# Functions in R

Profa. Thais

CE2 - 1/2020

# Programming

When writing your code, always remember that:

- Code is a vehicle for communication.
  - Even if you're not working with other people, you'll definitely be working with future-you!
- Good code style is like correct punctuation.
  - Youcanmanagewithoutit, but it sure makes things easier to read!
- Do not copy and paste a block of code more than twice!
  - Functions allow you to automate common tasks in a more powerful and general way than copy-and-pasting.

# Functions

```
df <- tibble::tibble(
  a = rnorm(10), b = rnorm(10),
  c = rnorm(10), d = rnorm(10))
df$a <- (df$a - min(df$a, na.rm = TRUE)) /
  (max(df$a, na.rm = TRUE) - min(df$a, na.rm = TRUE))
df$b <- (df$b - min(df$b, na.rm = TRUE)) /
  (max(df$b, na.rm = TRUE) - min(df$a, na.rm = TRUE))
df$c <- (df$c - min(df$c, na.rm = TRUE)) /
  (max(df$c, na.rm = TRUE) - min(df$c, na.rm = TRUE))
df$d <- (df$d - min(df$d, na.rm = TRUE)) /
  (max(df$d, na.rm = TRUE) - min(df$d, na.rm = TRUE))
```

# Functions

- What does the code do? Have you spotted any mistake?
- Writing a function has three big advantages over using copy-and-paste:

    1. You can give a function an evocative name that makes your code easier to understand.
    2. As requirements change, you only need to update code in one place, instead of many.
    3. You eliminate the chance of making incidental mistakes when you copy and paste (i.e. updating a variable name in one place, but not in another).

# Functions

- There are three key steps to creating a new function:

1. You need to pick a name for the function.
2. You list the inputs, or arguments, to the function inside function.
3. You place the code you have developed in body of the function, a { block that immediately follows function(...).

# Sintaxe

IQR() function: computes interquartile range of the x values.

```
IQR <- function (x, na.rm = FALSE, type = 7) {
diff(quantile(as.numeric(x), c(0.25, 0.75),
              na.rm = na.rm, names = FALSE,
              type = type))
}
```

- Sintaxe

```
name_function <- function (arguments) {
  body
}
```

# Exercise

Write a function to rescale a vector to have a range from 0 to 1.

```
df$a <- (df$a - min(df$a, na.rm = TRUE)) /
  (max(df$a, na.rm = TRUE) - min(df$a, na.rm = TRUE))
```

try to use range() function.

# Function names

The name of a function is important!

- Ideally, the name of your function will be short, but clearly evoke what the function does (But it's better to be clear than short).
- Generally, function names should be verbs, and arguments should be nouns.
- If your function name is composed of multiple words, I recommend using "snake_case" (be consistent).
- Where possible, avoid overriding existing functions and variables.

# Conditional execution

An if statement allows you to conditionally execute code. It looks like this:

```
if (condition) {
  # code executed when condition is TRUE
} else {
  # code executed when condition is FALSE
}
```

The condition must evaluate to either TRUE or FALSE.

- If it's a vector, you'll get a warning message.
- If it's an NA, you'll get an error.