Package: lehuynh (via r-universe)

September 11, 2024

Title Le-Huynh Truc-Ly's R Code and Templates

Version 0.1.1.9000

Author Truc-Ly Le-Huynh [aut, cre, cph] (<https://orcid.org/0000-0002-5227-2185>)

Maintainer Truc-Ly Le-Huynh <trucly.lehuynh@gmail.com>

Description Miscellaneous R functions (for graphics, data import, data transformation, and general utilities) and templates (for exploratory analysis, Bayesian modeling, and crafting scientific manuscripts).

License MIT + file LICENSE

URL https://github.com/le-huynh/lehuynh, https://le-huynh.github.io/lehuynh/

BugReports https://github.com/le-huynh/lehuynh/issues

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Suggests janeaustenr, testthat (>= 3.0.0), writexl

Config/testthat/edition 3

Imports brms, stats, ggplot2, usethis, ggpubr, purrr, stringr, magrittr, readxl, dplyr, fs, rio, tibble, tidyr, tidytext, igraph, networkD3

Repository https://le-huynh.r-universe.dev

RemoteUrl https://github.com/le-huynh/lehuynh

RemoteRef HEAD

RemoteSha 0c141be3544125625ae6c7116a283157e20a0c5a

Contents

`ggsave_elsevier` *Save a plot - Elsevier figure size*

Description

Save a plot using `ggplot2::ggsave()`. Plot size follows instructions of Elsevier journals.

Usage

```
ggsave_elsevier(
  filename,
  plot,
  width = c("one_column", "one_half_column", "full_page"),
  height,
  ...
)
```

Arguments

`filename`	A character string. File name to create on disk.
`plot`	Plot to save, ggplot or other grid object.
`width`	Plot width. See **Details** for more information.
`height`	Plot height in "mm".
`...`	Passed to `ggplot2::ggsave()`

Details

Instruction of Elsevier about sizing of artwork.

- Image width:
  - single column: 90 mm (255 pt)
  - 1.5 column: 140 mm (397 pt)
  - double column (full width): 190 mm (539 pt)
- Image height: maximum 240 mm.

Value

An image file containing the saved plot.

See Also

`ggplot2::ggsave()`

Examples

```
library(ggplot2)

fig <- ggplot(mtcars, aes(y = mpg, x = disp)) +
    geom_point(aes(colour = factor(cyl)))

## For demo, a temp. file path is created with the file extension .png
png_file <- tempfile(fileext = ".png")

ggsave_elsevier(png_file, plot = fig, width = "full_page", height = 120)
```

`import_data` *Import multiple files of the same format*

Description

This function imports multiple data files of the same format from a specified directory, optionally filtered by a pattern in the filenames.

Usage

```
import_data(path, file_format, pattern = NULL, ...)
```

Arguments

`path`	A character string specifying the directory path where the data files are located.
`file_format`	A character string specifying the type of files to import, using a glob pattern (e.g., "*.csv" for CSV files).
`pattern`	An optional character string specifying a regex pattern to filter filenames. Default is NULL.
`...`	Additional arguments passed to the `rio::import()` function.

Value

A named list where each element is the imported data from a file, with names corresponding to the filenames without the path and file extension.

Examples

```
## For demo, temp. file paths is created with the file extension .csv
csv_file1 <- tempfile(pattern = "test", fileext = ".csv")
csv_file2 <- tempfile(pattern = "file", fileext = ".csv")
csv_file3 <- tempfile(pattern = "test", fileext = ".csv")

## create CSV files to import
write.csv(head(cars), csv_file1)
write.csv(head(mtcars), csv_file2)
write.csv(head(iris), csv_file3)

## Import all CSV files in the directory
data_list <- import_data(path = tempdir(), file_format = "*.csv")

## Import all CSV files with names containing "test"
data_list <- import_data(path = tempdir(), file_format = "*.csv", pattern = "test")
```

`import_excel` *Import Excel file with multiple sheets*

Description

This function imports an Excel file with multiple sheets and returns a *named* `list` of imported sheets.

Usage

```
import_excel(file_path)
```

Arguments

`file_path`	A character string specifying the path to the Excel file.

Value

A named list where each element is the imported sheet from the Excel file, with names corresponding to the sheet names.

Examples

```
## For demo, a temp. file path is created with the file extension .xlsx
excel_file <- tempfile(fileext = ".xlsx")

## create Excel file with multiple sheets to import
writexl::write_xlsx(list(cars = head(cars), mtcars = head(mtcars)),
                    excel_file)

import_excel(file_path = excel_file)
```

`lehuynh_theme` *Le-Huynh's ggplot2 theme*

Description

Le-Huynh's ggplot2 theme: white background, black axis, black text

Usage

```
lehuynh_theme(base_size = 11, base_family = "", ...)
```

Arguments

`base_size`	Base font size
`base_family`	Base font family
`...`	Passed to `ggplot2::theme()`

Value

An object as returned by `ggplot2::theme()`

See Also

`ggplot2::theme()`, `ggplot2::theme_bw()`

Examples

```
library(ggplot2)

fig <- ggplot(mtcars, aes(y = mpg, x = disp)) +
    geom_point(aes(colour = factor(cyl)))

fig

fig + lehuynh_theme()
```

`MinMaxScaling` *Min-Max Standardization*

Description

Normalize / Standardize / Scale the data to the fixed range from 0 to 1. The minimum value of data gets transformed into 0. The maximum value gets transformed into 1. Other values get transformed into decimals between 0 and 1.

Usage

```
MinMaxScaling(x, y = x)
```

Arguments

`x`	A numeric vector to be scaled.
`y`	An optional numeric vector used to determine the scaling range. If not provided, the scaling range is determined by the values in `x`. Default: `y = x`.

Details

Min-max scaling is a normalization technique that transforms the values in a vector to a standardized range. The scaling is performed using the formula:

$$scaled_x = \frac{x - \min(y)}{\max(y) - \min(y)}$$

Value

A numeric vector of the same length as `x`, with values scaled to the range from 0 to 1.

Examples

```
dat1 = seq(from = 5, to = 30, length.out = 6)

MinMaxScaling(dat1)

dat2 = c(7, 13, 22)

MinMaxScaling(x = dat2, y = dat1)
```

`new_project` *Create a new project*

Description

This function sets up a new project within an **active R project** for reproducible purposes.

Usage

```
new_project()
```

Details

The project includes:

- **README.md**: the top level description of content in the project
- **Makefile**
- different folders to hold all *data*, *code*, *results* of data analysis, and *documents* related to the project
- templates: **manuscript.Rmd**, **code.R**, etc.

Value

A project containing folders and files for reproducible purposes.

Note

The function should be executed within an **active project**.

Recommended workflow:

1. Create a GitHub repository for the new project. At *Initialize this repository with a README*, choose NO.
2. Create a new RStudio Project via `git clone`.
3. Use function `new_project()` to generate folders and file templates.

References

Reproducibile Research Tutorial Series by Pat Schloss.

Examples

```
if(interactive()){
  new_project()
}
```

`ngrams_filter` *Filter and generate N-Grams from text data*

Description

Filter a dataset based on a specified column and group value, generate n-grams from a specified text column, then remove standard and user-defined stopwords from the n-grams.

Usage

```
ngrams_filter(
  data,
  group_column,
  group_name,
  text_column,
  ngrams,
  user_defined_stopwords = NULL
)
```

Arguments

`data`	A data frame containing the dataset to be processed.
`group_column`	A character string specifying the name of the column used to filter the data.
`group_name`	A character string specifying the value within the group column to filter the data by.
`text_column`	A character string specifying the name of the column containing text data to be tokenized into n-grams.
`ngrams`	An integer specifying the number of words in the n-grams to be generated.
`user_defined_stopwords`	A character vector of additional stopwords to be removed from the n-grams. Default is NULL.

Value

A data frame with the filtered data and generated n-grams, excluding the specified stopwords.

Examples

```
library(janeaustenr)

austen_books() %>%
         ngrams_filter(group_column = "book",
                       group_name = "Pride & Prejudice",
                       text_column = "text",
                       ngrams = 2)

austen_books() %>%
         ngrams_filter(group_column = "book",
                       group_name = "Pride & Prejudice",
                       text_column = "text",
                       ngrams = 2,
                       user_defined_stopwords = c("chapter", 1:50))
```

`plot_networkD3` *Plot network using NetworkD3*

Description

This function processes the input data to create an igraph object and then generates an interactive network plot based on the specified plot type. The plot can show the entire network, the largest component with a single color, or the largest component with different colors based on community detection. Node size and edge width are scaled based on node degree and edge weight, respectively.

Usage

```
plot_networkD3(
  data,
  col1 = "word1",
  col2 = "word2",
  plot_type = c("whole_network", "biggest_component_one_color",
    "biggest_component_community_color"),
  threshold,
  node_size = 20,
  edges_width = 10,
  opacity = 1,
  font_size = 15,
  ...
)
```

Arguments

`data`	A data frame containing the edge list for the network.
`col1, col2`	The name of two columns containing the symbolic edge list. Default is "word1" and "word2", respectively.
`plot_type`	A character string specifying the type of plot to generate. Options are "whole_network", "biggest_component_one_color", and "biggest_component_community_color". Default is "whole_network".
`threshold`	An integer specifying the minimum frequency of edges to be included in the network.
`node_size`	A numeric value specifying the base size of the nodes. Default is 20.
`edges_width`	A numeric value specifying the base width of the edges. Default is 10.
`opacity`	A numeric value specifying the opacity of the graph elements. Default is 1.
`font_size`	A numeric value specifying the font size of the node labels. Default is 15.
`...`	Additional arguments passed to `networkD3::forceNetwork()`.

Value

An interactive network plot created using `networkD3`.

Examples

```
library(janeaustenr)

data <- austen_books() %>%
          ngrams_filter(group_column = "book",
                        group_name = "Pride & Prejudice",
                        text_column = "text",
                        ngrams = 2)

# The whole network plot
plot_networkD3(data = data,
               threshold = 10)

# The biggest component plot with one color
plot_networkD3(data = data,
               plot_type = "biggest_component_one_color",
               threshold = 10)

# The biggest component plot with community based color
plot_networkD3(data = data,
               plot_type = "biggest_component_community_color",
               threshold = 10)
```

`ppc_brms` *Fitted versus observed plot for brmsfit Objects*

Description

Plot fitted versus observed values, including confidence interval (gray area) around best fit line (linear regression line) and prediction interval (dashed line).

Usage

```
ppc_brms(
  object,
  xtitle = "Observed value",
  ytitle = "Fitted value",
  dy = c(0.1, 0.1),
  dx = c(0.1, 0.1),
  cor = FALSE,
  equation = FALSE,
  xcor = NULL,
  ycor = NULL,
  xequ = NULL,
  yequ = NULL,
  ...
)
```

Arguments

`object`	An object of class **brmsfit**
`xtitle`	The text for the x-axis title
`ytitle`	The text for the y-axis title
`dy`	Distance from plot to y-axis
`dx`	Distance from plot to x-axis
`cor`	If TRUE, add correlation coefficients with p-values and R
`equation`	If TRUE, add regression line equation
`xcor, ycor`	`numeric` Coordinates (in data units) to be used for absolute positioning of the correlation coefficients
`xequ, yequ`	`numeric` Coordinates (in data units) to be used for absolute positioning of the regression line equation
`...`	Passed to `lehuynh_theme()`

Value

A ggplot object

Examples

```
## Not run:

library(brms)

mod <- brm(count ~ zAge + zBase * Trt + (1|patient) + (1|obs),
           data = epilepsy,
           family = poisson())

ppc_brms(mod)
ppc_brms(mod, dy = c(0.02, 0.1), dx = c(0.005, 0.1))
ppc_brms(mod, cor = TRUE, equation = TRUE, yequ = 100)

## End(Not run)
```

`tidytuesday` *Create a new folder for* #tidytuesday *challenge*

Description

This function sets up a new folder for *#tidytuesday* challenge within an **active project**. It creates a directory for the specified year and week, along with sub-directories for data, code, and plots. Template files are also added.

Usage

```
tidytuesday(year, week)
```

Arguments

`year`	An integer representing the year of the *#tidytuesday* challenge
`week`	An integer representing the week of interest, from 1 to 52

Details

The folder includes:

- **README.md**: plots for *#tidytuesday* challenge
- different folders to hold all *data*, *code*, *plots* of data analysis
- templates

Value

A folder containing folders and files for *#tidytuesday* challenge.

Note

Ensure that this function is called within an **active project**.

Examples

```
if(interactive()){
  tidytuesday(year = 2021, week = 25)
}
```

tsi *Calculate TSI (Trophic state index)*

Description

Calculate TSI. TSI range: 0 - 100.

Usage

```
tsi(x, type = c("chla", "TP", "TN", "SD"))
```

Arguments

`x`	numeric object
`type`	type of variable used to calculate TSI. See **Details** for more information.

Details

Trophic state classification (Carlson, 1996)

- <30-40, Oligotrophy
- 40-50, mesotrophy
- 50-70, eutrophy
- 70-100, hypereutrophy

Type of variable used to calculate TSI:

- SD: Secchi depth, meter
- chla: chlorophyll, ug/L or mg/m3
- TP: total Phosphorus, ug/L or mg/m3
- TN: total Nitrogen, mg/L

Carlson (1977): TSI-SD, TSI-Chla, TSI-TP

USEPA (2000): TSI-TN

Value

a numeric value.

References

Carlson, R. E. (1977). A trophic state index for lakes. Limnology and Oceanography, 22(2), 361-369.

Carlson, R. E., & Simpson, J. (1996). A Coordinator's Guide to Volunteer Lake Monitoring Methods. North American Lake Management Society, 73-92.

USEPA. (2000). Nutrient Criteria Technical Guidance Manual: Lakes and Reservoirs, 42-44.

Examples

```
chla <- c(0.12, 0.34, 0.94, 6.4)

tsi(chla, type = "chla")

tsi(chla, type = "TP")
```

Index