A primer on setting up projects in R for reproducibility

Frauke Stehr

5/26/2022

Contents

Introduction and disclaimer

During my studies, we had to submit various data-analysis assignments, but no-one ever showed us how to set up projects efficiently. My later work experience, both in the industry as well as in research, showed me just *how* important it is to set up your project efficiently from the start, so you can collaborate easily. An additional advantage is that it increases the reproducibility of your project, something that should be of clear value to any researcher, but will become important at the latest when you are submitting your paper to a journal.

In this primer, I try to write down how I eventually learned to structure my data projects using projects in R, adding privacy-conform version control with git, and writing reproducible papers and slides in R Markdown. Of course, one never stops to learn, so if you have an idea on how to improve the work-flow even further, please drop me an e-mail.

Disclaimer: Of course, there are numerous guides for any of these steps available online. All of them are much more detailed, focusing on one specific aspect of the project management process. When I started searching for sources on how to set up projects, I felt a bit overwhelmed by the abundance of resources. Therefore I decided to write my own short primer, which is more intended to give help you have a quick start rather than getting deep understanding of all the possibilities.

Setting up the project and folder structure

Why use R Projects and think about the folder structure?

A folder structure that works for me

Adding version control with git/github

Why add version control?

How to synchronize your project to your github account

What about privacy, e.g., GDPR?

Writing papers/slides in R Markdown

Why write papers/slides in R Markdown?

There are multiple reasons why you should want to write your reports in Markdown. First of all, you'll never have to copy and paste p-values, regression tables and graphs into your drafts again. You just calculate them in your paper. If the data changes, so does the p-value in your paper. No need to go through the paper thousand times to check that all values are the ones reflecting the current state of your analysis - nice! Second, Markdown as a syntax is designed for good *writing flow*, so all formatting goes much quicker than e.g., in LaTex or some WYSIWYG text editors. Plus, you can still use Latex's citation and math packages, so you don't need to get used to too much new syntax.

You can **knit** (in LaTex lingo *compile*) your Markdown document into many different output files. The most useful for research projects are LaTex-based pdfs (find the same document knit into a pdf here: link), and beamer slides (output as pdf) or ioslides (output as html). I currently prefer ioslides as of now, as 1) it is a widely operable file format, 2) it allows for animations using javascript, and 3) I can write all additional stylings in one .css file, keeping my Markdown script nice and tidy.

The basic syntax

Basic text formatting

Here I describe shortly some of the syntax you might want to use, mostly to show why Markdown is such an elegant syntax for writing. For a more comprehensive description, I recommend Section 2.5 of Xie, Allaire, and Grolemund (2022) 's RMarkdown guide.

Here are a couple of styling options you might find useful in the beginning:

- You can write in *italics* like so `*italics*`.

- For **bold**-face text use two asterisks.
- For both ***bold and italics*** use three asterisks.
- For `in-line code` use a pair of backticks, e.g. `` `code here` ``.
- Headings are started by a #. For a sub-header just add another #.

```
# This is a header
## This is a sub-header
### This is a sub-sub-header

- this is a list
- this is a second item
```

Including R chunks

You can embed an R code chunk by typing the chunk delimiters ```` ```{r} ```` and ```` ``` ````. For example, you can print a summary of your data like this:

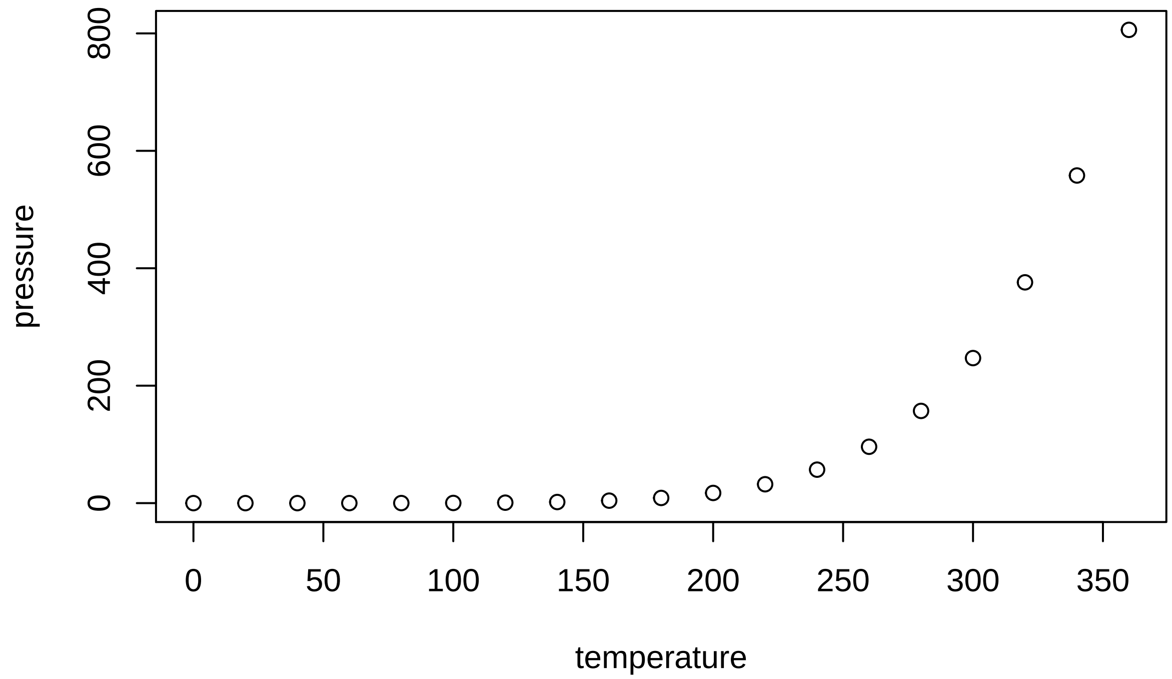
```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

More functions are described here.

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Including citations with bibtex

If you are used to manage citations in bibtex, you'll be happy to hear that this is also possible in Markdown. Include a `references.bib` in the folder that contains the Markdown file. You can then include citations like so `@xie` or `[@xie]` which renders to Xie, Allaire, and Grolemund (2022) and (Xie, Allaire, and Grolemund 2022), respectively.

Additional sources

How to set up an R project https://intro2r.com/rsprojs.html

Github integration https://intro2r.com/github_r.html

Rmarkdown setup https://intro2r.com/rmarkdown_r.html

darin ist auch dieses sehr detaillierte Buch vom Entwickler verlinkt: https://bookdown.org/yihui/rmarkdown/

References

Xie, Yihui, Joseph J Allaire, and Garrett Grolemund. 2022. *R Markdown: The Definitive Guide*. Chapman; Hall/CRC. https://bookdown.org/yihui/rmarkdown/.