Package: rutils (via r-universe)

October 27, 2024

Type Package

Title A Collection of R Functions

Version 0.0.0.9000

Description A collection of R functions commonly used in FRB-CESAB projects.

URL https://github.com/frbcesab/rutils

BugReports https://github.com/frbcesab/rutils/issues

License GPL (>= 2)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports graphics, grDevices, httr, imager, jsonlite, png, stats, tidyr, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

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

RemoteUrl https://github.com/FRBCesab/rutils

RemoteRef HEAD

RemoteSha e62ff16aec9a243861d6e9a75866b24ad56651cf

Contents

`address_to_coords` *Get coordinates of any location in the World*

Description

Retrieves geographic coordinates (longitude and latitude) of any location in the World using the OpenStreetMap Data Search Engine Nominatim http://nominatim.openstreetmap.org.

Usage

```
address_to_coords(address)
```

Arguments

`address`	a `character` of length 1. The location for which coordinates will be found.

Value

A `data.frame` with two columns: `lon`, the longitude and `lat`, the latitude of the location.

Examples

```
address_to_coords("Houston, TX, USA")
address_to_coords("5 rue de l'école de Médecine, Montpellier, France")
```

`coords_to_address` *Retrieve OSM address from coordinates*

Description

Retrieves any location name in the World from geographic coordinates (longitude and latitude) using the OpenStreetMap Data Search Engine Nominatim http://nominatim.openstreetmap.org.

Usage

```
coords_to_address(coords)
```

Arguments

coords	a numeric of length 2. The first element is the latitude and the second the longitude.

Value

A character of length 1 with the OSM name of the location.

Examples

```
coords_to_address(c(43.61277, 3.873486))
coords_to_address(c(29.75894, -95.3677))
```

---

get_phylopic_image *Download a PhyloPic silhouette*

Description

This function downloads a silhouette from the PhyloPic database. With the new API, some functions of the package rphylopic don't work anymore. This function aims to temporary replace the function rphylopic::image_data() as this package seems to be looking for a new maintainer.

Usage

```
get_phylopic_image(uuid, size = 512)
```

Arguments

uuid	a character of length 1. The UUID (unique taxa identifier in the Phylopic database).
size	an integer of length 1. The size of the PNG to download. An error will be raised in the size is not available.

Value

An array of dimensions red x green x blue x alpha.

Examples

```
## Not run:
mammals_pic <- get_phylopic_image("8cad2b22-30d3-4cbd-86a3-a6d2d004b201")

## Plot the silhouette ----
ggplot2::ggplot(x = Sepal.Length, y = Sepal.Width, data = iris,
                geom = "point") +
rphylopic::add_phylopic(mammals_pic, alpha = 1)

## End(Not run)
```

`get_world_basemap` *Download IPBES World countries boundaries*

Description

This function downloads a spatial vector of world countries boundaries as defined by the IPBES (available at https://zenodo.org/record/3928281). A `zip` file is downloaded in the folder `path` (working directory by default) and files are extracted in the same folder. If this folder doesn't exist it will be created.

Note: if you delete the zip `file (after files extraction)` and `re-run this function`, the `zip` file will be downloaded again. It this is your wish do not forget to use `force = TRUE` to erase previous files.

Original Projection System: WGS84

Usage

```
get_world_basemap(path = ".", force = FALSE)
```

Arguments

`path`	a `character` of length 1. The folder inside which the `zip` will be downloaded and extracted. Default is `getwd()`.
`force`	a `logical`. If `TRUE` previous files will be erased. Default is `FALSE`.

Value

No return value.

Examples

```
## Not run:
## Download and extract zip file ----
get_world_basemap()

## Check ----
list.files()

## Re-run function ----
get_world_basemap()              # No download (zip file already present)
get_world_basemap(force = TRUE)  # Erase previous files

## End(Not run)
```

`hiking_time` *Hiking time calculation*

Description

Estimates the hiking time for a planned route by considering horizontal distance, elevation gain and elevation loss.

This function implements the DIN 33466 of the German Alpine Club. Different speeds are assumed according to this standard:

- 4 kilometers per hour in the horizontal direction
- 300 vertical meters in the ascent per hour
- 500 vertical meters in the descent per hour

Usage

```
hiking_time(distance, elevation_gain, elevation_loss)
```

Arguments

`distance`	a `numeric` of length 1. The total distance of the hike (in kilometers). Must be strictly positive.
`elevation_gain`	a `numeric` of length 1. The total elevation gain (cumulative positive elevation). Must be positive or zero.
`elevation_loss`	a `numeric` of length 1. The total elevation loss (cumulative negative elevation). Must be positive or zero.

Value

A `character` of length 1 returning the hiking time as `"23h59"`.

Examples

```
## Lac du Montagnon par le Col d'Iseye
## https://www.visorando.com/randonnee-lac-du-montagnon-par-le-col-d-iseye/

hiking_time(distance = 20, elevation_gain = 1460, elevation_loss = 1460)
```

`leading_zero` *Add leading zeros to a vector*

Description

Adds n leading zeros to a vector so that each element of the vector has the same number of characters. See examples below.

Usage

```
leading_zero(x)
```

Arguments

`x`	a `character` or a `numeric` vector.

Value

A `character` vector of the same length as `x`.

Examples

```
## On numeric vector ----
x <- 1:10
leading_zero(x)

x <- 50:100
leading_zero(x)

## On floating vector ----
x <- c(1, 9.5, 10, 10.2)
leading_zero(x)

## On character vector ----
x <- as.character(1:10)
leading_zero(x)

## Creating sortable identifiers ----
x1 <- as.character(10:1)
x2 <- leading_zero(x)

sort(x1)
sort(x2)

x2 <- paste0("ID", x2)
sort(x2)
```

len *A shortcut for the function length*

Description

A shortcut for the function length

Usage

```
len(x)
```

Arguments

x	an R object.

Value

See ?length for a complete description.

Examples

```
length(letters)

len(letters)
```

multi_merge *Combine different matrices by row names and column names*

Description

This function combines different matrices by row names and column names by performing a 2-dimension full join. Gaps are filled with NA (default) or 0 (argument na_to_zero).

Usage

```
multi_merge(..., na_to_zero = FALSE)
```

Arguments

...	one or several matrix objects.
na_to_zero	a logical value. If TRUE gaps (i.e. unknown edges) are coded as 0. Otherwise they are coded as NA (default).

Value

A matrix object.

Examples

```
mat1 <- matrix(rep(1, 9), nrow = 3)
colnames(mat1) <- c("A", "B", "C")
rownames(mat1) <- c("A", "B", "C")

mat2 <- matrix(rep(1, 9), nrow = 3)
colnames(mat2) <- c("D", "E", "F")
rownames(mat2) <- c("D", "E", "F")

mat3 <- matrix(rep(1, 9), nrow = 3)
colnames(mat3) <- c("F", "G", "H")
rownames(mat3) <- c("G", "A", "H")

multi_merge(mat1, mat2, mat3)

multi_merge(mat1, mat2, mat3, na_to_zero = TRUE)
```

---

plot_2d_img *Plot two variables using images png as points*

Description

Plot two variables using images png as points.Works with any king of png, the best stratgegy is to use png with no backgrounds. Parameter scale and size are important to tune to make the plot readable and not to heavy Must be plotted (or saved) on a 1:1 dimension

Usage

```
plot_2d_img(
  df,
  scale,
  size,
  pathimages,
  cexaxis,
  cexlab,
  labelx,
  labely,
  lm,
  xR,
  yR,
  colR,
  cexR,
  colline,
  lwline,
  ltyline,
  ...
)
```

Arguments

df	a data.frame with three colums x coord,y coord,img_names
scale	a numeric between 0 and 1 (used to resize the images)
size	the parameter size of the function imager::resize (between -0L and -100L) used to reduce the resolution of the image (proportion between 0 and 100%)
pathimages	path to the images
cexaxis	cex.axis of the plot function
cexlab	cex.lab of the plot function
labelx	label of the x axis
labely	label of the y axis
lm	TRUE or FALSE (compute and draw the lm)
xR	x coordinate of the position of the R2 on the plot
yR	y coordinate of the position of the R2 on the plot
colR	color of the R2 on the plot
cexR	size of the R2 on the plot
colline	col of the abline used to illustrate the lm
lwline	lwd of the abline used to illustrate the lm
ltyline	lty of the abline used to illustrate the lm
...	other plot options; see ?par (as mar or mpg)

Value

a plot

Examples

```
## Not run:
set.seed(3)
df <- cbind.data.frame(x=runif(10, 0, 10),y=runif(10, 0, 10),
img_names=paste0("img_",rutils::leading_zero(1:10)))
pathimages <- system.file("extdata", "plot2dimg", package = "rutils")
plot_2d_img(df,scale=0.2,size=-35L,pathimages,
lm=TRUE,xR=1.5,yR=10,colR="gray",cexR=1.5,colline="gray",lwline=2,ltyline=2,
labelx="Variable x",labely="Variable y",mar=c(8, 9, 4.1, 2.1),mgp=c(6,2,0),
cex.lab=2.5,cex.axis=1.5)

## End(Not run)
```

rename_col *Rename any columns of a data.frame (from 1 to as much columns as you want)*

Description

Renames any columns of a `data.frame` (from 1 to as much columns as you want).

Usage

```
rename_col(df, old.col.names, new.col.names)
```

Arguments

`df`	a `data.frame`
`old.col.names`	a `character` vector of column names to rename
`new.col.names`	a `character` vector of column new names

Value

The same `data.frame` as `df`.

Examples

```
df <- cbind.data.frame(a = 1:10, b = 1:10, c = 1:10, d = 1:10)
df

rename_col(df,
           old.col.names = c("a", "d"),
           new.col.names = c("aa", "dd"))
```

signi *Produce produces significance symbols for the values of p (ns., *, **, ***)*

Description

Produce produces significance symbols for the values of p (ns., *, **, ***)

Usage

```
signi(p)
```

Arguments

`p`	a p value

Value

a character "ns", "*", *""*, **'"'**, **'"'**

Examples

```
signi(p=0.004)
```

`to_binomial_name` *Correct species binomial name case*

Description

Corrects species binomial name case.

Usage

```
to_binomial_name(x)
```

Arguments

`x`	a `character` vector with misspelled binomial names.

Value

A `character` vector of same length as `x`.

Examples

```
x <- c("sAlMO saLAR (Linnee, 1765)", "tyranosurus_REX rex.1")
to_binomial_name(x)
```

Index