

# Package ‘lehuynh’

June 5, 2024

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

**Version** 0.1.1

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

**Maintainer** Truc-Ly Le-Huynh <[truclly.lehuynh@gmail.com](mailto:truclly.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

**RoxygenNote** 7.2.3

**Suggests** testthat (>= 3.0.0), writextl

**Config/testthat/edition** 3

**Imports** brms, stats, ggplot2, usethis, ggpubr, purrr, stringr,  
magrittr, readxl, dplyr, fs, rio, tibble

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2024-06-05 14:50:02 UTC

## Contents

ggsave_elsevier . . . . .	2
import_data . . . . .	3
import_excel . . . . .	4
lehuynh_theme . . . . .	5
MinMaxScaling . . . . .	5
new_project . . . . .	6

ppc_brms . . . . .	7
tidytuesday . . . . .	9
tsi . . . . .	10

<b>Index</b>	<b>12</b>
--------------	-----------

---

ggsave_elsevier	<i>Save a plot - Elsevier figure size</i>
-----------------	---

---

## 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 <b>Details</b> for more information.
height	Plot height in "mm".
...	Passed to <code>ggplot2::ggsave()</code>

## 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 <code>rio::import()</code> 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	<i>Le-Huynh's ggplot2 theme</i>
---------------	---------------------------------

---

### 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 <code>ggplot2::theme()</code>

### 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	<i>Min-Max Standardization</i>
---------------	--------------------------------

---

### 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

[Reproducible Research Tutorial Series](#) by Pat Schloss.

## Examples

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

---

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 <b>brmsfit</b>
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 <a href="#">lehuynh_theme()</a>

**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	<i>Create a new folder for #tidytuesday challenge</i>
-------------	---

---

## 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 <i>#tidytuesday</i> 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/m<sup>3</sup>
- TP: total Phosphorus, ug/L or mg/m<sup>3</sup>
- 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

`ggplot2::ggsave()`, 2  
`ggplot2::theme()`, 5  
`ggplot2::theme_bw()`, 5  
`ggsave_elsevier`, 2

`import_data`, 3  
`import_excel`, 4

`lehuynh_theme`, 5  
`lehuynh_theme()`, 8

`MinMaxScaling`, 5

`new_project`, 6

`ppc_brms`, 7

`rio::import()`, 3

`tidytuesday`, 9  
`tsi`, 10