# Using R Packages for Reproducible Workflows

Michael Dumelle

September 22, 2021

# Contents

# Overview

Welcome to the 2021 EPA R Workshop titled “Using R Packages for Reproducible Workflows” by me, Michael Dumelle – I am glad to have you here! Before proceeding, let’s first start up R and download the devtools package.

```r
install.packages("devtools") # if required
```

The workshop’s companion R package (Using **R** **P**ackages for **R**eproducible **W**orkflows) is available for download via

```r
devtools::install_github(repo = "michaeldumelle/RPRW", ref = "main")
library(RPRW)
```

Here are the sections of this workshop:

1. Building an R Package
2. A Research Compendia for an overview of an effective research compendia
3. Turning An R Package into a Reproducible Research Compendia
4. Extensions
5. Exercise Solutions

# Chapter 1

# Building an R Package

## 1.1 What is an R Package?

An R package is a collection of code, data, documentation, and tests with a particular structure that can be shared with others. R packages are commonly downloaded from the Comprehensive R Archive Network (CRAN). You can install them from CRAN with `install.packages("package_name")`, load them in your workspace with `library("package_name")`, and get help by running (`help(package = "package_name")`).

One of the reasons R packages are so useful is because they are the fundamental way to share code in R. If your code is in a package, others can easily download and use it. If they are familiar with R packages, they likely will be familiar with how to use yours! But sharing R code is not the only benefit of creating R packages. Learning how to build an R package will provide several other benefits to future you!

Future you will benefit from creating your own R packages because they enforce a particular structure. This structure

1. Saves you time – you don't need to think about how to organize your files, R packages have a template!
    - This was especially helpful for me because before learning how to create R packages, I would save my R files in all sorts of locations on my computer with all sorts of names. This made it *very challenging* to come back to my work later and find a particular file.
2. Gives you standardized tools – people have created extremely useful tools that work with R packages, so take advantage of them!
    - The R package devtools, which we downloaded earlier, contains many of these standarized tools.
3. Requires documentation – This is especially helpful for future you.

    - Before I started using R packages, when I would come back my old code, I was convinced someone else wrote it – I basically had to rewrite it all to understand it. R packages help prevent this.
4. Is reproducible – R packages are built from R projects (see here and here), so file paths are relative, not absolute!
    - `read_csv("a_fun_csv_file.csv")` works on my machine – and yours!
    - While R projects are not the fundamental focus on today, I highly, highly recommend you use them for every data analysis project that you are not using an R package for.
5. Guides your data analysis – We will talk about this today
    - See Marwick et al. [2018] for more!

### 1.1.1 Exercises

1. What are some of your favorite R packages?

2. Of those we have talked about so far, what benefits of R packages are most appealing to you?

## 1.2 Creating an R Package

### 1.2.1 The Motivating Dataset

Suppose we want to build an R package that summarizes length (in kilometers) and discharge (meters per second cubed) of North American rivers based on the names of the rivers. Below is our data of interest

```
rivers <- data.frame(
  Missouri = c(3768, 1956),
  Mississippi = c(3544, 18400),
  Yukon = c(3190, 6340),
  Colorado = c(2330, 40),
  Arkansas = c(2322, 1004),
  Columbia = c(2000, 7730),
  Red = c(1811, 852),
  Canadian = c(1458, 174)
)
rownames(rivers) <- c("length", "discharge")
rivers
#>           Missouri Mississippi Yukon Colorado Arkansas Columbia  Red Canadian
#> length        3768        3544  3190     2330     2322     2000 1811     1458
#> discharge     1956       18400  6340       40     1004     7730  852      174
```

## 1.3 Additional Resources

- R Packages by Hadley Wickham and Jenny Bryan
- Writing an R package from scratch by Hilary Parker

# Chapter 2

# Cross-references

Cross-references make it easier for your readers to find and link to elements in your book.

## 2.1 Chapters and sub-chapters

There are two steps to cross-reference any heading:

1. Label the heading: `# Hello world {#nice-label}`.
   - Leave the label off if you like the automated heading generated based on your heading title: for example, `# Hello world` = `# Hello world {#hello-world}`.
   - To label an un-numbered heading, use: `# Hello world {-#nice-label}` or `{# Hello world .unnumbered}`.
2. Next, reference the labeled heading anywhere in the text using `\@ref(nice-label)`; for example, please see Chapter 2.
   - If you prefer text as the link instead of a numbered reference use: any text you want can go here.

## 2.2 Captioned figures and tables

Figures and tables *with captions* can also be cross-referenced from elsewhere in your book using `\@ref(fig:chunk-label)` and `\@ref(tag:chunk-label)`, respectively.

See Figure **??**.

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

Don't miss Table **??**.

```r
knitr::kable(
  head(pressure, 10), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

# Chapter 3

# Parts

You can add parts to organize one or more book chapters together. Parts can be inserted at the top of an .Rmd file, before the first-level chapter heading in that same file.

Add a numbered part: `# (PART) Act one {-}` (followed by `# A chapter`)

Add an unnumbered part: `# (PART\*) Act one {-}` (followed by `# A chapter`)

Add an appendix as a special kind of un-numbered part: `# (APPENDIX) Other stuff {-}` (followed by `# A chapter`). Chapters in an appendix are prepended with letters instead of numbers.

# Chapter 4

# Footnotes and citations

## 4.1 Footnotes

Footnotes are put inside the square brackets after a caret `^[]`. Like this one [1].

## 4.2 Citations

Reference items in your bibliography file(s) using `@key`.

For example, we are using the **bookdown** package [Xie, 2021] (check out the last code chunk in index.Rmd to see how this citation key was added) in this sample book, which was built on top of R Markdown and **knitr** [Xie, 2015] (this citation was added manually in an external file book.bib). Note that the `.bib` files need to be listed in the index.Rmd with the YAML `bibliography` key.

The RStudio Visual Markdown Editor can also make it easier to insert citations: https://rstudio.github.io/visual-markdown-editing/#/citations

[1] This is a footnote.

# Chapter 5

# Blocks

## 5.1 Equations

Here is an equation.

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \tag{5.1}$$

You may refer to using `\@ref(eq:binom)`, like see Equation (5.1).

## 5.2 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref(thm:tri)`, for example, check out this smart theorem 5.1.

**Theorem 5.1.** *For a right triangle, if $c$ denotes the* length *of the hypotenuse and $a$ and $b$ denote the lengths of the* ***other*** *two sides, we have*

$$a^2 + b^2 = c^2$$

Read more here https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html.

## 5.3 Callout blocks

The R Markdown Cookbook provides more help on how to use custom blocks to design your own callouts: https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html

# Chapter 6

# Sharing your book

## 6.1 Publishing

HTML books can be published online, see: https://bookdown.org/yihui/bookdown/publishing.html

## 6.2 404 pages

By default, users will be directed to a 404 page if they try to access a webpage that cannot be found. If you'd like to customize your 404 page instead of using the default, you may add either a `_404.Rmd` or `_404.md` file to your project root and use code and/or Markdown syntax.

## 6.3 Metadata for sharing

Bookdown HTML books will provide HTML metadata for social sharing on platforms like Twitter, Facebook, and LinkedIn, using information you provide in the `index.Rmd` YAML. To setup, set the `url` for your book and the path to your `cover-image` file. Your book's `title` and `description` are also used.

This `gitbook` uses the same social sharing data across all chapters in your book-all links shared will look the same.

Specify your book's source repository on GitHub using the `edit` key under the configuration options in the `_output.yml` file, which allows users to suggest an edit by linking to a chapter's source file.

Read more about the features of this output format here:

https://pkgs.rstudio.com/bookdown/reference/gitbook.html

Or use:

```
?bookdown::gitbook
```

# Bibliography

Ben Marwick, Carl Boettiger, and Lincoln Mullen. Packaging data analytical work reproducibly using r (and friends). *The American Statistician*, 72(1): 80–88, 2018.

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. URL http://yihui.org/knitr/. ISBN 978-1498716963.

Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*, 2021. URL https://CRAN.R-project.org/package=bookdown. R package version 0.23.