Data Technologies

Computational tools for working with data

Martin Frigaard

2023-06-07

Table of contents

Welcome!

This book describes the essential tools and techniques for working with data, especially those 'behind-the-scenes' tasks that will take considerable time and effort but don't receive much attention.

Code

This section covers what code is, how it's stored, commonly used languages for working with data, the command-line interface, regular expressions, and computational notebooks.

Object-oriented programming

In this section covers the object-oriented programming (OOP) paradigm, common features, and why OOP languages are useful for dealing with data. Two languages (R and Python) are introduced as examples to compare and contrast.

Functions

Data

Git

Preface

This book is focused on computational tools for working with data. Data science is a rapidly evolving field, so in many ways, this text is an attempt to capture whatever 'best practices' can be codified or generalized.

The topics include coding languages, computational documents, data storage and file formats, version control systems, and general computer science topics.

This text emphasizes using tools that require typing code on a keyboard (rather than using a mouse to point and click).

This book was written in Quarto (learn more about Quarto books).

Introduction

> *This is a book created from markdown and executable code. See Knuth (1984) for additional discussion of literate programming.*

I've left the boilerplate text above because 1) everyone should know about Donald Knuth and his contributions to computational notebooks, and 2) I wanted to keep a working example for the `references.qmd` and `references.bib` files when I build the book.

How the sausage is made

I've included the steps for publishing this book *inside* the book because it's serves as an excellent example of the topics contained in this text. This book is written in quarto which is an '*open-source scientific and technical publishing system.*' All of the tools in this text are open-source, which means the source code (i.e. files) to create it are '*made freely available for possible modification and redistribution.*'

The code files used to create this book are displayed in the folder tree below:

```
.
+-- _book
+-- _freeze
+-- _quarto.yml
+-- cover.png
+-- data-tech.Rproj
+-- files.qmd
+-- fun-arguments.qmd
+-- fun-environments.qmd
+-- fun-names.qmd
+-- fun-structure.qmd
+-- index.html
+-- index.qmd
+-- intro.html
+-- intro.qmd
+-- intro.rmarkdown
+-- intro_files
```

```
+-- notebooks.qmd
+-- oop-basics.qmd
+-- oop-in-r.qmd
+-- plain-text.qmd
+-- preface.html
+-- preface.qmd
+-- references.bib
+-- references.qmd
+-- renv
+-- renv.lock
\-- site_libs
```

These files are stored in a GitHub repository. GitHub is a platform for hosting open source projects that use Git, the world's most popular distributed version control system.

Using Git

Basic knowledge of Git has become somewhat necessary when you decide to enter the data technology ecosystem (or other open-source projects). I won't be diving into the Git workflow here, but I *will* cover the commands I used to store and publish this book.

1. I created a repository on GitHub.com like this one: mjfrigaard/data-tech
2. Add (`-A`) and commit (`commit -m`) the files in the book folder:

```
$ git add -A
$ git commit -m "first commit"
```

3. `push` local files to GitHub repo

```
$ git remote add origin git@github.com:mjfrigaard/data-tech.git
$ git branch -M main
$ git push -u origin main
```

4. Create (checkout) an empty `gh-pages` branch

```
$ git checkout --orphan gh-pages
Switched to a new branch 'gh-pages'
$ git reset --hard
$ git commit --allow-empty -m "Initialising gh-pages branch"
Initialising gh-pages branch
```

5. `push` the book files to the `gh-pages` branch

```
$ git push origin gh-pages
Enumerating objects: 2, done.
Counting objects: 100% (2/2), done.
Writing objects: 100% (2/2), 176 bytes | 176.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'gh-pages' on GitHub by visiting:
remote:      https://github.com/mjfrigaard/data-tech/pull/new/gh-pages
remote:
To github.com:mjfrigaard/data-tech.git
 * [new branch]      gh-pages -> gh-pages
```

5. Switch (`checkout`) back to `main` branch

```
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

6. `publish` to `gh-pages` branch

```
$ quarto publish gh-pages
? Update site at https://mjfrigaard.github.io/data-tech/? (Y/n)
```

```
? Update site at https://mjfrigaard.github.io/data-tech/? (Y/n) Yes
```

Part I

Code

Writing code

Language is a technology that gives us the ability to express our ideas with precision. Programming languages (i.e., 'code') are comprised of the statements (grammar and syntax) we use to communicate our intentions to a computer.

> ***"The code that translates a web of conceptual relations in our heads into an early-to-late order in our mouths, or into a left-to-right order on the page, is called syntax. The rules of syntax, together with the rules of word formation (the ones that turn 'kill' into 'kills', 'killed', and 'killing'), make up the grammar of English. Different languages have different grammars, but they all convey conceptual relationships by modifying and arranging words." - The Sense of Style, Steven Pinker (2014)***

Grammar: the rules for how a language uses symbols (words, numbers, punctuation, etc.).

Syntax: how to arrange words and phrases into sentences to create meaning.

Code must conform with the rules of programming language's syntax and grammar to execute correctly.

Plain text

Open-source code is written in plain text files.

Machine readable

In order to work, the machine (i.e. computer) must be able to execute the code you've written.

Human readable

Code is always read by at least two people: you, and future you.

HTML

Hypertext markup language

Markdown

R

Python

JavaScript

Code files

The command line

The command line is...

Regular expressions

Computational notebooks

Part II

Object-oriented programming

Basics

Object-oriented programming is...

R

R is a ‘functional, object-oriented programming language.’

Python

Part III

Functions

Names

Naming your functions should...

Arguments

Function arguments can be

Structure

Function structure refers to...

Environments

Inside your function...

Knuth, Donald E. 1984. "Literate Programming." *Comput. J.* 27 (2): 97–111. https://doi.org/10.1093/comjnl/27.2.97.