

R 4 Epidemiology

2023-12-15

Table of contents

Welcome	3
Acknowledgements	3
Introduction	4
Goals	4
Text conventions used in this book	5
Other reading	5
Contributing	6
Typos	6
I Coding Tools and Best Practices	13
1 Quarto Files	14
II Collaboration	15
2 Using git and GitHub	16
III References	17
References	18
Appendices	19
A Glossary	19

Welcome

Welcome to R for Epidemiology!

This electronic textbook was originally created to accompany the Introduction to R Programming for Epidemiologic Research course at the [University of Texas Health Science Center School of Public Health](#). However, we hope it will be useful to anyone who is interested in R, epidemiology, or human health and well-being.

Acknowledgements

This book is currently a work in progress (and probably always will be); however, there are already many people who have played an important role (some unknowingly) in helping develop it thus far. First, we'd like to offer our gratitude to all past, current, and future members of the R Core Team for maintaining this *amazing, free* software. We'd also like to express our gratitude to everyone at [Posit](#). You are also developing and *giving away* some amazing software. In particular, we'd like to acknowledge [Garrett Grolemund](#) and [Hadley Wickham](#). Both have had a huge impact on how we use and teach R. We'd also like to thank our students for all the feedback they've given us while taking our courses. In particular, we want to thank [Jared Wiegand](#) and Yiqun Wang for their many edits and suggestions.

This electronic textbook was created and published using [R](#), [RStudio](#), the [Quarto](#), and [GitHub](#).

Introduction

Goals

We’re going to start the introduction by writing down some basic goals that underlie the construction and content of this book. We’re writing this for you, the reader, but also to hold ourselves accountable as we write. So, feel free to read if you are interested or skip ahead if you aren’t.

The goals of this book are:

1. **To teach you how to use R and RStudio as tools for applied epidemiology.**¹
Our goal is not to teach you to be a computer scientist or an advanced R programmer. Therefore, some readers who are experienced programmers may catch some technical inaccuracies regarding what we consider to be the fine points of what R is doing “under the hood.”
2. **To make this writing as accessible and practically useful as possible without stripping out all of the complexity that makes doing epidemiology in real life a challenge.** In other words, We’re going to try to give you all the tools you need to *do* epidemiology in “real world” conditions (as opposed to ideal conditions) without providing a whole bunch of extraneous (often theoretical) stuff that detracts from *doing*. Having said that, we will strive to add links to the other (often theoretical) stuff for readers who are interested.
3. **To teach you to accomplish common *tasks*,** rather than teach you to use functions or families of functions. In many R courses and texts, there is a focus on learning all the things a function, or set of related functions, can do. It’s then up to you, the reader, to sift through all of these capabilities and decided which, if any, of the things that *can* be done will accomplish the tasks that you are *actually trying* to accomplish. Instead, we will strive to start with the end in mind. What is the task we are actually trying to accomplish? What are some functions/methods we could use to accomplish that task? What are the strengths and limitations of each?

¹In this case, “tools for applied epidemiology” means (1) understanding epidemiologic concepts; and (2) completing and interpreting epidemiologic analyses.

4. **To start each concept by showing you the end result** and then deconstruct how we arrived at that result, where possible. We find that it is easier for many people to understand new concepts when learning them as a component of a final product.
5. **To learn concepts with data** instead of (or alongside) mathematical formulas and text descriptions, where possible. We find that it is easier for many people to understand new concepts by seeing them in action.

Text conventions used in this book

- We will hyperlink many keywords or phrases to their [glossary](#) entry.
- Additionally, we may use **bold** face for a word or phrase that we want to call attention to, but it is not necessarily a keyword or phrase that we want to define in the glossary.
- **Highlighted inline code** is used to emphasize small sections of R code and program elements such as variable or function names.

Other reading

If you are interested in R4Epi, you may also be interested in:

- [Hands-on Programming with R](#) by Garrett Golemund. This book is designed to provide a friendly introduction to the R language.
- [R for Data Science](#) by Hadley Wickham, Mine Çetinkaya-Rundel, and Garrett Golemund. This book is designed to teach readers how to do data science with R.
- [Statistical Inference via Data Science: A ModernDive into R and the Tidyverse](#). This book is designed to be a gentle introduction to the practice of analyzing data and answering questions using data the way data scientists, statisticians, data journalists, and other researchers would.
- [Reproducible Research with R and RStudio](#) by Christopher Gandrud. This book gives you tools for data gathering, analysis, and presentation of results so that you can create dynamic and highly reproducible research.
- [Advanced R](#) by Hadley Wickham. This book is designed primarily for R users who want to improve their programming skills and understanding of the language.

Contributing

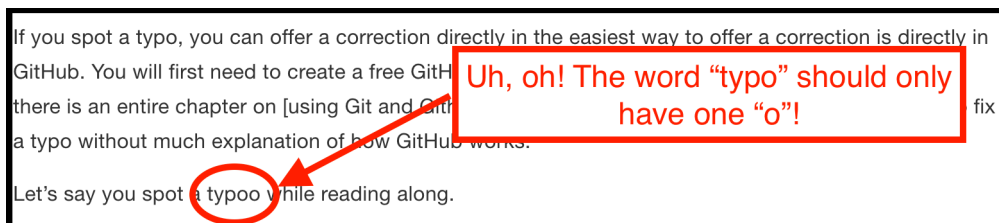
Over the years, we have learned so much from our students and colleagues, and we anticipate that there is much more we can learn from you – our readers. Therefore, we welcome and appreciate all constructive contributions to R4Epi!

Typos

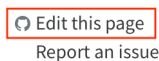
The easiest way for you to contribute is to help us clean up the little typos and grammatical errors that inevitably sneak into the text.

If you spot a typo, you can offer a correction directly in GitHub. You will first need to create a free GitHub account: [sign-up at github.com](https://github.com). Later in the book, we will cover using GitHub in greater depth (See Chapter 2). Here, we're just going to walk you through how to fix a typo without much explanation of how GitHub works.

Let's say you spot a typo while reading along.



Next, click the edit button in the toolbar as shown in the screenshot below.



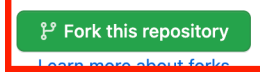
Left off here. - I need to change the screenshots and the description of the process because it changes a little with Quarto. Open Safari to update the screenshots.

The first time you click the icon, you will be taken to the R4Epi repository on GitHub and asked to fork it. For our purposes, you can think of a GitHub repository as being similar to a shared folder on Dropbox or Google Drive.



You need to fork this repository to propose changes.

Sorry, you're not able to edit this repository directly – you need to fork it and propose your changes from there instead.



**Fork the
Repository**

“Forking the repository” basically just means “make a copy of the repository” on your GitHub account. In other words, copy all of the files that make up the R4Epi textbook to your GitHub account. Then, you can fix the typos you found in your *copy* of the files that make up the book instead of directly editing the *actual* files that make up the book. This is a safeguard to prevent people from accidentally making changes that shouldn’t be made.

Note

Forking the R4Epi repository does not cost any money or add any files to your computer.

After you fork the repository, you will see a text editor on your screen.

You're making changes in a project you don't have write access to. We've created a fork of this project for you to commit your proposed changes to. Submitting a change will create a new branch in your fork, so you can send a pull request.

r4epi / index.Rmd in brad-cannell:master

<> Edit file Preview changes

Spaces 2

```
1 ---
2 title: "R for Epidemiology"
3 author: "Brad Cannell"
4 date: "`r Sys.Date()`"
5 description: "This is the textbook for Brad Cannell's Introduction to R Programming for Epidemiologic Research course."
6 cover-image: "r4epi_icon.png"
7 site: bookdown::bookdown_site
8 output: bookdown::gitbook
9 documentclass: book
10 bibliography: [book.bib, packages.bib]
11 link-citations: yes
12 csl: ama.csl
13 github-repo: "brad-cannell/r4epi"
14 url: 'https://brad-cannell.github.io/r4epi/'
15 ---
16
17 ```{r include=FALSE}
18 # automatically create a bib database for R packages
19 knitr::write_bib(c(
20   .packages(), 'bookdown', 'knitr', 'rmarkdown'
21 ), 'packages.bib')
22
23 # To fix error that says: "Error: Input files not all in same directory, please supply explicit wd
24 # Execution halted"
25 options(bookdown.render.file_scope = FALSE)
26 ```
27
28 # Welcome {-}
29
30 <!-- R4Epi Hex Sticker -->
31 
```

The text editor will display the contents of the file used to make the chapter you were looking at when you clicked the `edit` button. In this example, it was a file named `contributing.qmd`. The `.qmd` file extension means that the file is a Quarto file. We will learn more about Quarto files in Chapter 1, but for now just know that Quarto files can be used to create web pages and other documents that contain a mix of R code, text, and images. You may not understand the code you see in the text editor yet, but you will probably be able to skim through the text and find the typo you want to fix.

Next, scroll down through the text until you find the typo and fix it.


```
87  ### typos {-}
88
89  The easiest way for you to contribute is to help us clean up the little typos and grammatical errors that inevitably sneak into the text.
90
91  If you spot a typo, you can offer a correction directly in GitHub. You will first need to create a GitHub
92  account: sign-up at github.com(https://github.com/join). Later in the book, there is an entire chapter on using Git and Github. Here, we're
93  you through how to fix a typo without much explanation of how GitHub works.
94
95  Let's say you spot a typo while reading along.
96
97  ```{r echo=FALSE}
```

Now, the only thing left to do is propose your typo fix to the authors. To do so, simply scroll to the bottom of the same screen where you made the edits to the file. There, you will see a **Propose changes** form box you can fill out. In the first line, type a brief (i.e., 72 characters or less) summary of the change you made. There is also a box to add a more detailed description of what you did, but you shouldn't need to use it for a simple typo fix.



Propose changes

Fixed a typo in index.Rmd

Add an optional extended description...

Propose changes

Cancel

Next, click the **Propose changes** button. That will take you to another screen where you will be able to create a pull request. This screen is kind of busy, but try not to let it overwhelm you.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base repository: brad-cannell/r4epi base: master head repository: brad-cannell-test-user/r4epi compare: patch-1

✓ **Able to merge.** These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#) **Create pull request**

1 commit 1 file changed 1 contributor

Commits on Jan 3, 2022

Fixed a typo in index.Rmd
brad-cannell-test-user committed 2 minutes ago

Verified

Showing 1 changed file with 1 addition and 1 deletion.

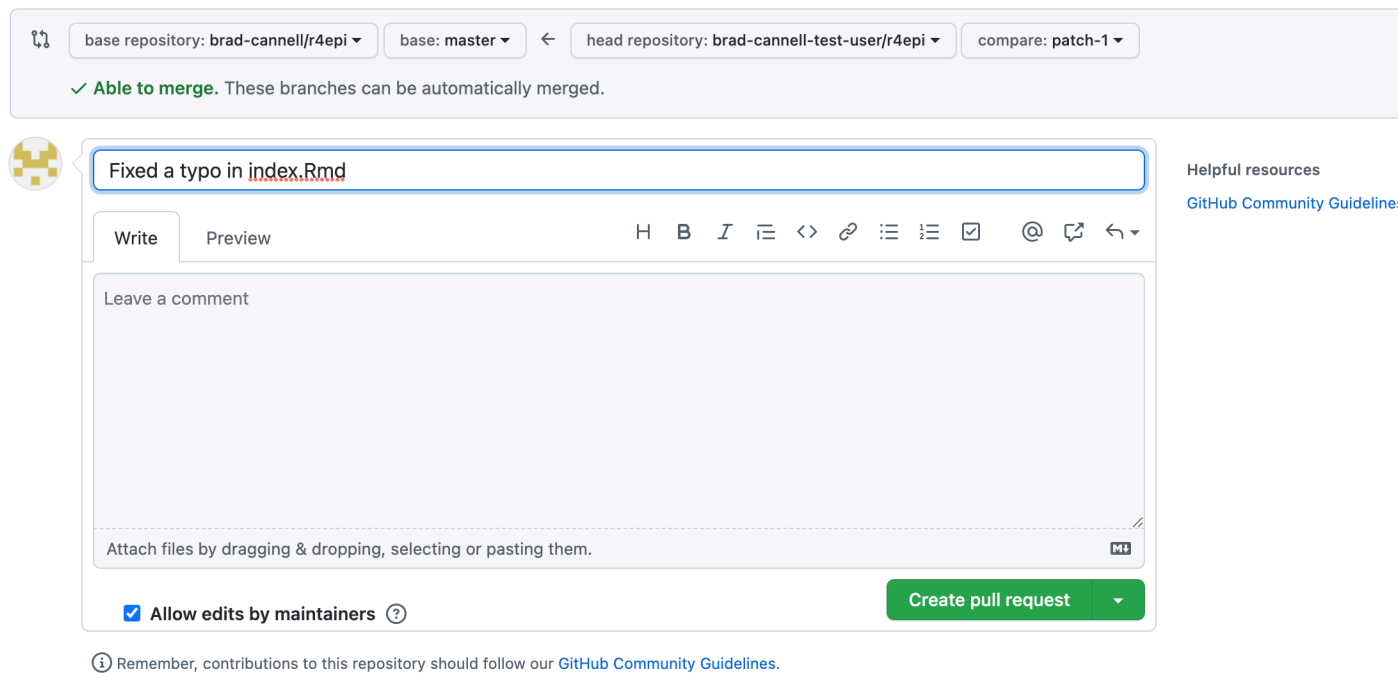
2 index.Rmd

```
@@ -90,7 +90,7 @@ The easiest way for you to contribute is to help us clean up the little typos an
90 90
91 91 If you spot a typo, you can offer a correction directly in the easiest way to offer a correction is directly in GitHub. You will fir
create a free GitHub account: [sign-up at github.com](https://github.com/join). Later in the book, there is an entire chapter on [us
Github]. Here, we're just going to walk you through how to fix a typo without much explanation of how GitHub works.
92 92
93 93 - Let's say you spot a typoo while reading along.
+ 93 + Let's say you spot a typo while reading along.
94 94
95 95 ```{r echo=FALSE}
96 96 knitr::include_graphics("img/typo_on_screen.png")
```

For now, we will focus on the three different sections of the screen that are highlighted with a red outline. We will start at the bottom and work our way up. The red box that is closest to the bottom of the screenshot shows us that the change we made was on line 93. We removed the word “typoo” (highlighted in red) and added the word “typo” (highlighted in green). The red box in the middle of the screenshot shows us the brief description we wrote for our proposed change – “Fixed a typo in index.Rmd”. Finally, the red box closest to the top of the screenshot is surrounding the **Create pull request** button. Let’s go ahead and click it now.

Open a pull request

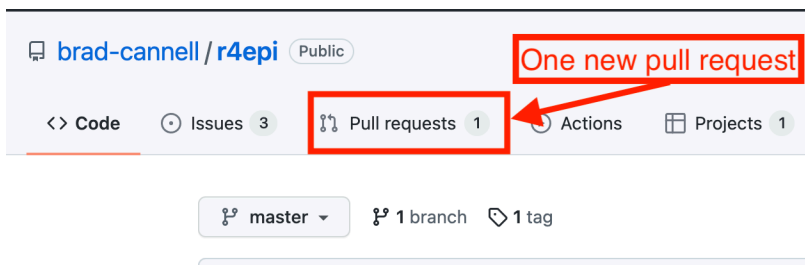
Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



The screenshot shows the GitHub interface for creating a pull request. At the top, there are dropdown menus for 'base repository: brad-cannell/r4epi', 'base: master', 'head repository: brad-cannell-test-user/r4epi', and 'compare: patch-1'. Below these, a green checkmark indicates 'Able to merge. These branches can be automatically merged.' The main form has a title field containing 'Fixed a typo in index.Rmd'. Below the title is a 'Write' tab and a 'Preview' tab. The 'Write' tab is active, showing a large text area for 'Leave a comment' and a smaller area for 'Attach files by dragging & dropping, selecting or pasting them.' At the bottom of the form, there is a checkbox labeled 'Allow edits by maintainers' which is checked, and a green button labeled 'Create pull request'. To the right of the form, there are links for 'Helpful resources' and 'GitHub Community Guidelines'.



After doing so, we will get one final chance to amend the description of our proposed changes. We don't want to change our description, so let's go ahead and click **Create pull request** one more time. Our job is done! It is now up to the authors to review the changes we've proposed and "pull" them into the file in their repository.

In case you are curious, here is what the process looks like on the authors' end. First, when we open the R4Epi repository page on GitHub, we will see that there is a new pull request.



When we open the pull request, we can see the proposed changes to the file.

Fixed a typo in index.Rmd #30

 Open **brad-cannell-te...** wants to merge 1 commit into **brad-cannell:master** from **brad-cannell-test-user:patch-1** 


 Conversation 0  Commits 1  Checks 0  Files changed 1


Changes from all commits ▾ File filter ▾ Conversations ▾ Jump to ▾ 


0 / 1 files viewed 


```
✓ 2 index.Rmd
@@ -90,7 +90,7 @@ The easiest way for you to contribute is to help us clean up the little typos an
90 90
91 91 If you spot a typo, you can offer a correction directly in the easiest way to offer a correction is directly in GitHub. You will first n
free GitHub account: [sign-up at github.com](https://github.com/join). Later in the book, there is an entire chapter on [using Git and G
we're just going to walk you through how to fix a typo without much explanation of how GitHub works.
92 92
93 - Let's say you spot a typoo while reading along.
93 + Let's say you spot a typo while reading along.
94 94
95 95 ```{r echo=FALSE}
96 96 knitr::include_graphics("img/typo_on_screen.png")
```


Then, all we have to do is click the **Merge pull request** button and the fixed file is “pulled in” to replace the file with the typo.





 **All checks have passed**
3 neutral and 2 successful checks [Hide all checks](#)


 **Header rules** Completed in 23s — No header rules processed [Details](#)

 **Pages changed** Completed in 23s — All files already uploaded [Details](#)

 **Redirect rules** Completed in 23s — No redirect rules processed [Details](#)

 **Mixed content** Successful in 23s — No mixed content detected [Details](#)

 **deploy/netlify** — Deploy Preview ready! [Details](#)

 **This branch has no conflicts with the base branch**
Merging can be performed automatically.

Merge pull request ▾ You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Left off here. Moving over content from index.Rmd.

12

Part I

Coding Tools and Best Practices

1 Quarto Files

Part II

Collaboration

2 Using git and GitHub

Part III

References

References

1. GitHub. About repositories. Published online December 2023.
2. R Development Core Team. *An Introduction to R.*; 2020.
3. RStudio. R markdown. Published online 2020.
4. RStudio. RStudio. Published online 2020.

A Glossary

Console. Coming soon.

Data frame. For our purposes, data frames are just R’s term for data set or data table. Data frames are made up of columns (variables) and rows (observations). In R, all columns of a data frame must have the same length.

Functions. Coming soon.

- **Arguments:** Arguments always go *inside* the parentheses of a function and give the function the information it needs to give us the result we want.
- **Pass:** In programming lingo, you *pass* a value to a function argument. For example, in the function call `seq(from = 2, to = 100, by = 2)` we could say that we passed a value of 2 to the `from` argument, we passed a value of 100 to the `to` argument, and we passed a value of 2 to the `by` argument.
- **Returns:** Instead of saying, “the `seq()` function *gives us* a sequence of numbers...” we could say, “the `seq()` function *returns* us a sequence of numbers...” In programming lingo, functions *return* one or more results.

Repository GitHub’s documentation says “a repository contains all of your code, your files, and each file’s revision history. You can discuss and manage your work within the repository.”¹ A repository can exist *locally* as a set of files on your computer. A repository can also exist *remotely* as a set of files on a server somewhere, for example, on GitHub.

Global environment. Coming soon.

Objects. Coming soon.

R. R is an integrated suite of software facilities for data manipulation, calculation and graphical display. R is very much a vehicle for newly developing methods of interactive data analysis. It has developed rapidly and has been extended by a large collection of packages. However, most programs written in R are essentially ephemeral, written for a single piece of data analysis.²

R markdown documents. R markdown documents are text files that can be used to clean and analyze your data interactively as well as share your final results in many different formats (e.g., Microsoft Word, PDF, and even websites). R markdown documents weave together R code, narrative text, and multimedia content together into a polished final product.³

RStudio. RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management. RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro (Debian/Ubuntu, Red Hat/CentOS, and SUSE Linux).⁴