R for Reproducibility

Daniela Palleschi

2024-02-29

Table of contents

Preface

> ⚠ Under construction
>
> This web-book will serve as a full-length version of the materials for the course 'Open Science Practices: Implementing a Reproducible Workflow in R'. Until it is up-and-running, the slides can be viewed at the following link: https://daniela-palleschi.github.io/r4repro_SoSe2024/

This web-book is the companion to the course 'Open Science Practices: Implementing a Reproducible Workflow in R' (and other iterations) given by Daniela Palleschi at the Humboldt-Universität zu Berlin. The tools discussed in this book are specific to the R enviornment, but the concepts are universal and programming language agnostic.

Aims of the book

- introduce students to basic concepts of reprodubility and replication in the scope of language research
- teach students how to implement a self-contained project-oriented analysis workflow
- help students develop good habits for project/data management
- teach students how to write up analyses for a paper or thesis using dynamic reports
- introduce students to concepts like version control and containerization

What you will not learn in this book

- how to analyse data
- how to fit models
- how to plot data
- how to design an experiment

About this book

This book was created using a Quarto Book RProject. The glossary for each chapter was created using the `glossary` package (DeBruine, 2023). The source code for each chapter is viewable by clicking the `<> Code` button at the top right of the page. The source code for the whole book is also available via GitHub, and is fully reproducible.

Part I

Conceptualisation

1 Open Science Practices

> ⚠ Under construction
>
> Coming soon! In the meantime, you can view the slides for this topic here (Open Science) and here (Replciation Crisis).

2 Reproducibility

> ⚠ Under construction
>
> Coming soon! In the meantime, you can view the slides for this topic here (Reproducibility).

3 Documentation

> ⚠ Under construction
>
> Coming soon!

Part II

Basic workflow

4 RProjects

An ode to self-contained projects

> **Under construction**
>
> Coming soon!In the meantime, you can view the slides for this topic here (Rprojects).

5 Folder structure

> **Under construction**
>
> Coming soon!

6 Writing reproducible code

Under construction

Coming soon!In the meantime, you can view the slides for this topic here (Writing reproducible code) and here (Data wrangling).

7 Code review

> ⚠ Under construction
>
> Coming soon!

Part III

Research workflow

8 Pre-registration

> **Under construction**
>
> Coming soon!

9 Version control

> **Under construction**
>
> Coming soon!

10 Storing results

> ⚠ Under construction
>
> Coming soon!

Part IV

Writing

11 Publishing your analyses

> **Under construction**
>
> Coming soon!

12 Writing it up

> **Under construction**
>
> Coming soon!

13 Collaboration

> **Under construction**
>
> Coming soon!

14 Summary

> **Under construction**
>
> Coming soon!

Part V

Exercises

15 Exercises

> ⚠ Under construction
>
> Coming soon!

References

DeBruine, L. (2023). *Glossary: Glossaries for markdown and quarto documents.* https://github.com/debruine/glossary

A Glossary

```
library(tidyverse)
library(glossary)
glossary_path(here::here("glossary/glossary.yml"))
# glossary_persistent(TRUE)
```

```
x <-
  readr::read_delim(here::here("glossary/glossary.yml"), col_names = F, delim = "|") |>
  mutate(col = rep(c("term", "definition"), times = 4)) |>
  mutate(X1 = str_replace(X1, ': ', '')) |>
  mutate(X1 = str_trim(X1)) |>
  mutate(col = as.factor(col)) |>
  select(-X2)

glossary_table() |> as_tibble()
```

```
# A tibble: 0 x 0
```

B Project setup

B.1 CRAN packages

You don't necessarily need to keep track of these, as usually RStudio will automatically inform you of missing packages used in an Rmd or qmd script (e.g., after updating R). But still useful to keep track

```
install.packages("remotes")
install.packages("devtools")
```

B.2 Developer packages

B.2.1 Glossary

Set-up glossary using the `glossary` package (**glossary?**). For now I'll use an as-yet-unreleased version, which includes terms that share first words with other terms (e.g., `replication` and `replication crisis`).

```
# install.packages("glossary")
remotes::install_github("https://github.com/debruine/glossary")
```

```
library(glossary)
glossary_path("glossary/glossary.yml")
```

Add a glossary term.

```
glossary_add(term = "power",
             def = "The probability of rejecting the null hypothesis when it is false, for a
)
```

Set my preferred glossary theme.

```
glossary_popup("click")

glossary_style(color = "purple",
               text_decoration = "underline",
               def_bg = "#333",
               def_color = "white")
```

```
# append default styles to an external CSS file
write(glossary_style(), "glossary/glossary.css", append = TRUE)
```

B.3 Workflow

B.3.1 renv

Initialise a lockfile and take a snapshot.

Listing B.1 `in the Console`

```
# initialise
renv::init()

# take snapshot
renv::snapshot()
```

If it's been a while since you're updated your packages, you can update them all with:

```
renv::hydrate()

# or
renv::update()
```

If you've recently updated R, you'll need to re-install your packages. You can simply restore your lockfile.

```
# restore to your most recent package versions
renv::restore()
```

B.3.2 rbbt

Install rbbt. N.B., if you're not in a remote RProject and are using renv, this might return an error. To mitigate this, try running `renv::deactivate`, and `renv::activate()` to activate it again.

```
devtools::install_github("paleolimbot/rbbt")
```