Collaborative Data Science Practices

Will Beasley

2018-10-17

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

1. Encapsulation
2. Leverage team member's strenghts & avoid weaknesses
    1. Focused code files
    2. Metadata for content experts
3. Scales
    1. Single source & single analysis
    2. Multiple sources & multiple analyses
4. Consistency
    1. Across Files
    2. Across Languages
    3. Across Projects

Chapter 3

Prototypical File

1. Clear Memory
2. Load Sources
3. Load Packages
4. Declare Globals
5. Load Data
6. Tweak Data
7. (Unique Content)
8. Verify Values
9. Specify Output Columns
10. Save to Disk or Database

Chapter 4

Data at Rest

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
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 5

Patterns

1. Ellis
2. Arch
3. Ferry
4. Scribe
5. Analysis
6. Presentation -Static
7. Presentation -Interactive
8. Metadata

Chapter 6

Security & Private Data

1. File-level permissions
2. Database permissions
3. Public & Private Repositories
    1. Scrubbing GitHub history

Chapter 7

Automation

1. Flow File in R
2. Makefile
3. SSIS
4. cron Jobs & Task Scheduler
5. sink log files

Chapter 8

Scaling Up

1. Data Storage
    1. Local File vs Conventional Database vs Redshift
    2. Usage Cases
2. Data Processing
    1. R vs SQL
    2. R vs Spark

Chapter 9

Parallel Collaboration

1. Social Contract
    1. Issues
    2. Organized Commits & Coherent Diffs
    3. Branch & Merge Strategy
2. Code Reviews
    1. Daily Reviews of PRs
    2. Periodic Reviews of Files
3. Remote
    1. Headset & sharing screens

Chapter 10

Documentation

1. Team-wide
2. Project-specific
3. Dataset origin & structure
4. Issues & Tasks
5. Flow Diagrams
6. Setting up new machine (example)

Chapter 11

Publishing Results

1. To Other Analysts
2. To Researchers & Content Experts
3. To Technical-Phobic Audiences

Chapter 12

Testing, Validation, & Defensive Programming

1. Testing Functions
2. Defensive Programming
    1. Throwing errors
3. Validator
    1. Benefits for Analysts
    2. Benefits for Data Collectors

Chapter 13

Troubleshooting and Debugging

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

Chapter 14

Considerations when Selecting Tools

1. General
    1. The Component's Goal
    2. Current Skillset of Team
    3. Desired Future Skillset of Team
    4. Skillset of Audience
2. Languages
3. R Packages
4. Database

Chapter 15

Growing a Team

1. Recruiting
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
3. Bridges Outside the Team
    1. Monthly User Groups
    2. Annual Conferences

Chapter 16

Introduction

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter 16. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter **??**.

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

```r
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 16.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 16.1.

```r
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).

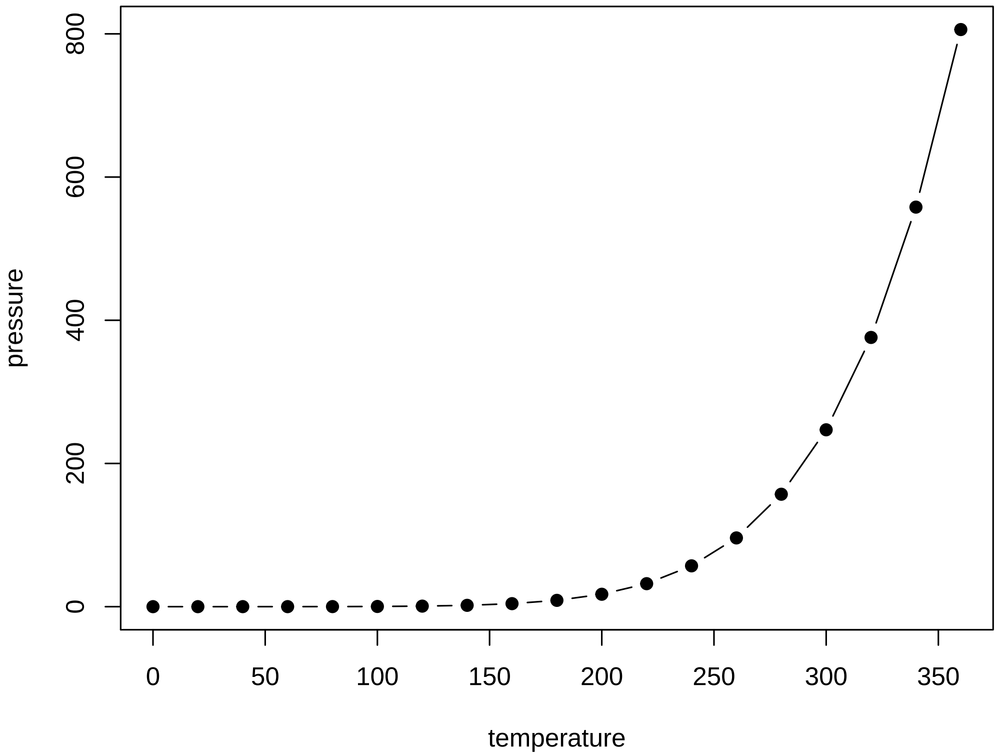

Figure 16.1: Here is a nice figure!

Table 16.1: 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 17

Scratch Pad of Loose Ideas

17.1 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 0.7.