Package: caMisc (via r-universe)

September 17, 2024

Type Package

Title Different Functions

Version 1.0.2

Date 2023-08-02

Author Christoffer Moesgaard Albertsen [aut, cre], Technical University of Denmark [cph]

Maintainer Christoffer Moesgaard Albertsen <cmoe@aqua.dtu.dk>

Description More about what it does (maybe more than one line)

License BSD_2_clause + file LICENSE

Imports rmarkdown, MASS, jpeg, png, tiff, jsonlite, XML, RCurl, utils, methods

LinkingTo RcppEigen

RoxygenNote 7.2.3

Repository https://calbertsen.r-universe.dev

RemoteUrl https://github.com/calbertsen/caMisc

RemoteRef HEAD

RemoteSha 51710160f4ece1d5cc36c8fbc66a5cc500743aee

Contents

addShade *Add shade to a color name*

Description

Add shade to a color name

Usage

```
addShade(name, shade = 0)
```

Arguments

`name`	name of the color
`shade`	shade value (between 0 and 1). For zero, the same color is returned.

Value

a new color name

Author(s)

Christoffer Moesgaard Albertsen

addTint *Add tint to a color name*

Description

Add tint to a color name

Usage

```
addTint(name, tint = 0)
```

Arguments

`name`	name of the color
`tint`	tint value (between 0 and 1). For 0, the same color is returned

Value

a new color name

Author(s)

Christoffer Moesgaard Albertsen

addTone *Add tone to a color name*

Description

Add tone to a color name

Usage

```
addTone(name, tone = 0)
```

Arguments

`name`	name of the color
`tone`	tone value (between 0 and 1). For 0, the same color is returned.

Value

a new color name

Author(s)

Christoffer Moesgaard Albertsen

addTrans *Add alpha value to a color name*

Description

Add alpha value to a color name

Usage

```
addTrans(name, alpha = 1)
```

Arguments

`name`	name of the color
`alpha`	alpha value (between 0 and 1)

Value

a new color name

Author(s)

Christoffer Moesgaard Albertsen

axisInside *Draw figure axis inside plot*

Description

Draw figure axis inside plot

Usage

```
axisInside(side)
```

Arguments

`side`	side of plot to draw on

Value

Nothing, but plots as side effect

Author(s)

Christoffer Moesgaard Albertsen

`beanplot` *Beanplot of something*

Description

Beanplot of something

Usage

```
beanplot(x, ...)
```

Arguments

`x`	Some object
`...`	Other arguments

Value

Creates a plot

Author(s)

Christoffer Moesgaard Albertsen

`bgtext` *Add text with background to plot*

Description

Add text with background to plot

Usage

```
bgtext(
  x,
  y,
  labels,
  ...,
  cex = 1,
  font = NULL,
  bg = "white",
  bgex = 1,
  border = NA
)
```

Arguments

`x`	x-coordinate
`y`	y-coordinate
`labels`	label to add
`...`	arguments passed to text
`cex`	size of text
`font`	font of text
`bg`	background color
`bgex`	background extend factor
`border`	border color of background

Author(s)

Christoffer Moesgaard Albertsen

`buildFromGithub` *Download and build an R package from GitHub*

Description

Download and build an R package from GitHub

Usage

```
buildFromGithub(
  repo,
  ref,
  subdir = NULL,
  buildArgs = c("--no-build-vignettes")
)
```

Arguments

`repo`	GitHub user and repository separated by /
`ref`	Reference to commit or branch. Default is master
`subdir`	Path to subdir containing the package. Should be NULL if the package is in the top directory
`buildArgs`	Character vector of arguments passed to R CMD build. Default is "–no-build-vignettes"

Value

Nothing

Author(s)

Christoffer Moesgaard Albertsen

`collapse` *Collapse vector to string*

Description

Collapse vector to string

Usage

```
collapse(
  x,
  collap = c(rep(", ", length(x) - 2), ifelse(length(x) > 2, ", and ", " and "))
)
```

Arguments

`x`	vector of values
`collap`	rule for collapsing

Value

a string

Author(s)

Christoffer Moesgaard Albertsen

`displayNum` *Convert number to display text*

Description

Convert number to display text

Usage

```
displayNum(
  x,
  capitalize = FALSE,
  big.mark = ",",
  decimal.mark = ".",
  small.mark = "",
  digits = 0
)
```

Arguments

`x`	number
`capitalize`	Capitalize first letter?

Value

LaTeX code for the fraction

Author(s)

Text representation of number

`dtucols` *DTU color palette*

Description

DTU color palette

Usage

```
dtucols(x)
```

Arguments

`x`	name of colors to return

Value

color codes

Author(s)

Christoffer Moesgaard Albertsen

`dzslides_presentation` *Convert to a DZSlides presentation*

Description

Format for converting from R Markdown to a DZSlides presentation.

Usage

```
dzslides_presentation(
  incremental = FALSE,
  fig_width = 8,
  fig_height = 6,
  fig_retina = if (!fig_caption) 2,
  fig_caption = FALSE,
  dev = "png",
  smart = TRUE,
  self_contained = TRUE,
  highlight = "default",
  mathjax = "default",
  template = "default",
  css = NULL,
  includes = NULL,
  keep_md = FALSE,
  lib_dir = NULL,
  md_extensions = NULL,
  pandoc_args = NULL,
  ...
)
```

Arguments

`incremental`	should lists be incremental?
`fig_width`	figure width
`fig_height`	figure height
`fig_retina`	retina figure?
`fig_caption`	figure caption?
`dev`	figure device
`smart`	smart?
`self_contained`	self contained?
`highlight`	highlighting style
`mathjax`	mathjax
`template`	template path
`css`	extra css

`includes`	extra includes
`keep_md`	keep markdown file?
`lib_dir`	...
`md_extensions`	markdown extensions to use
`pandoc_args`	extra pandoc arguments
`...`	other arguments

Value

R Markdown output format to pass to `render`

Author(s)

Christoffer Moesgaard Albertsen

Examples

```
## Not run:

library(rmarkdown)

# simple invocation
render("pres.Rmd", dzslides_presentation())

# specify an option for incremental rendering
render("pres.Rmd", dzslides_presentation(incremental = TRUE))

## End(Not run)
##' @export
```

---

fd *Format number*

Description

Small wrapper for formatC to limit text needed

Usage

```
fd(x, digits = 1)
```

Arguments

`x`	number
`digits`	number of digits

Value

string

Author(s)

Christoffer Moesgaard Albertsen

`formatDate` *Format date with locale*

Description

Format date with locale

Usage

```
formatDate(x, format = "", locale = Sys.getlocale("LC_TIME"), ...)
```

Arguments

`x`	date
`format`	format to use
`locale`	locale to use
`...`	passed to strftime

Value

formatted date string

Author(s)

Christoffer Moesgaard Albertsen

gcovTable *Create coverage table from list of gcov objects*

Description

Create coverage table from list of gcov objects

Usage

```
gcovTable(x, ...)
```

Arguments

`x`	list of gcov objects
`...`	additional arguments passed to formatC

Value

A coverage table

Author(s)

Christoffer Moesgaard Albertsen

getPixelMatrix *Get a pixel matrix from an image file*

Description

Get a pixel matrix from an image file

Usage

```
getPixelMatrix(file, grey = FALSE)
```

Arguments

`file`	Path to image file
`grey`	Should output be greyscale?

Value

A matrix of pixel values (0-255)

Author(s)

Christoffer Moesgaard Albertsen

grad *Calculate gradient of a function*

Description

Calculate gradient of a function

Usage

```
grad(
  func,
  x,
  h = abs(1e-04 * x) + 1e-04 * (abs(x) < sqrt(.Machine$double.eps/7e-07)),
  ...
)
```

Arguments

func	function
x	parameter values
h	step size
...	passed to func

Value

gradient vector

greenBlindness *Emulate green blindness in colors*

Description

Based on https://personal.sron.nl/~pault/data/colourschemes.pdf

Usage

```
greenBlindness(red, green, blue, alpha = 1, names = NULL, maxColorValue = 255)
```

Arguments

red	Red RGB value (or color HEX code or name - alpha is ignored)
green	Green RGB value
blue	Blue RGB value
alpha	alpha value to use
names	Names for the resulting vector
maxColorValue	Maximum color value for red, green, blue

Value

New colors

Author(s)

Christoffer Moesgaard Albertsen

---

imagePlot *Plot image from file*

Description

Plot image from file

Usage

```
imagePlot(
  x,
  objectFit = c("fill", "contain", "cover", "none", "scale-down"),
  halign = c("c", "l", "r"),
  valign = c("c", "t", "b"),
  maxWidthPct = 1,
  maxHeightPct = 1,
  add = FALSE,
  noMargin = TRUE,
  ...
)
```

Arguments

x	path to image
objectFit	How the image should fill the plot: "fill", "contain", "cover", "none","scale-down"
halign	Horizontal alignment: c, l, r
valign	Vertical alignment: c, t, b
noMargin	Plot with oma and mar set to zero
...	other arguments

Value

Plots the image

Author(s)

Christoffer Moesgaard Albertsen

installDependencies *Download, build, and install package dependencies*

Description

Download, build, and install package dependencies

Usage

```
installDependencies(
  descriptionPath,
  buildArgs = c("--no-build-vignettes"),
  installArgs = c(),
  dependencies = c("Depends", "Imports", "LinkingTo")
)
```

Arguments

`descriptionPath`	Path to a DESCRIPTION file
`buildArgs`	Character vector of arguments passed to R CMD build.
`installArgs`	Character vector of arguments passed to R CMD INSTALL.
`dependencies`	Character vector of dependency types to install ("Depends","Imports","LinkingTo","Enhances", or "Suggests")

Value

Nothing

Author(s)

Christoffer Moesgaard Albertsen

installFromGithub *Download, build, and install an R package from GitHub*

Description

Download, build, and install an R package from GitHub

Usage

```
installFromGithub(
  repo,
  ref,
  subdir = NULL,
  buildArgs = c("--no-build-vignettes"),
  installArgs = c(),
  dependencies = c("Depends", "Imports", "LinkingTo"),
  https = TRUE
)
```

Arguments

`repo` GitHub user and repository separated by /

`ref` Reference to commit or branch. Default is master

`subdir` Path to subdir containing the package. Should be NULL if the package is in the top directory

`buildArgs` Character vector of arguments passed to R CMD build.

`installArgs` Character vector of arguments passed to R CMD INSTALL.

`dependencies` Character vector of dependency types to install ("Depends","Imports","LinkingTo","Enhances", or "Suggests")

Value

Nothing

Author(s)

Christoffer Moesgaard Albertsen

`jacobian` *Calculate jacobian of a function*

Description

Calculate jacobian of a function

Usage

```
jacobian(
  func,
  x,
  h = abs(1e-04 * x) + 1e-04 * (abs(x) < sqrt(.Machine$double.eps/7e-07)),
  ...
)
```

Arguments

`func`	function
`x`	parameter values
`h`	step size
`...`	passed to func

Value

jacobian matrix

`makeMapLegend` *Add map legend for choropleth map*

Description

Add map legend for choropleth map

Usage

```
makeMapLegend(x, cols, txt)
```

Arguments

`x`	Variable values plotted
`cols`	Colors
`txt`	Title / description

Author(s)

Christoffer Moesgaard Albertsen

`makeSquare` *Make a pixel matrix/array square*

Description

Make a pixel matrix/array square

Usage

```
makeSquare(p, value = 0, asp = 1)
```

Arguments

`p`	Pixel matrix/array
`value`	Value of added pixels

Author(s)

Christoffer Moesgaard Albertsen

`packageSkeleton` *Create a package skeleton*

Description

Create a package skeleton

Usage

```
packageSkeleton(pkg, path = file.path("."))
```

Arguments

`pkg`	Name of the new package
`path`	(existing) Directory of package

Value

Used for side effects

Author(s)

Christoffer Moesgaard Albertsen

`particlefilter` *Bootstrap particle filter*

Description

Function to run a bootstrap particle filter for user defined model.

Usage

```
particlefilter(N, T, G, M, F = NULL, envir = .GlobalEnv, seed = NULL)
```

Arguments

`N`	Number of particles
`T`	Number of time steps
`G`	Function to simulate hidden states (of dimension p) given previous (See Details)
`M`	Function to calculate log-likelihood of data for a given particle
`F`	Do not use!
`envir`	Environment the functions should be evaluated in (containing data and parameters)
`seed`	Random seed to start the filter

Details

The G function should be of the form G <- function(t,X) and return a numeric vector of simulate values from $X_t|X_{t-1} = X$. For $t = 1$ the function will be called as G(1,NULL). The M function should be of the form M <- function(t, X) and return the density of the observation at time t given the latent state X_t

Value

A list with a p x N x T array X containing the simulated particles (X) and a vector of length T with the negative log-likelihood contribution at each time point.

Author(s)

Christoffer Moesgaard Albertsen <`cmoe@aqua.dtu.dk`>

Examples

```
## Univariate example
env <- new.env()
env$pars <- list(sdobs = 0.4, sdstate = 0.3)
local({xx <- cumsum(rnorm(100,0,pars$sdstate))},env)
local({dat <- xx + rnorm(100,0,pars$sdobs)},env)
G <- function(t,X){
            if(t == 1){
               return(rnorm(1,0,pars$sdstate))
            }else{
               return(rnorm(1,X,pars$sdstate))
            }
}
M <- function(t,X){
    return(dnorm(dat[t],X,pars$sdobs, TRUE))
}
pest <- particlefilter(N = 1000, T = 100,
                 G=G,M=M,
                 envir=env, seed=1)
## Not run:
plot(env$dat)
lines(apply(pest$X[1,,],2,mean),col="red")
lines(apply(pest$X[1,,],2,mean) + 2 * apply(pest$X[1,,],2,sd),col="red",lty=2)
lines(apply(pest$X[1,,],2,mean) - 2 * apply(pest$X[1,,],2,sd),col="red",lty=2)
lines(env$xx)

## End(Not run)

## Bivariate example
env <- new.env()
env$pars <- list(sdobs = 0.4, sdstate = 0.3)
local({xx <- cbind(cumsum(rnorm(100,0,pars$sdstate)),
             cumsum(rnorm(100,0,pars$sdstate)))},env)
local({dat <- xx + matrix(rnorm(2 * 100,0,pars$sdobs),ncol=2)},env)
G <- function(t,X){
            if(t == 1){
               return(rnorm(2,0,pars$sdstate))
            }else{
               return(rnorm(2,X,pars$sdstate))
            }
}
M <- function(t,X){
    return(sum(dnorm(dat[t,],X,pars$sdobs,TRUE)))
}
pest <- particlefilter(N = 1000, T = 100,
                 G=G,M=M,
                 envir=env, seed=1)
## Not run:
layout(cbind(1,c(2,3)))
plot(env$dat)
lines(env$xx)
lines(apply(pest$X,c(3,1),mean),col="red")
plot(env$dat[,1])
lines(apply(pest$X[1,,],2,mean),col="red")
lines(apply(pest$X[1,,],2,mean) + 2 * apply(pest$X[1,,],2,sd),col="red",lty=2)
lines(apply(pest$X[1,,],2,mean) - 2 * apply(pest$X[1,,],2,sd),col="red",lty=2)
lines(env$xx[,1])
plot(env$dat[,2])
lines(apply(pest$X[2,,],2,mean),col="red")
lines(apply(pest$X[2,,],2,mean) + 2 * apply(pest$X[2,,],2,sd),col="red",lty=2)
lines(apply(pest$X[2,,],2,mean) - 2 * apply(pest$X[2,,],2,sd),col="red",lty=2)
lines(env$xx[,2])

## End(Not run)
```

read_gcov *Read content of .gcov file*

Description

Read content of .gcov file

Usage

```
read_gcov(file)
```

Arguments

`file`	path to .gcov file

Value

a gcov object

Author(s)

Christoffer Moesgaard Albertsen

`read_massif` *Read valgrind Massif output*

Description

Read valgrind Massif output

Usage

```
read_massif(file, keep_details = FALSE)
```

Arguments

`file`	file to read
`keep_details`	keep details?

Value

massif S3 object

Author(s)

Christoffer Moesgaard Albertsen

solarPosition *Calculate solar angles*

Description

Calculate solar angles

Usage

```
solarPosition(date, lat, lon)
```

Arguments

`date`	date (UTC) in the format "YYY-mm-dd HH:MM:SS"
`lat`	latitude of observer
`lon`	longitude of observer

Value

list of values

Author(s)

Christoffer Moesgaard Albertsen Modified from https://doi.org/10.1016/j.renene.2021.03.047

tofrac *Print number as (LaTeX) fraction*

Description

Print number as (LaTeX) fraction

Usage

```
tofrac(x, dollar = TRUE)
```

Arguments

`x`	number
`dollar`	

Value

LaTeX code for the fraction

Author(s)

Christoffer Moesgaard Albertsen

`toGreyscale` *Turn pixel into grey scale*

Description

Turn pixel into grey scale

Usage

```
toGreyscale(p)
```

Arguments

`p`	Pixel matrix/array

Value

A grey scale pixel matrix/array

Author(s)

Christoffer Moesgaard Albertsen

Index