

# Good programming practices Exercises

## Synthesis helpdesk workshop

Autumn 2021, Noëlle Schenk, Caterina Penone, Hugo Saiz

How the Exercise Session works : You will be given time to practice one (or several) of the eight following topics with a tutorial. If possible, work with your own code/ code snippets. During the whole session, you will have the possibility to ask us questions.

### Topic 1 : RStudio with Git and GitHub (intermediate)

1. Try to set up git and connect it to RStudio.  
You might want to follow the tutorial at : <https://happygitwithr.com/install-git.html>  
(especially from point 6 in the tutorial on)
2. Try to connect git and RStudio to GitHub  
Tutorial : <https://happygitwithr.com/install-git.html>  
(points 4 and 8 in the tutorial)

If you struggle to connect GitHub to an existing project :  
<https://hansenjohnson.org/post/sync-github-repository-with-existing-r-project/>

Common issues :

- Some issues with installing Git and GitHub with RStudio were discussed last winter in the [Slack chat from this workshop](#).

### Topic 2 : Unit testing (advanced)

*Requirement* : You should know or want to learn using R functions

Recommended for all code that goes further than reading in data and right away analyse it. It's e.g. recommended if you spend some time reformatting your data, imputing missing values, ...

Install the R package testthat

Nice blogpost to refresh what unit testing is : <https://towardsdatascience.com/unit-testing-in-r-68ab9cc8d211>

Tutorial with small reproducible examples : <https://katherinemwood.github.io/post/testthat/>

Another example of the use of testthat : <https://www.johndcook.com/blog/2013/06/12/example-of-unit-testing-r-code-with-testthat/>

### Topic 3 : How to write a reproducible example (all levels)

Good online tutorial : <http://adv-r.had.co.nz/Reproducibility.html>

### Topic 4 : Rmarkdown (all levels)

Jump-Start : Create a new Rmarkdown file and try around. **A newly created Rmarkdown automatically contains a mini-tutorial** : in RStudio > File > New File > Rmarkdown > a window pops up, enter the title of the document and your name.

Full introduction with text and videos : <https://rmarkdown.rstudio.com/lesson-1.html>

Quick Tour video (by RStudio) : [https://rmarkdown.rstudio.com/authoring\\_quick\\_tour.html](https://rmarkdown.rstudio.com/authoring_quick_tour.html)

### Topic 5 : Try out renv (advanced)

Renv is the R package that stores the current environment of your R project, i.e. all necessary packages and their versions.

Tutorial : <https://www.rstudio.com/blog/renv-project-environments-for-r/>

### Topic 6 : Try out Taxsize (intermediate)

Use examples: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3901538/>

Book taxsize manual : <https://taxize.dev/index.html>

### Topic 7 : Try out stringr (all levels)

stringr official tutorial : <https://stringr.tidyverse.org/>

Synthesis tutorial : [https://github.com/biodiversity-exploratories-synthesis/Synthesis\\_courses\\_workshops/blob/main/Help\\_desk\\_2021/2021\\_synthesis\\_Rhelpdesk\\_regex\\_and\\_text\\_processing.R](https://github.com/biodiversity-exploratories-synthesis/Synthesis_courses_workshops/blob/main/Help_desk_2021/2021_synthesis_Rhelpdesk_regex_and_text_processing.R)

## Topic 8 : Try out styler (all levels)

Install the styler package in RStudio.

Example use of styler with RStudio

1. copy a bad example of code from below
2. Select the piece of code you wish to style
3. Go to RStudio Menu bar above > Addins > search for "styler" > click "Style selection"

Bad examples of code

```
average<-mean(feet/12+inches,na.rm=TRUE)
```

```
OneFile <- c(1,2,3) # styler does not correct variable names in CamelCase
```

Apply styler on pieces of your own code.

- What aspects of your code change with styler?
- According to the [tidyverse style guide](#) : Which style recommendations do you already follow, and which not?
- Which of those you don't currently follow will you start following?

## Additional topics on demand

If you are interested in one of the below topics, or in anything else, feel free to practice it now and ask questions and/or have us adding it to the list of exercises.

- Splitting a programming problem into sub-problems
- data wrangling\* with tidyverse (e.g. [https://hbctraining.github.io/Intro-to-R/lessons/tidyverse\\_data\\_wrangling.html](https://hbctraining.github.io/Intro-to-R/lessons/tidyverse_data_wrangling.html))
- data wrangling\* with data.table

*\*Data wrangling, sometimes referred to as data munging, is the process of transforming and mapping data from one "raw" data form into another format with the intent of making it more appropriate and valuable for a variety of downstream purposes such as analytics. The goal of data wrangling is to assure quality and useful data. Data analysts typically spend the majority of their time in the process of data wrangling compared to the actual analysis of the data. Source : [Wikipedia](#)*