape(square, c(0.5, 1), c(0.5, 0)), 0.05, reverse = TRUE)

canvas()
dist <- 0.02
paradox(scale_shape(square, .5), dist)
paradox(scale_shape(square, .5, c(0.5, 0)), dist, reverse = TRUE)
paradox(scale_shape(square, .5, .5), dist)
paradox(scale_shape(square, .5, c(0, 0.5)), dist, reverse = TRUE)

canvas()
dist <- 0.02
paradox(scale_shape(square, .5), dist, reverse = TRUE)
paradox(scale_shape(square, .5, c(0.5, 0)), dist, reverse = TRUE)
paradox(scale_shape(square, .5, .5), dist, reverse = TRUE)
paradox(scale_shape(square, .5, c(0, 0.5)), dist, reverse = TRUE)

canvas()
dist <- 0.02

cols1 <- colorRampPalette(c("red", "blue"))(30)
cols2 <- colorRampPalette(c("blue", "red"))(30)
cols3 <- colorRampPalette(c("green", "purple"))(30)
cols4 <- colorRampPalette(c("purple", "green"))(30)

paradox(scale_shape(square, .5), dist,
  reverse = TRUE,
  cols = c(rbind(cols1, cols2, cols3, cols4))
)
paradox(scale_shape(square, .5, c(0.5, 0)), dist,
  reverse = FALSE,
  cols = c(rbind(cols3, cols4, cols1, cols2))
)
paradox(scale_shape(square, .5, .5), dist,
  reverse = FALSE,
  cols = c(rbind(cols1, cols2, cols3, cols4))
)
paradox(scale_shape(square, .5, c(0, 0.5)), dist,
  reverse = TRUE,
  cols = c(rbind(cols3, cols4, cols1, cols2))
)
}

```

git_init

git_init

Description

FUNCTION_DESCRIPTION

Usage

```
git_init(path = ".", commit = TRUE)
```

Arguments

path	PARAM_DESCRIPTION, Default: ''
commit	PARAM_DESCRIPTION, Default: TRUE

Details

DETAILS

Value

no return value

See Also

[init,add,commit](#)

Examples

```
if (interactive()) {  
  # initialise a git repository and stage all files  
  git_init(commit = FALSE)  
}
```

installed_packages *installed_packages*

Description

FUNCTION_DESCRIPTION

Usage

```
installed_packages()
```

Details

DETAILS

Value

OUTPUT_DESCRIPTION

Examples

```
## Not run:
if(interactive()){
  #EXAMPLE1
}

## End(Not run)
```

open_dir	<i>open_dir</i>
----------	-----------------

Description

FUNCTION_DESCRIPTION

Usage

```
open_dir(dir = ".")
```

Arguments

dir PARAM_DESCRIPTION, Default: '.'

Details

DETAILS

Value

OUTPUT_DESCRIPTION

Examples

```
## Not run:
if(interactive()){
  #EXAMPLE1
}

## End(Not run)
```

quick_help	<i>quick_help</i>
------------	-------------------

Description

FUNCTION_DESCRIPTION

Usage

```
quick_help(person, project, path = getOption("cpmtools.quick_help_path"))
```

Arguments

person	PARAM_DESCRIPTION
project	PARAM_DESCRIPTION
path	PARAM_DESCRIPTION, Default: getOption("cpmtools.quick_help_path")

Details

DETAILS

Value

OUTPUT_DESCRIPTION

See Also

[taf.skeleton init,add,commit](#)

Examples

```
## Not run:  
if(interactive()){  
  #EXAMPLE1  
}  
  
## End(Not run)
```

sourceTAF2

sourceTAF2

Description

FUNCTION_DESCRIPTION

Usage

sourceTAF2(script, quiet = FALSE)

Arguments

script PARAM_DESCRIPTION

quiet PARAM_DESCRIPTION, Default: FALSE

Details

DETAILS

Value

OUTPUT_DESCRIPTION

Examples

```
## Not run:
if(interactive()){
  #EXAMPLE1
}

## End(Not run)
```

taf_gitignore*taf_gitignore*

Description

FUNCTION_DESCRIPTION

Usage

taf_gitignore(path = ".", append = FALSE)

Arguments

path PARAM_DESCRIPTION, Default: '?'
 append PARAM_DESCRIPTION, Default: FALSE

Details

DETAILS

Value

OUTPUT_DESCRIPTION

Examples

```
## Not run:
if(interactive()){
  #EXAMPLE1
}

## End(Not run)
```

taf_roxy	<i>create a blank data set file</i>
----------	-------------------------------------

Description

description

Usage

taf_roxy(dataset)

Arguments

dataset the name of the dataset that will be created

Details

DETAILS

Value

OUTPUT_DESCRIPTION

Examples

```
## Not run:
if (interactive()) {
  # EXAMPLE1
}

## End(Not run)
```

update_r_win	<i>update_r_win</i>
--------------	---------------------

Description

FUNCTION_DESCRIPTION

Usage

```
update_r_win(check.only = TRUE)
```

Arguments

check.only if TRUE R binaries are not downloaded and installed, only a check is performed

Value

OUTPUT_DESCRIPTION

Examples

```
## Not run:
if(interactive()){
  #EXAMPLE1
}

## End(Not run)
```

write.taf.json	<i>write.taf.json</i>
----------------	-----------------------

Description

FUNCTION_DESCRIPTION

Usage

```
write.taf.json(x, file = NULL, dir = NULL, ...)
```

Arguments

x	PARAM_DESCRIPTION
file	PARAM_DESCRIPTION, Default: NULL
dir	PARAM_DESCRIPTION, Default: NULL
...	PARAM_DESCRIPTION

Details

DETAILS

Value

no return value

See Also

[read_json](#)

write_sam_upload	<i>Write SAM assesseent to a TAF json file</i>
------------------	--

Description

Write a SAM assessment to a json file int the correct format to be uploaded to the TAF assessment results database

Usage

```
write_sam_upload(fit, dir = NULL)
```

Arguments

fit	a fitted object from a SAM model fit.
dir	an optional directory name.

Index

add, [5](#), [7](#)
add.roxy, [2](#)

canvas, [2](#)
commit, [5](#), [7](#)

git_init, [4](#)

init, [5](#), [7](#)
installed_packages, [5](#)

makeOxygen, [2](#)

open_dir, [6](#)

paradox (canvas), [2](#)
paradoxes (canvas), [2](#)

quick_help, [7](#)

read_json, [11](#)

scale_shape (canvas), [2](#)
sourceTAF2, [8](#)

taf.skeleton, [7](#)
taf_gitignore, [8](#)
taf_roxy, [9](#)
tidy_source, [2](#)

update_r_win, [10](#)

write.taf.json, [10](#)
write_sam_upload, [11](#)