Collaborative Data Science Practices

Will Beasley

2019-01-15

Contents

Chapter 1

Prerequisites

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

The **bookdown** package can be installed from CRAN or Github:

```
install.packages("bookdown")
# or the development version
# devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading `#`.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): https://yihui.name/tinytex/.

Chapter 2

Architecture Principles

2.1 Encapsulation

2.2 Leverage team member's strenghts & avoid weaknesses

1. Focused code files
2. Metadata for content experts

2.3 Scales

1. Single source & single analysis
2. Multiple sources & multiple analyses

2.4 Consistency

1. Across Files {#consistency-files}
2. Across Languages
3. Across Projects

Chapter 3

Prototypical File

As stated before, in **??**, using a consistent file structure can (a) improve the quality of the code because the structure has been proven over time to facilitate good practices and (b) allow your intentions to be more clear to teammates because they are familiar with the order and intentions of the chunks.

We use the term "chunk" for a section of code because it corresponds with knitr terminology (Xie, 2015), and in many cases, the chunk of our R file connects to a knitr Rmd file.

3.1 Clear Memory

3.2 Load Sources

3.3 Load Packages

3.4 Declare Globals

3.5 Load Data

3.6 Tweak Data

3.7 (Unique Content)

3.8 Verify Values

3.9 Specify Output Columns

3.10 Save to Disk or Database

Chapter 4

Prototypical Repository

https://github.com/wibeasley/RAnalysisSkeleton

4.1 Analysis

4.2 Data Public

1. Raw
2. Derived
3. Metadata
4. Database
5. Original

4.3 Data Unshared

4.4 Documentation

4.5 Manipulation

4.6 Stitched Output

4.7 Utility

Chapter 5

Data at Rest

5.1 Data States

1. Raw
2. Derived
    1. Project-wide File on Repo
    2. Project-wide File on Protected File Server
    3. User-specific File on Protected File Server
    4. Project-wide Database
3. Original

5.2 Data Containers

1. csv
2. rds
3. SQLite
4. Central Enterprise database
5. Central REDCap database
6. Containers to avoid for raw/input
    1. Proprietary like xlsx, sas7bdat

Chapter 6

Patterns

6.1 Ellis

6.2 Arch

6.3 Ferry

6.4 Scribe

6.5 Analysis

6.6 Presentation -Static

6.7 Presentation -Interactive

6.8 Metadata

Chapter 7

Security & Private Data

7.1 File-level permissions

7.2 Database permissions

7.3 Public & Private Repositories

1. Scrubbing GitHub history

Chapter 8

Automation

8.1 Flow File in R

8.2 Makefile

8.3 SSIS

8.4 cron Jobs & Task Scheduler

8.5 Sink Log Files

Chapter 9

Scaling Up

9.1 Data Storage

1. Local File vs Conventional Database vs Redshift
2. Usage Cases

9.2 Data Processing

1. R vs SQL
2. R vs Spark

Chapter 10

Parallel Collaboration

10.1 Social Contract

1. Issues
2. Organized Commits & Coherent Diffs
3. Branch & Merge Strategy

10.2 Code Reviews

1. Daily Reviews of PRs
2. Periodic Reviews of Files

10.3 Remote

1. Headset & sharing screens

Chapter 11

Documentation

11.1 Team-wide

11.2 Project-specific

11.3 Dataset Origin & Structure

11.4 Issues & Tasks

11.5 Flow Diagrams

11.6 Setting up new machine

(example)

Chapter 12

Publishing Results

12.1 To Other Analysts

12.2 To Researchers & Content Experts

12.3 To Technical-Phobic Audiences

Chapter 13

Testing, Validation, & Defensive Programming

13.1 Testing Functions

13.2 Defensive Programming

1. Throwing errors

13.3 Validator

1. Benefits for Analysts
2. Benefits for Data Collectors

Chapter 14

Troubleshooting and Debugging

14.1 Finding Help

1. Within your group (eg, Thomas and REDCap questions)
2. Within your university (eg, SCUG)
3. Outside (eg, Stack Overflow; GitHub issues)

14.2 Debugging

1. `traceback()`, `browser()`, etc

Chapter 15

Considerations when Selecting Tools

https://github.com/OuhscBbmc/RedcapExamplesAndPatterns/blob/master/DocumentationGlobal/ResourcesInstallation.md

15.1 Required Installation

15.2 Recommended Installation

15.3 Optional Installation

15.4 Asset Locations

Chapter 16

Considerations when Selecting Tools

16.1 General

16.1.1 The Component's Goal

While disussing the advantages and disadvanages of tools, a colleague once said, "Tidyverse packages don't do anything that I can't already do in Base R, and sometimes it even requires more lines of code". Regardless if I agree, I feel these two points are irrelevant. Sometimes the advantage of a tool isn't to expand existing capabilities, but rather to facilitate development and maintaince for the same capability.

Likewise, I care less about the line count, and more about the readability. I'd prefer to maintain a 20-line chunk that is familar and readable than a 10-line chunk with dense phrases and unfamiliar functions. The bottleneck for most of our projects is human time, not execution time.

16.1.2 Current Skillset of Team

16.1.3 Desired Future Skillset of Team

16.1.4 Skillset of Audience

16.2 Languages

16.3 R Packages

16.4 Database

Chapter 17

Growing a Team

17.1 Recruiting

17.2 Training to Data Science

1. Starting with a Researcher
2. Starting with a Statistician
3. Starting with a DBA
4. Starting with a Software Developer

17.3 Bridges Outside the Team

1. Monthly User Groups
2. Annual Conferences

Chapter 18

Introduction

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter 18. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter 2.

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 18.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 18.1.

```
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

You can write citations, too. For example, we are using the **bookdown** package (Xie, 2018) in this sample book, which was built on top of R Markdown and **knitr** (Xie, 2015).

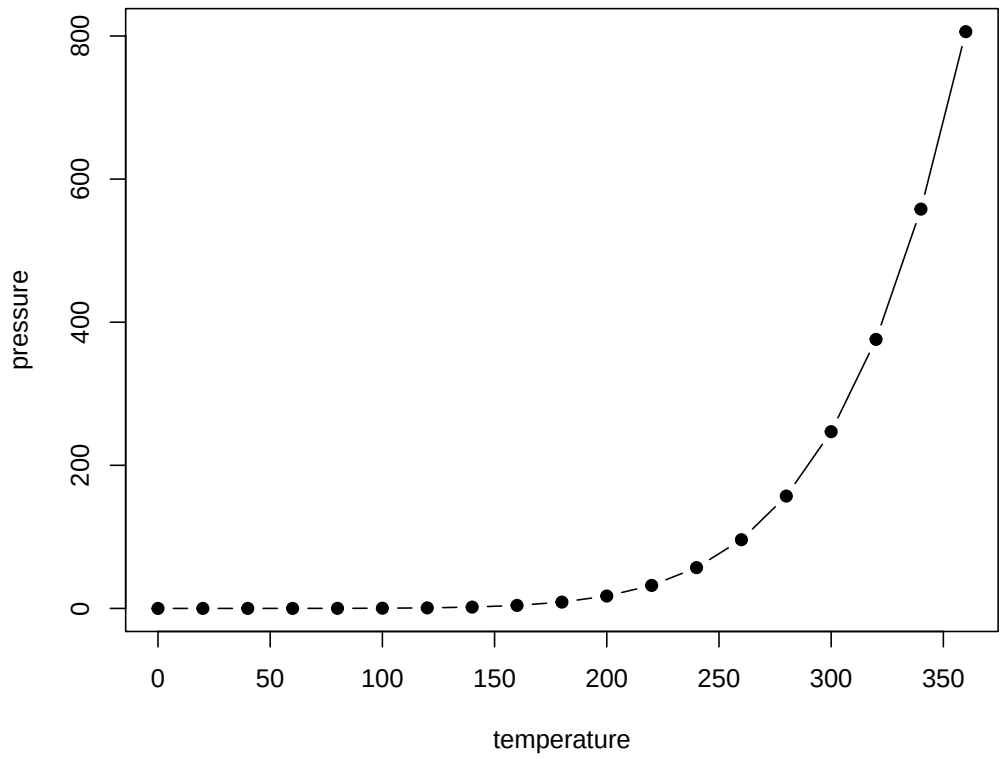

Figure 18.1: Here is a nice figure!

Table 18.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

Chapter 19

Scratch Pad of Loose Ideas

19.1 Chapters & Sections to Form

1. Tools to Consider
    1. tidyverse
    2. odbc

Bibliography

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2018). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.9.