Project Setup*

- ▶ Create a new RStudio project (optional)
- ▶ Add **ALL** required files (data, code, documentation, etc.)
- ▶ Your project must be **standalone**

Research Compendium*

- ▶ Organize your files into subfolders (**`data`**, **`R`**, **`analyses`**, **`biblio`**, etc.)
- ▶ Document your project with a **`DESCRIPTION`** file[¶] (title, author, description, dependencies, etc.)
- ▶ Specify the System Requirements
- ▶ Add a **`LICENSE`**
- ▶ Use the R package **`{here}`** & relative paths instead of **`setwd()`**
- ▶ Use a **`make.R`** file to orchestrate your project

Versioning*

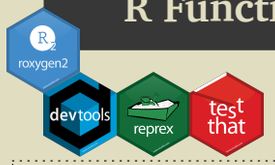
- ▶ Start versioning your code with **`git`**
- ▶ Share your project on GitHub (or GitLab) to collaborate
- ▶ Add a **`README`** to describe your project

Literate programming*

- ▶ Use the R package **`{rmarkdown}`** to integrate analyses, results, and textual elements in the same document (articles, vignettes, etc.)
- ▶ Use the R package **`{xaringan}`** to create your slides

[¶] With a **`DESCRIPTION`** file:
- ▶ Use **`remote::install_deps()`** instead of **`install.packages()`**
- ▶ Use **`devtools::load_all()`** instead of **`library()`** and **`source()`**

Write R Functions



- ▶ Do not duplicate your code and develop R functions (in the **`R/`** folder)
- ▶ Use **`devtools::load_all()`** instead of **`source()`**
- ▶ **Document** your code with the Roxygen syntax and use the function **`devtools::document()`**
- ▶ Add Unit Tests with the R package **`{testthat}`**
- ▶ Add Reproducible Examples

CI / CD

- ▶ Use services like Travis CI, Appveyor, and **`GitHub Actions`** to automate your workflow (checks, tests, deployment, etc.)

Build an R Package

- ▶ If you have a compendium and well-documented R functions building an R package is quite easy!
- ▶ Adapt your compendium to follow R packages structure
- ▶ Limit your dependencies
- ▶ Watch out the **`.Rbuildignore`**

Pipeline Toolkits

- ▶ If your analysis is time consuming, use Make-like R packages
- ▶ Only outdated computations are re-run: you are sure that your work is up-to-date
- ▶ **`{drake}`** is superseded: please use **`{targets}`**

Set Dependencies Versions

Create a Docker Image

Use **`DOCKER`** (or others) and...
- ▶ ... you will never have system compatibility issues
- ▶ ... your project will be 100% reproducible
- ▶ Take a look at the **`ROCKER`** initiative

The Degrees of Reproducibility with R

https://frbcesab.github.io/datatoolbox

Follow at least these steps to be a minimum reproducible

References:
- ▶ *Cooper N & Hsing PY (2017) A Guide to Reproducible Code in Ecology and Evolution. British Ecological Society.*
- ▶ *Gandrud C (2018) Reproducible Research with R and RStudio (2nd ed.). CRC Press.*
- ▶ *Marwick B et al. (2018) Packaging Data Analytical Work Reproducibly using R (and friends). PeerJ Preprints 6:e3192v2.*
- ▶ *https://ropensci.github.io/reproducibility-guide/sections/references*