



20 SEPTEMBER 2018

Herman Teirlinck,
01.69 - Paul Janssen

What have I done?!?

SORRY, no R last month...

... but check this out!

p.s. Thierry did some cool stuff and got approval of Jenny from Rstudio!

p.p.s. Could you give Damiano some thumbs up for his tidyR idea?

p.p.p.s. And Peter can make hexagon logo's with R!





FUNCTIONS

Functions

```
function_name <- function(var){  
  Do something  
  return(new_variable)  
}
```

Example

```
square <- function(x){  
  squared <- x*x  
  return(squared)  
}
```

Install the package suite:

```
This is base R, bro!
```

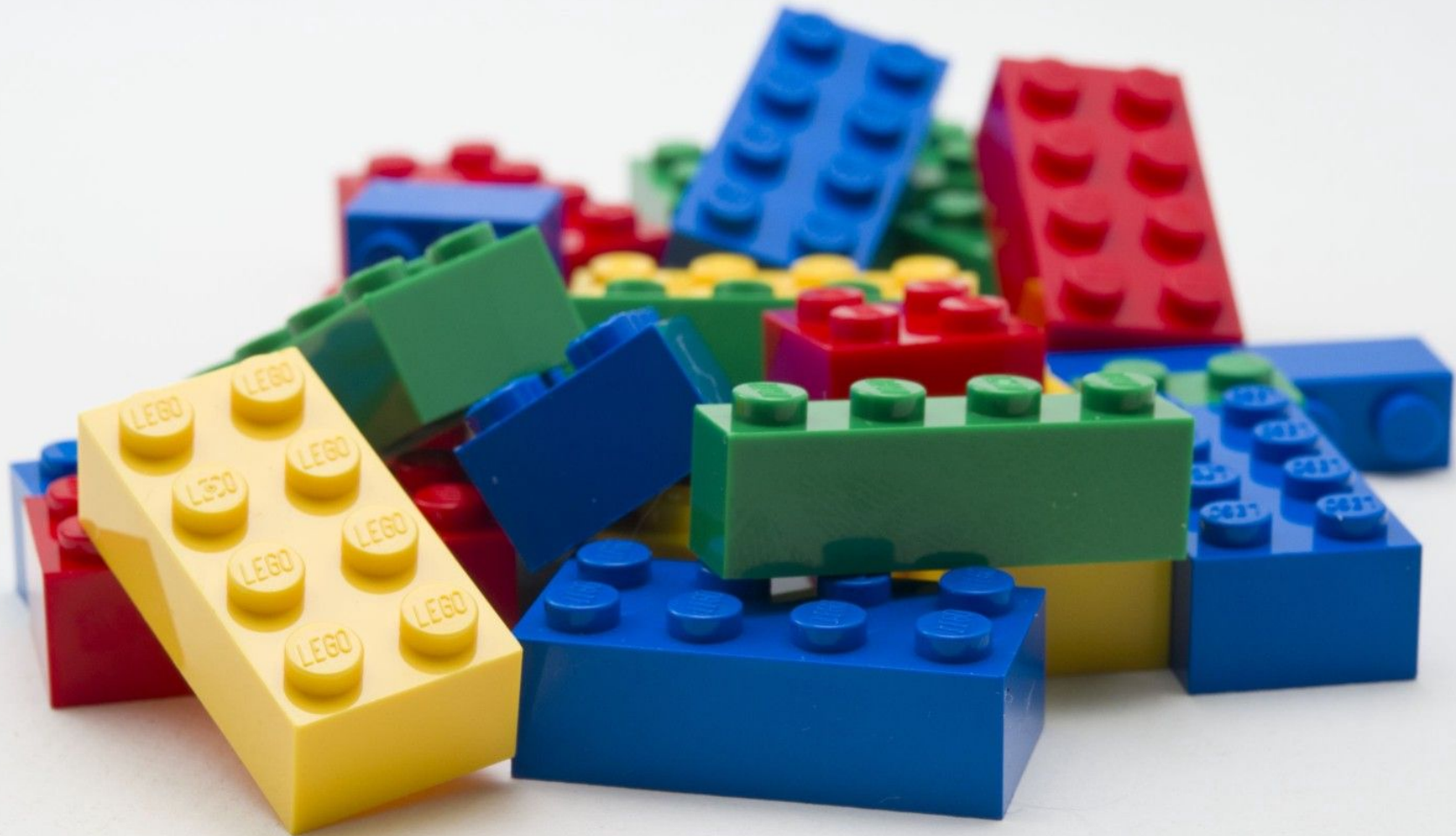
Load the package suite:

```
This is base R, bro!
```

Why functions?

```
# Redo with functions
temp1_celsius <- c(23.1, 25.5, 24.3, 24.4, 24.9, 25.7, 27.0)
druk1_atm <- 1.0000

celsius2kelvin <- function(celsius) {
  return(celsius + 273.15)
}
atm2pascal <- function(atm) {
  return(atm * 101325)
}
volume_gas_law <- function(t, p, n = 1, R = 8.314462) {
  return(n * R * t / p)
}
volume1 <- volume_gas_law(t = celsius2kelvin(temp1_celsius), p = atm2pascal(druk1_atm), n = n)
# double temperatuur in celsius
volume2 <- volume_gas_law(t = celsius2kelvin(temp1_celsius*2), p = atm2pascal(druk1_atm), n = n)
# half temperatuur in celsius
volume3 <- volume_gas_law(t = celsius2kelvin(temp1_celsius/2), p = atm2pascal(druk1_atm), n = n)
# Change altitude, pressure at 2000m
druk4_atm <- 0.7896
volume4 <- volume_gas_law(t = celsius2kelvin(temp1_celsius), p = atm2pascal(druk4_atm), n = n)
```



Share your snippets during the coding session!


Go to https://hackmd.io/OedO-zCaS7C_-oM-iSO-0w and post your code in between backticks:

For example:

```
```\n\nlibrary(lubridate)\n\nmy_data <- ... \n\n```
```



# The concept

We defined a number of challenges. If you were able to achieve a challenge, add a  to your laptop screen.

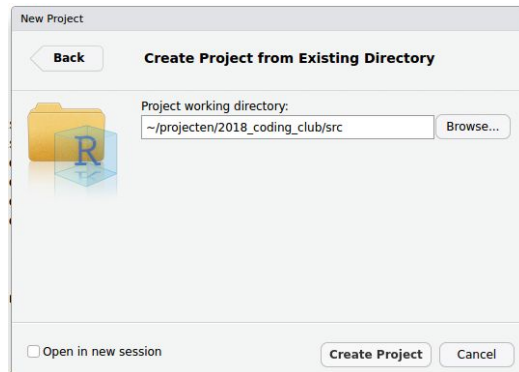
The objective is that **everyone** achieves !

- Someone has more  than you? **Ask for help!**
- Someone has less  than you? **Provide help!**

- Download coding club material and work locally, **not in sync** with the Google drive



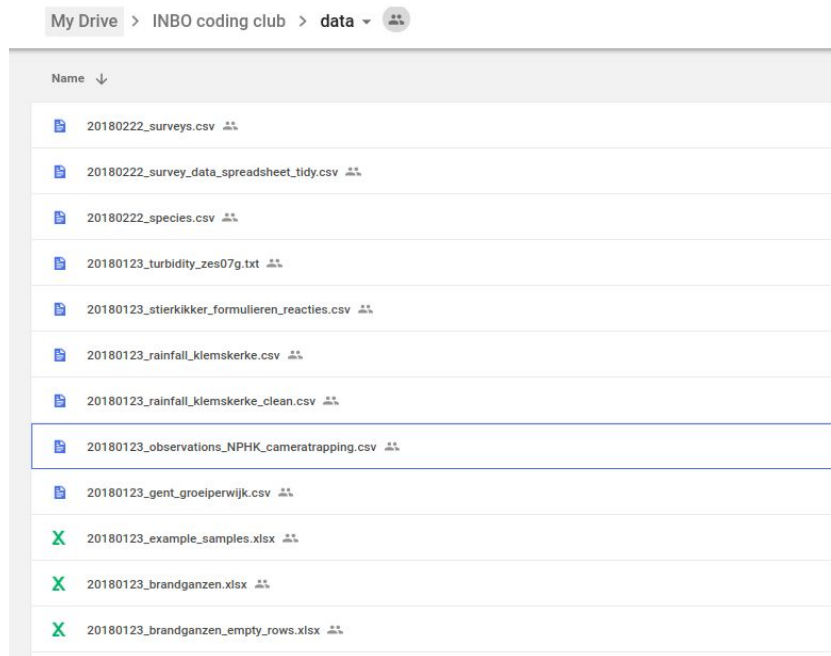
- Create new Rstudio project in the **/src** folder




- Download coding club material and work locally, not in sync with the Google drive
- Create new Rstudio project in the **src** folder...
- Use relative paths to data files!

```
> library(readr)
```

```
> read_csv2("../data/20180123_gent_groeiperwijk.csv")
```



- 
- Download the 20180821\_decay\_measurements\_x.csv (met x 1 tot 3) files from [data folder](#)
  - Download the file [20180920\\_challenge\\_1.R](#) and run the file
  - Write function get\_info\_decay() so that you can retrieve these summary values from 20180821\_decay\_measurements\_1.csv using your function:

```
library(readr)

decay_1 <- read_csv("../data/20180821_decay_measurements_1.csv")

Get summary stats about concentration decay
min_decay_1 <- min(decay_1$conc_data, na.rm = TRUE)# min
max_decay_1 <- max(decay_1$conc_data, na.rm = TRUE)
t_min_decay_1 <- decay_1$time[which.min(decay_1$conc_data)]# time to min
d-50
d_50_decay_1 <- decay_1$time[which.min(abs(max(decay_1$conc_data)/2 - decay_1$conc_data))]

summarize these info in data frame (output)
info_decay_1 <- data.frame(min_decay = min_decay_1,
 t_min_decay = t_min_decay_1,
 max_decay = max_decay_1,
 d_50_time = d_50_decay_1,
 stringsAsFactors = FALSE)
```



- Adapt the function `get_info_decay()` so that you can reuse your function to tackle `20180821_decay_measurements_2.csv` as well, without altering the data itself.

**Tip:** select column by name with double square brackets instead of \$:

```
df[[col_name]] instead of
df$col_name
```



Functions allow you to reuse your own coding work more easily, i.e. *Don't Repeat Yourself (DRY)*, aka less copy pasting of code...

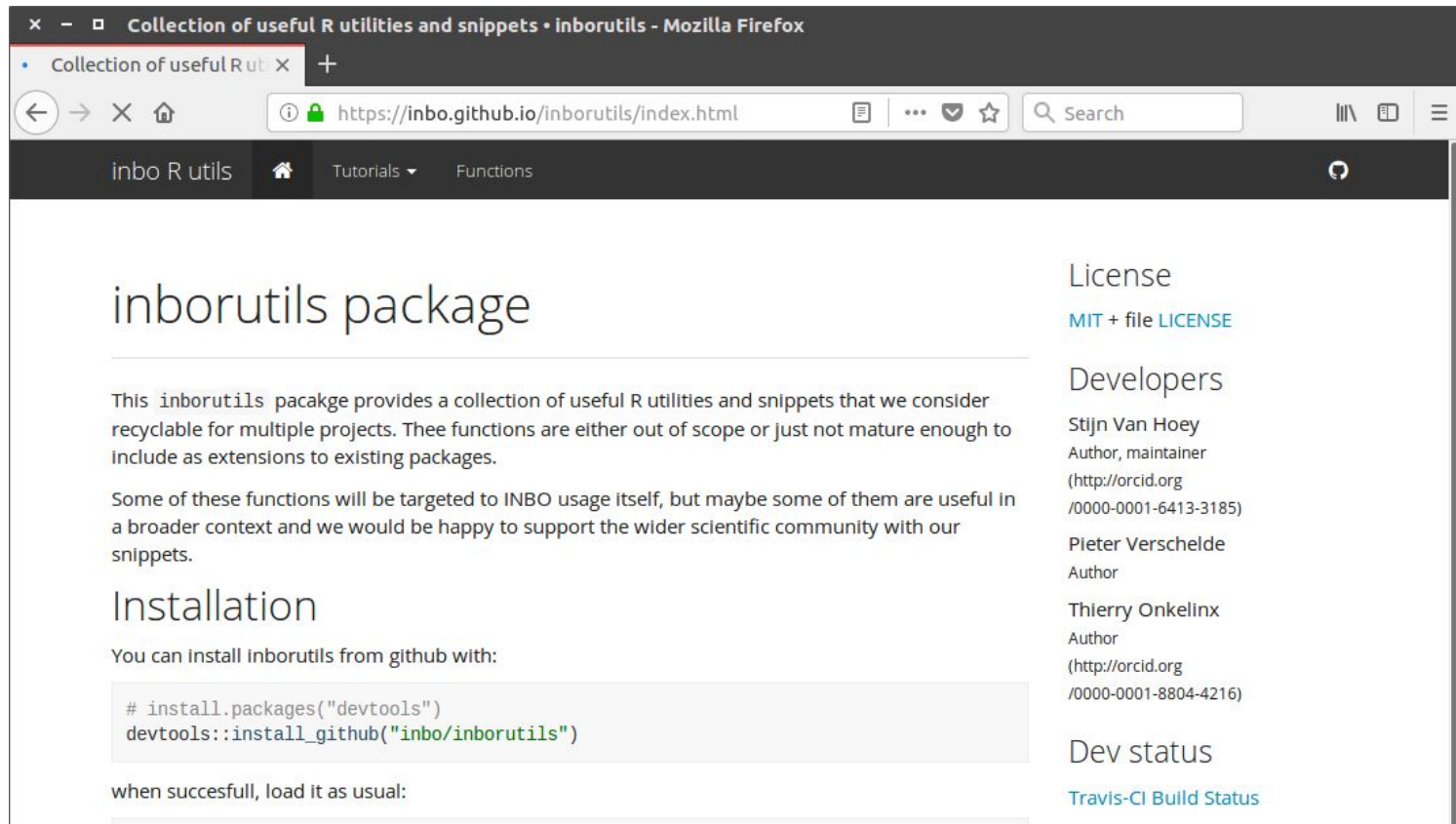
Let's use this idea for ggplot theme adaptations as well. Consider the following theme adaptations you need to do across a project:

```
theme(axis.text.x = element_text(size = 15),
 axis.title.x = element_text(size = 15, face = "bold"),
 axis.text.y = element_text(size = 15),
 axis.title.y = element_text(size = 15, face = "bold"))
```

Create a function and apply it to existing ggplots\* to add these theme options to the plot.

\* an example plot is provided at [20180920\\_challenge\\_3.R](#)

# Made functions useful for colleagues?



The screenshot shows a web browser window with the title "Collection of useful R utilities and snippets • inborutils - Mozilla Firefox". The address bar shows the URL "https://inbo.github.io/inborutils/index.html". The page content includes a navigation bar with "inbo R utils", "Tutorials", and "Functions". The main heading is "inborutils package". The text describes the package as a collection of useful R utilities and snippets. It mentions that some functions will be targeted to INBO usage itself, but some may be useful in a broader context. The "Installation" section provides instructions on how to install the package from github using the `install.packages("devtools")` and `devtools::install_github("inbo/inborutils")` commands. The right sidebar contains sections for "License" (MIT + file LICENSE), "Developers" (listing Stijn Van Hoey, Pieter Verschelde, and Thierry Onkelinx), and "Dev status" (with a link to Travis-CI Build Status).

Collection of useful R utilities and snippets • inborutils - Mozilla Firefox

Collection of useful R ut X +

https://inbo.github.io/inborutils/index.html

inbo R utils Tutorials Functions

## inborutils package

This `inborutils` package provides a collection of useful R utilities and snippets that we consider recyclable for multiple projects. These functions are either out of scope or just not mature enough to include as extensions to existing packages.

Some of these functions will be targeted to INBO usage itself, but maybe some of them are useful in a broader context and we would be happy to support the wider scientific community with our snippets.

## Installation

You can install `inborutils` from github with:

```
install.packages("devtools")
devtools::install_github("inbo/inborutils")
```

when successful, load it as usual:

## License

[MIT + file LICENSE](#)

## Developers

Stijn Van Hoey  
Author, maintainer  
(<http://orcid.org/0000-0001-6413-3185>)

Pieter Verschelde  
Author

Thierry Onkelinx  
Author  
(<http://orcid.org/0000-0001-8804-4216>)

## Dev status

[Travis-CI Build Status](#)

# Go in depth or a check under the hood?

Functions · Advanced R. - Mozilla Firefox

ons · Advanced R. x +

adv-r.had.co.nz/Functions.html#function-arguments

Advanced R by Hadley Wickham

Table of contents ▾

Want a physical copy of this material? [Buy a book from Amazon!](#)

## Contents

- Function components
- Lexical scoping
- Every operation is a function call
- Function arguments
- Special calls
- Return values
- Quiz answers

How to contribute

[Edit this page](#)

## Functions

Functions are a fundamental building block in this book, you need to already created many R functions. The focus of this chapter is to turn your understanding of what functions into techniques in this chapter, but not blocks for more advanced techniques.

The most important thing to understand is that you can work with them exactly as you will be explored in depth in functions. Quiz

Answer the following questions and provide answers at the end of the chapter.

1. What are the three components of a function?
2. What does the following code do?

Programming with R: The Call Stack - Mozilla Firefox

Programming with R: The Call Stack x +

https://swcarpentry.github.io/r-novice-inflammation/

Home Code of Conduct Setup Episodes ▾ Extras ▾ License Improve this page ✎

Search...

Programming with R

The Call Stack

Overview

Teaching: 15 min

Exercises: 0 min

Questions

- What is the call stack, and how does R know what order to do things in?







Zaal: Herman Teirlinck - 01.72 - Kaat Tilley

Datum: 2018-10-23, van 10:00 tot 12:00

*(registration announced via [DG\\_useR@inbo.be](mailto:DG_useR@inbo.be))*