

CBW's Bookdown Template Documentation

2024-12-02

Contents

1	CBW's Bookdown Documentation	5
2	Getting Started	7
2.1	Installation	7
3	Creating [RC] and Deploying Your New Workshop	9
3.1	Workshop Setup [RC]	9
3.2	Setting Up Team Access (Nia will fill this in)	14
4	Command Line & Git Clone	15
4.1	Introduction to the Command Line	15
4.2	Getting the Template on Your Local Computer - Git Clone! . . .	16
5	So What Do These Files Mean?	19
5.1	Bookdown Simple Explanation	19
5.2	Open in RStudio	19
5.3	Explaining RStudio	20
5.4	Build the book	21
5.5	File Setup Explanation	22
5.6	Push to GitHub via RStudio	22
6	Storing all formatting content	27
6.1	Chapters	27
6.2	A section	27
6.3	Cross-references	27
6.4	Chapters and sub-chapters	28
6.5	Captioned figures and tables	28
6.6	Parts	29
6.7	Footnotes	29
6.8	Citations	30
6.9	Blocks	30
6.10	Equations	30
6.11	Theorems and proofs	30
6.12	Callout blocks	30

6.13 Sharing your book	31
6.14 Publishing	31
6.15 404 pages	31
6.16 Metadata for sharing	31
7 How to Render Code	33
8 Brain Dump / FAQ	35
8.1 Danger Zones	35
8.2 Potential Errors & Bugs	35
8.3 FAQ	36
I DEVELOPERS GUIDE	37
9 Build Site	39
9.1 How to edit <code>_bookdown.yml</code>	39
9.2 How to edit <code>_output.yml</code> (RC)	39
9.3 Mandatory “ <code>index.Rmd</code> ” landing page	39
9.4 Build the book:	39
10 Git Instructions	41
10.1 How to Make a Git Repo (RC)	41
10.2 Updating GitHub via RStudio	44
11 How to Deploy Your Workshop Website	49

Chapter 1

CBW's Bookdown Documentation

Welcome to CBW's documentation for creating a workshop website using Bookdown. Bookdown is an R package that is used to build books, and in our case, the websites hosting CBW's workshops!

You only need to know markdown and whatever coding language you will be using to learn bookdown!

Note, this is the documentation to create a workshop using *bookdown*. If **Jupyter Book** suits you better, see here.

If you don't know which one to use, click [here](#) to learn more!

Chapter 2

Getting Started

Bookdown is an open-source R package that helps write books and articles. We will be building our bookdown-based workshop websites using this, specifically the gitbook template. (This is just the name of the specific template style, you will be working in a workshop template that CBW has prepared for you!)

If you're ready to start making a workshop website in bookdown, let's setup your device (PC, laptop)!

First, let's explain installations.

2.1 Installation

Since bookdown is an R package, you will need R. Plus, our ideal IDE (integrated development environment i.e. the platform we will be working in) is RStudio.

1. Download and install R (You need R 3.6.0+ installed for RStudio) here. Follow the instructions for your operating system (Linux/macOS/Windows). [Maybe we should have more installation instructions - a video?]
2. Download and install RStudio here. Scroll down to find downloads for non-macOS.
3. Installl the bookdown R package: Open RStudio and in the console (in the bottom left window of RStudio) run the following command:
`install.packages("bookdown")`

We're ready to start working with CBW's bookdown workshop template now!

Chapter 3

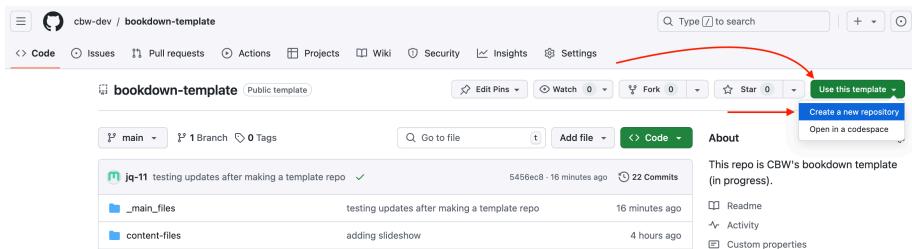
Creating [RC] and Deploying Your New Workshop

Certain aspects of the setup for workshops will be different depending on your role. Headers ending in “[RC]” are for Regional Coordinators. Headers without “[RC]” are assumed to be relevant to both RC and workshop teams.

3.1 Workshop Setup [RC]

Regional Coordinators will be tasked with creating and slightly editing each new repository for each new workshop.

1. First, let's go the bookdown template.
2. Click on the “Use this template” green button, which is to the left of the title of the repository “bookdown-template”. Then, press the dropdown option: “Create a new repository”, as seen below.



10CHAPTER 3. CREATING [RC] AND DEPLOYING YOUR NEW WORKSHOP

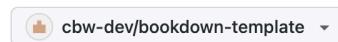
3. You will be brought to a “Create a new repository” page. Fill out the blanks as seen below. That is, change the owner to “bioinformatics.ca” [NOTE FOR TESTING PURPOSES: use cbw-dev], make it public, fill in the repository name and description according to CBW Guidelines. “Include all branches” does not need to be selected (??? - might need to look into this?).

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk ().*

Repository template



Start your repository with a template repository's contents.

Include all branches

Copy all branches from cbw-dev/bookdown-template and not just the default branch.

Owner *



Repository name *

workshop-name 

✓ workshop-name is available.

Great repository names are short and memorable. Need inspiration? How about [probable-octo-pancake](#) ?

Description (optional)

Clear description of the workshop 

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

 You are creating a public repository in your personal account.

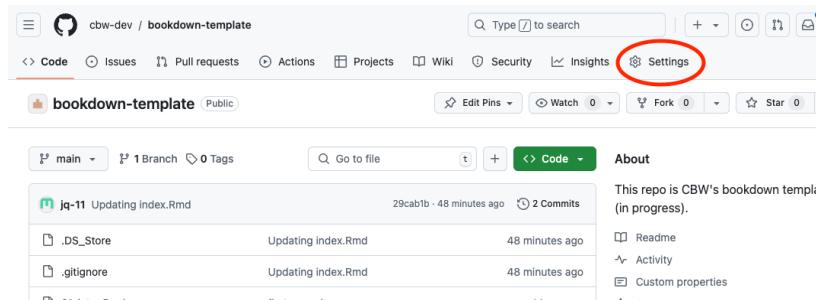
Create repository

This may take a couple seconds to generate. After it loads, you will be brought to a new repository for the new workshop!

Now, let’s turn this into a website - let’s deploy!

3.1.1 How to Deploy Your Workshop Website

1. In the top navigation bar, select settings.

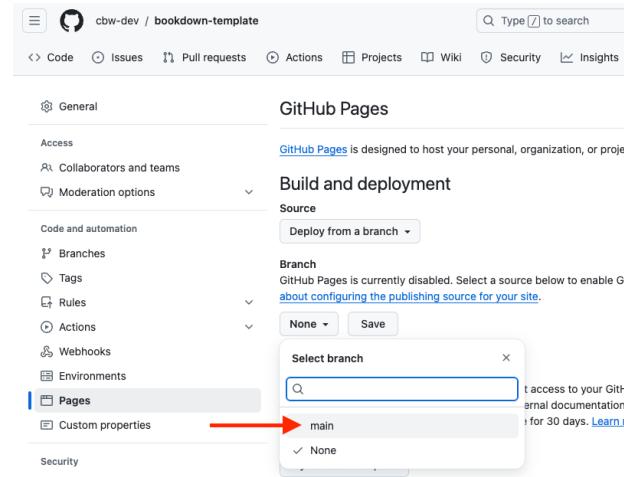


2. Then, go to the pages sidebar.

A screenshot of the 'General' tab in the GitHub repository settings for 'bookdown-template'. The sidebar on the left lists various settings categories: Access, Collaborators and teams, Moderation options, Code and automation, Branches, Tags, Rules, Actions, Webhooks, Environments, Pages (which is highlighted with a red arrow), and Custom properties. The main content area shows the 'General' section with fields for Repository name (set to 'bookdown-template') and a 'Template repository' checkbox. It also includes sections for 'Default branch' (set to 'main') and 'Social preview'.

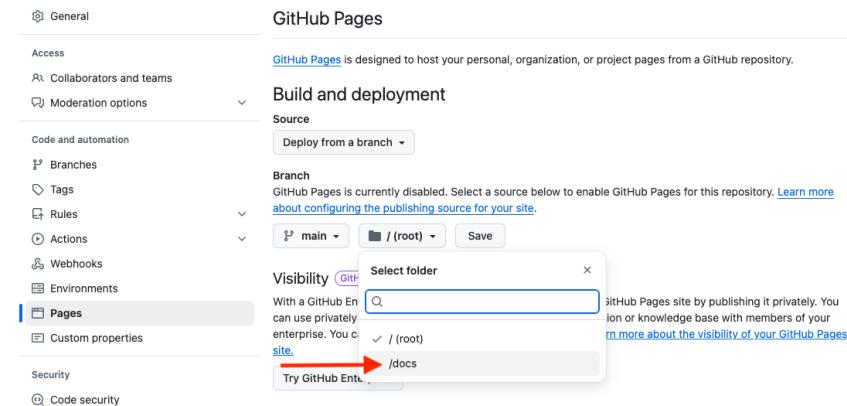
3. “Deploy from a branch” is already selected, which is what we want. We

12CHAPTER 3. CREATING [RC] AND DEPLOYING YOUR NEW WORKSHOP



must change the branch from “none” to main.

4. Then, change the folder from `/ root` to `/docs`. Then press save.



Great! Now we’re waiting on the page to build and deploy, which should take less than a minute.

To see updates, go to the **Actions** page (found along the top navigation bar). This will help you understand how the deploy is working, and if it succeeded or failed.

The screenshot shows the GitHub Actions 'Actions' tab for the repository 'cbw-dev / bookdown-template'. The 'All workflows' section is selected. Three workflow runs are listed:

- pages build and deployment #3:** Status: Queued (now). Triggered by 'jq-11'. Last updated 38 minutes ago.
- pages build and deployment #2:** Status: Success (1 minute ago). Took 26s. Triggered by 'jq-11'. Last updated 38 minutes ago.
- pages build and deployment #1:** Status: Failed (4 minutes ago). Error: 'forgot to make the .nojekyll'. Triggered by 'jq-11'. Last updated 38 minutes ago.

You can click **pages build and deployment** for updates. It will give you errors (which may not be very clear) or the link of your deployed page!

The screenshot shows the details of a GitHub Actions workflow run. The run is identified as **pages build and deployment #3**, triggered via dynamic 38 minutes ago by 'jq-11'. The status is **Success** with a total duration of **29s** and one artifact produced.

The workflow graph shows the following sequence of jobs:

```

graph LR
    build[build] --> report1[report-build-status]
    report1 --> deploy[deploy]
    deploy --> report2[report-build-status]
    
```

The 'Artifacts' section lists the produced artifact:

Name	Size
github-pages	383 KB

14 CHAPTER 3. CREATING [RC] AND DEPLOYING YOUR NEW WORKSHOP

The screenshot shows a GitHub Actions build summary for a workflow named "pages build and deployment #1". The workflow triggered via dynamic 41 minutes ago and failed with a total duration of 37s. The artifacts section is empty. The workflow graph shows three steps: "build" (failed, 22s), "report-build-status" (success, 3s), and "deploy" (pending). Annotations show one error in the "build" step, which is a warning message from Jekyll about generating files in a directory that does not exist.

```
graph LR; build[build] -- "22s" --> report[report-build-status]; report -- "3s" --> deploy[deploy]
```

Annotations
1 error

build
Logging at level: debug GitHub Pages: github-pages v232 GitHub Pages: jekyll v3.10.0 Theme: jekyll-theme-primer Theme source: /usr/local/bundle/gems/jekyll-theme-primer-0.6.0 Requiring: jekyll.github-metadata Requiring: jekyll-seo-tag Requiring: jekyll-coffeescript Requiring: jekyll-commonmark-ghpage Requiring: jekyll-gist Requiring: jekyll-github-metadata Requiring: jekyll-paginate Requiring: jekyll-relative-links Requiring: jekyll-optimal-front-matter Requiring: jekyll-readme-index Requiring: jekyll-default-layout Requiring: jekyll-titles-from-headings GitHub Metadata Initializing... Source: /github/workspace/.docs Destination: /github/workspace/.docs/_site Incremental build: disabled. Enable with --incremental Generating... Generating: JekyllOptionalFrontMatter:Generator finished in 1.2012e-05 seconds. Generating: JekyllReadmeIndex:Generator finished in 6.061e-06 seconds. Generating: JekyllPage:Pagination finished in 2.875e-06 seconds. Generating: JekyllRelativeLinks:Generator finished in 2.2362e-05 seconds. Generating: JekyllDefaultLayout:Generator finished in 1.074e-05 seconds. Generating: JekyllTitlesFromHeadings:Generator finished in 6.602e-06 seconds. Rendering: assets/css/style.scss Pre-Render Hooks: assets/css/style.scss Rendering Markup: assets/css/style.scss github-pages 232 | Error: No such file or directory @ dir_chdir0 - /github/workspace/docs
Show less

A Very Specific Build and Deployment Warning

The screenshot shows three GitHub Actions status messages. The first two are from the "pages build and deployment" workflow, both by "github-pages bot". The first message is a success (1 minute ago, 28s). The second message is a warning (1 minute ago, 36s), which is the same warning from the previous screenshot: "No such file or directory @ dir_chdir0 - /github/workspace/docs". The third message is from the "removing test update" workflow, pushed by "jq-11" (main branch) (2 minutes ago, 1m 21s).

This is a very specific (and unlikely) warning. It occurs when 1 deploy hasn't finished, but another deploy began. THIS IS NOT A CONCERN. This is a warning message you do not have to worry about!

3.2 Setting Up Team Access (Nia will fill this in)

Chapter 4

Command Line & Git Clone

4.1 Introduction to the Command Line

This is for those who have no (or extremely little) experience with the command line.

Using the command line, you can use text commands to interact with your computer's operating system. For us, we will be using it to move around our folders and to git clone our workshop into our computer, so we can work on it using RStudio!

Note:

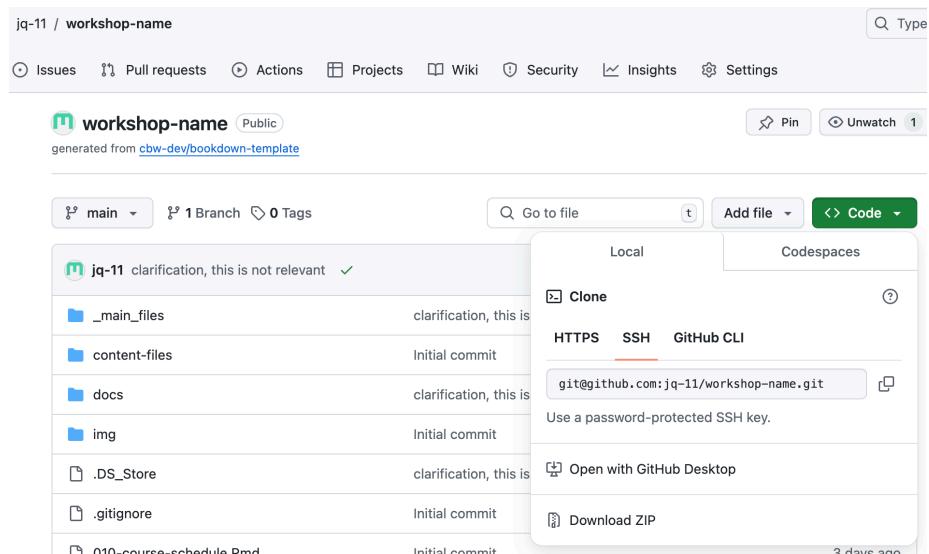
Do not be worried about using terminal, especially git commands in terminal! Once we are all setup, we will never have to touch the terminal and write these commands again!

[FILL IN LATER - JULIA CAN GUIDE NEHA]

- terminal (mac, unix based) vs command prompt (windows)
- pwd
- ls + enter
- cd
- . = current folder
- .. = previous folder
- ~ = home
- up and down arrows

4.2 Getting the Template on Your Local Computer - Git Clone!

1. Navigate to where in your local file system you want to have your workshop in (hint: cd + enter).
2. Find the ssh for your workshop repository: first, click the green button entitled < > Code and see the drop down options. Click the SSH tab, as seen below, and then copy the text below it. The text should be something like git@github.com:bioinformaticsdotca/WORKSHOP-NAME.git, as seen below.



Then put this command into your command line, within the folder you want the workshop folder to be in.

For example, if my organization is named: jq-11 and my workshop name is “workshop-name”, I would enter this into my terminal/command prompt.

```
git clone *git@github.com:jq-11/workshop-name.git*
```

4. You should be ready to go! With your given permissions, you should be able to git push adn git pull fine!

NOTE: Consider having only one team member (or perhaps your RC) make git pushes or control pull requests. To avoid merge conflicts, designate 1 team member to control actual changes to your workshop repo. Other team members can fork or create branches, and create a pull request that the designated team member can check and overlook.

4.2. GETTING THE TEMPLATE ON YOUR LOCAL COMPUTER - GIT CLONE!17

But what do any of these files mean? Which ones do I edit? Which ones shouldn't I edit? How do I open this in RStudio? And how exactly is a page made from all these files??? It's time for you to go to the next page :D

Chapter 5

So What Do These Files Mean?

Ok now we have our workshop locally (on our computer), which is made up of all these files and folders?

Before we dive deep into what to do with these folders, let's explore how bookdown actually works and how to understand RStudio.

5.1 Bookdown Simple Explanation

Here is a general summary of how Bookdown creates html websites from .Rmd files.

Essentially, knitr renders and runs all the code, and the output are converted into markdown. After knitr, we essentially have a bunch of only markdown files.

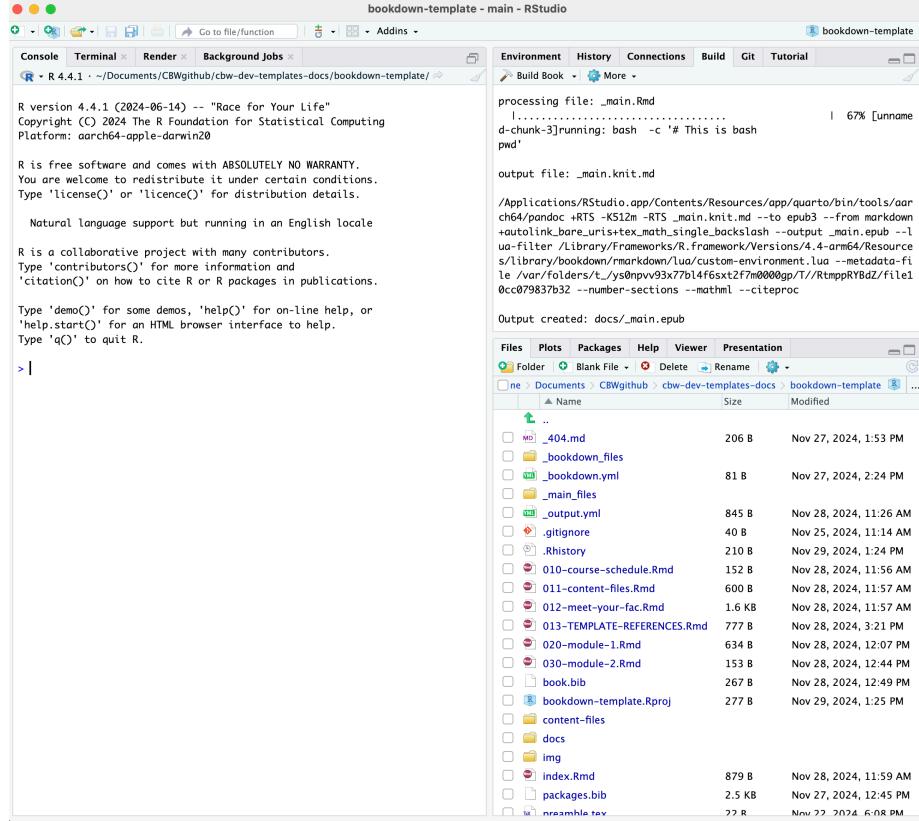
Pandoc translates this markdown into html, so that we get a website! It can be helpful to know when and how these packages work, to help debug later on!

Now, let's figure out RStudio. Skip to file setup if you already know how to use RStudio (and it's built in git control window).

5.2 Open in RStudio

Enter the folder you just git cloned using Findr/File Command, it should be titled “[workshop-name]”. Right click on [workshop-name].Rproj and press “Open in RStudio”. There is only one file with this file extension. The .Rproj file is what you will open each time you want to work on this workshop! You must explicitly **open the .Rproj** file to build properly!

A RStudio window should open up and look something like the image below.



5.3 Explaining RStudio

In the bottom left, have our console and other debug related windows (such as terminal!). Any code we run will appear in the console. We can access the terminal (just like editing in the Terminal app) under the “Terminal” tab.

In the bottom right, we have all of our files and subfolders. These files will be explained below. This window also contains helpful views, like “Viewer” and “Plots”. We will touch on these later.

Try opening `index.Rmd`: a new pane will open in the top left that shows the contents of `index.Rmd`. This is where we will be editing our files! Notice, the “Knit” button.

In your top right, we have a different window with more different views. The most relevant windows to us are the “Build” and the “Git” windows.

No “Git” Window?

Try closing (and maybe even restarting RStudio) and then reopening it. A “Git” tab should appear to the right of the “Build” tab and to the left of the “Tutorial” tab.

5.4 Build the book

Try pressing “Build Book” within the “Build” window. Your build window is going to fill up with text, and soon, a website is going to pop-up as your new window. This is the website you will be editing to create your workshop!

By building the book, all of these files were compiled and converted to .html files, that create a website. Each time we make local changes to our files and we want them to appear in our website, we need to rebuild the book. Note that each time we build our book, the files we edited will be saved first (we don’t have to save before building!).

5.4.1 Other Ways to Build Your Book

1. Build the book from the R console:

```
bookdown::render_book()
```

2. Press the keyboard buttons: cmd OR ctrl + shift + B

5.4.2 Knit Your Book

Building can take a long time. If you are editing just one file, you can press the “Knit” button that is at the top of the window with your file. This will run the code in the page, and show you what that page would look like in the website (as well as saving that file).

Note: Other pages in your website will not update.

A quicker way to knit is using the keyboard controls

```
cmd OR ctrl + shift + K
```

5.4.3 Preview Your Book

If you want live updates to your changes, you can preview the page as you edit the book when you save individual .Rmd files. You can start the server in a work session by using the RStudio add-in “Preview book”, or from the R console (in the bottom left window):

```
bookdown::serve_book()
```

But which files do we edit? Well alas, it’s time to discuss the file setup.

5.5 File Setup Explanation

[INSERT TREE ASCII] - asked IT whether I can install homebrew

Look at all these .Rmd files! Recall that .Rmd -> .md -> .html files (from the diagram earlier). Hence, most of the files we really need to edit or create more of are .Rmd files.

However, 2 other important files are the `_output.yml` files and the `_bookdown.yml` files. They help tell bookdown what we want and what to do, especially when making our website. Everything you need to edit here is made clear in the template, so do not worry too much about getting the ride code so that your project builds!

The only .Rmd that must have some configuration details is the landing page: `index.Rmd`. This is what fills up the beginning of the `index.Rmd` file, before the `---`, which tells bookdown to stop looking for configuration information.

Let's move on to discussing the breakdown of the .Rmd files. Each new page is defined by a new header, which starts with `#`, each subheaders have increasingly more `#` symbols (`##`, `###`, and more all create smaller subheaders). Try to have only 1 single `#` as a header on one page! (You may get warnings otherwise).

But then how does bookdown know which page goes next on our sidebar? The order of the sidebar is completely dependent on the alphabetical order of the files (see your bottom right!). Our template has numbering first, to help ease our understanding.

5.6 Push to GitHub via RStudio

Now, we know what our files mean and how to edit them. How do we get this onto GitHub? We can write git commands into our Terminal/Command Prompt, or alternatively (and more easily), RStudio has a built-in git interface.

Now, we will be able to see a Git window in the top right. Click “Git” to open this window.

Let's say we only edited `index.Rmd`, now we see the newly edited files. Changed files that need to be updated on GitHub will show up in this window, like how `index.Rmd` is seen above. (Do not worry too much about `.DS_Store` and `.gitignore` do.) Let's try to push this change to GitHub.

13. Select all the edited files.

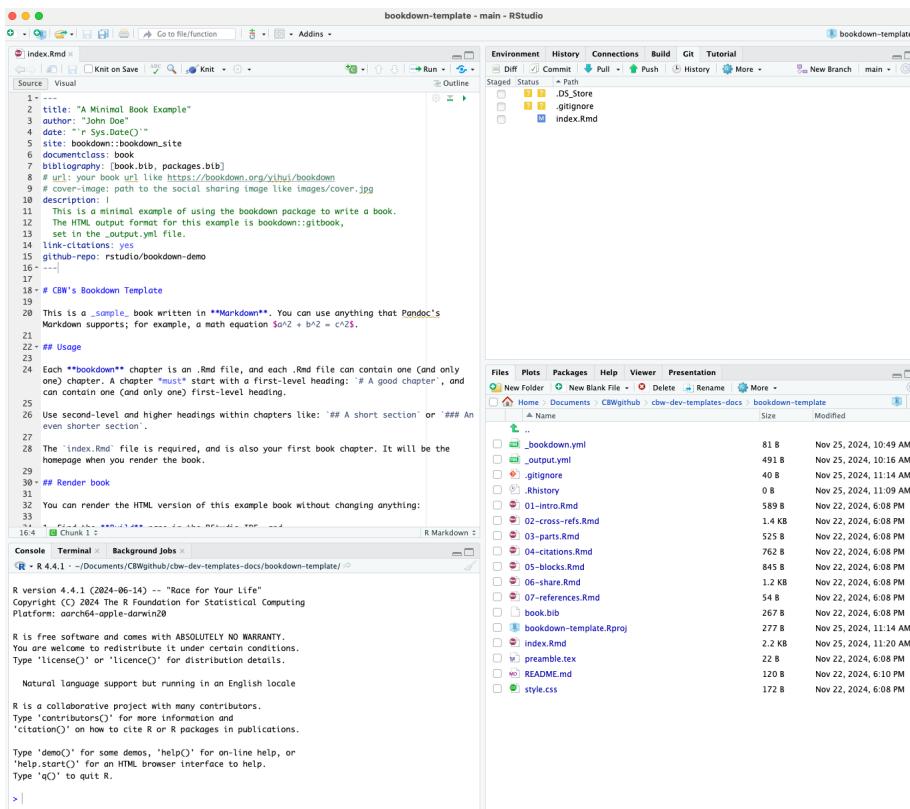
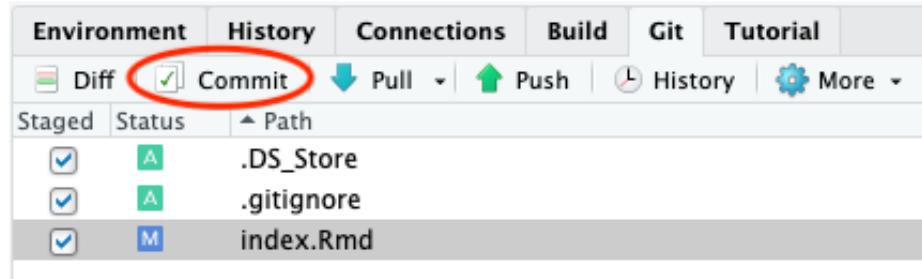
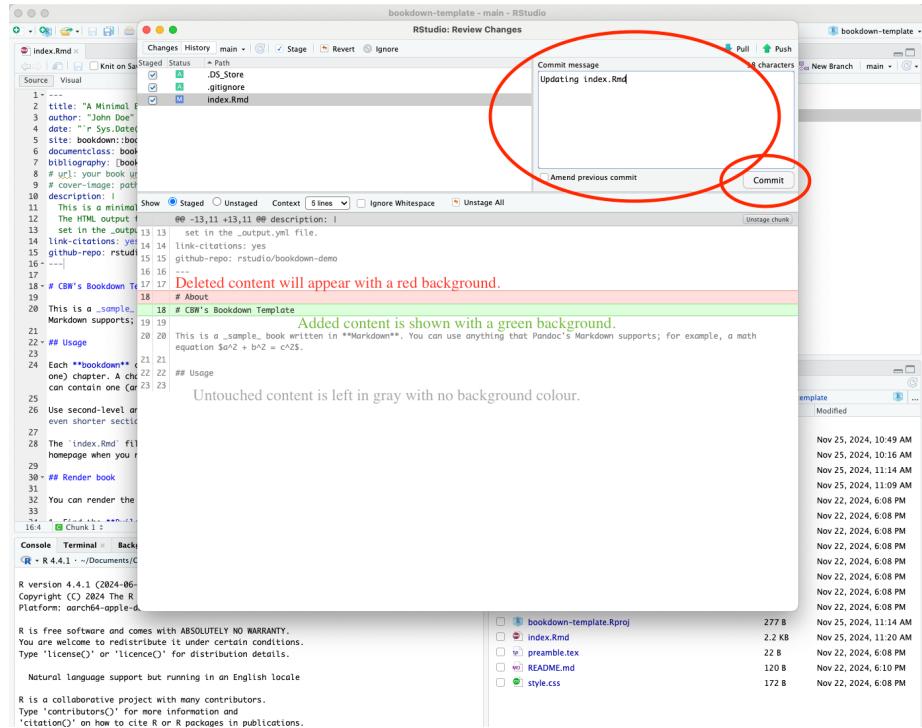


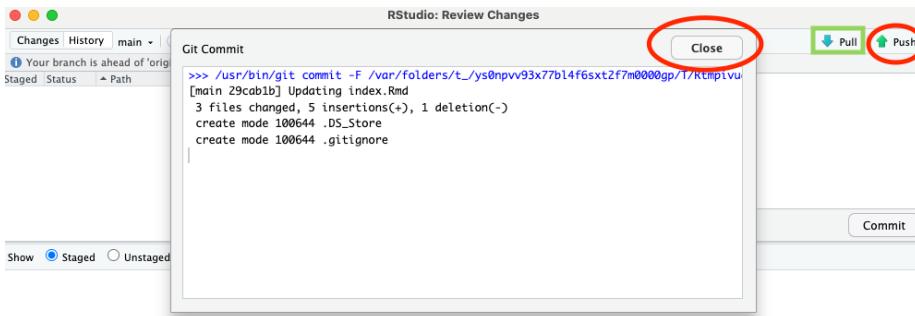
Figure 5.1: RStudio with Git window open



14. Then, click the Commit button, which appears above your selected items.
A window pane will appear (shown below).



15. Add a commit message in the corresponding box, and then press commit below it.
16. A new window will show up, detailing your updates. Close this window and then press Push to push your updates to GitHub.



Now, we're done! We should see the updates on GitHub now. Also note, if we ever want to pull updates from GitHub, there is also a **Pull** button in the Git window within RStudio!

Git pushing puts your edits onto GitHub, *git pulling* takes the edits made on GitHub, and brings them to your local computer. For example, if one of your workshop team members made an edit, you want to have that edit on your computer before you start editing! It's a good idea to do this before you start editing, in case somehow your edits conflicts with their edits.

Git pushing will automatically update the website, you can see the updates and progress in the actions window we saw previously. (Check out your website on the web once it's done deploying!)

Chapter 6

Storing all formatting content

- How to edit index.Rmd

6.1 Chapters

All chapters start with a first-level heading followed by your chapter title, like the line above. There should be only one first-level heading (#) per .Rmd file.

6.2 A section

All chapter sections start with a second-level (##) or higher heading followed by your section title, like the sections above and below here. You can have as many as you want within a chapter.

An unnumbered section

Chapters and sections are numbered by default. To un-number a heading, add a {.unnumbered} or the shorter {-} at the end of the heading, like in this section.

6.3 Cross-references

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

6.4 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 6.3.
 - If you prefer text as the link instead of a numbered reference use: any text you want can go here.

6.5 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(tab:chunk-label)`, respectively.

See Figure 6.1.

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

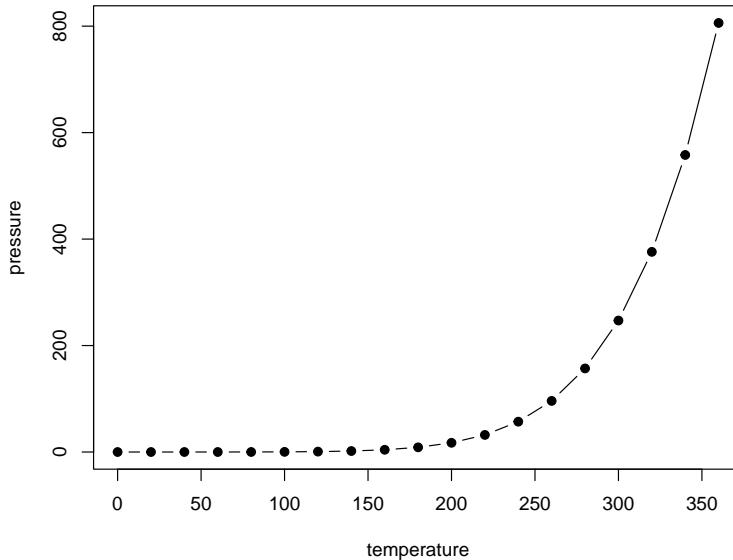


Figure 6.1: Here is a nice figure!

Table 6.1: Here is a nice table!

temperature	pressure
0	0.0002
20	0.0012
40	0.0060
60	0.0300
80	0.0900
100	0.2700
120	0.7500
140	1.8500
160	4.2000
180	8.8000

Don't miss Table 6.1.

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

6.6 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.

6.7 Footnotes

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

¹This is a footnote.

6.8 Citations

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

For example, we are using the **bookdown** package [Xie, 2024] (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>

6.9 Blocks

6.10 Equations

Here is an equation.

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (6.1)$$

You may refer to using `\@ref(eq:binom)`, like see Equation (6.1).

6.11 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref(thm:tri)`, for example, check out this smart theorem 6.1.

Theorem 6.1. *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>.

6.12 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>

6.13 Sharing your book

6.14 Publishing

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

6.15 404 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.

6.16 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
```


Chapter 7

How to Render Code

Chapter 8

Brain Dump / FAQ

8.1 Danger Zones

- renaming your file with this project, doesn't change the .Rprog file!
- GIT TIPS PAGE ???
- BEST PRACTICES PAGE
- MAKE SURE TO EXPLAIN WHAT HAPPENS WHEN MERGE CONFLICTS APPEAR

Ex. Forgot to pull before editing? - Your git window will say (at the top of the window) your branch is ahead of the main branch - You will probably have to deal with a merge conflict - [this is pretty in-depth]

- update favicon.ico image in template with one with higher quality

8.2 Potential Errors & Bugs

- If you get a bug where your website builds into a website that looks like it is made using very simple html, you may have to change your permissions. If you see “**Permission denied**” in your warning messages, trying running this command `chmod -R u+w docs` in terminal, in the folder containing your docs folder [CLARIFY ?]
- “**Could not produce X output**” - by default, bookdown builds all possible formats: gitbook, pdf, epub versions. We only need the gitbook, so if you're having issues, change your build settings to only creating the gitbook [CLARIFY + INCLUDE IMAGES]
- A very common bug is “**missing X package**”, just install it using this command in your console (the bottom left window in RStudio)

```
dio)      install.packages("missing package name, include these
surrouding quotations")
```

8.3 FAQ

Part I

DEVELOPERS GUIDE

Chapter 9

Build Site

9.1 How to edit `_bookdown.yml`

- add a new line, `output_dir: "docs"` to `_bookdown.yml`
- build the site
- add a `.nojekyll` file into the produced docs folder

9.2 How to edit `_output.yml (RC)`

- after `before:` change your workshop name link
- after `edit:` put the link to the workshop repo, and end the link with `/%s`
- save

9.3 Mandatory “`index.Rmd`” landing page

Each **bookdown** chapter is an .Rmd file, and each .Rmd file can contain one (and only one) chapter. A chapter *must* start with a first-level heading: `# A good chapter`, and can contain one (and only one) first-level heading.

Use second-level and higher headings within chapters like: `## A short section` or `### An even shorter section`.

The `index.Rmd` file is required, and is also your first book chapter. It will be the homepage when you render the book.

9.4 Build the book:

- “Build” button in RStudio IDE /OR/ `bookdown::render_book()`

- Preview the book: - updates on saves in viewer window `bookdown::serve_book()`

Before building

9.4.1 Render book

You can render the HTML version of this example book without changing anything:

1. Find the **Build** pane in the RStudio IDE, and
2. Click on **Build Book**, then select your output format, or select “All formats” if you’d like to use multiple formats from the same book source files.

Or build the book from the R console:

```
bookdown::render_book()
```

To render this example to PDF as a `bookdown::pdf_book`, you’ll need to install XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): <https://yihui.org/tinytex/>.

9.4.2 Preview book

As you work, you may start a local server to live preview this HTML book. This preview will update as you edit the book when you save individual .Rmd files. You can start the server in a work session by using the RStudio add-in “Preview book”, or from the R console:

```
bookdown::serve_book()
```

Chapter 10

Git Instructions

10.1 How to Make a Git Repo (RC)

1. Go to <https://github.com/cbw-dev> (CHANGE?) and scroll to your repositories.
2. Click the green “New” button to the right of the repositories search bar.
3. Create the new repository. Give it a *name* and *description*. Select *Public* instead of private, as shown below.

Warning

MAKE NAMING CONVENTION

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk ().*

Repository template

No template ▾

Start your repository with a template repository's contents.

Owner *

 cbw-dev ▾

Repository name *

/ bookdown-template

 bookdown-template is available.

Great repository names are short and memorable. Need inspiration? How about `supreme-octo-eureka` ?

Description (optional)

This repo is CBW's bookdown template (in progress).

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses](#).

 You are creating a public repository in the cbw-dev organization.

Create repository

- Click the green *Create repository* button at the bottom.

Now, we already have a local project. Now we want it on GitHub, so everyone on your team can make changes to the workshop! Let's make the GitHub connection (i.e let's add our local code to GitHub!)

10.1.1 How to Make the Git Connection (Adding your Local Repo to GitHub)

After the previous step, you will be brought to this page. The only things that will differ are the name of the repo.

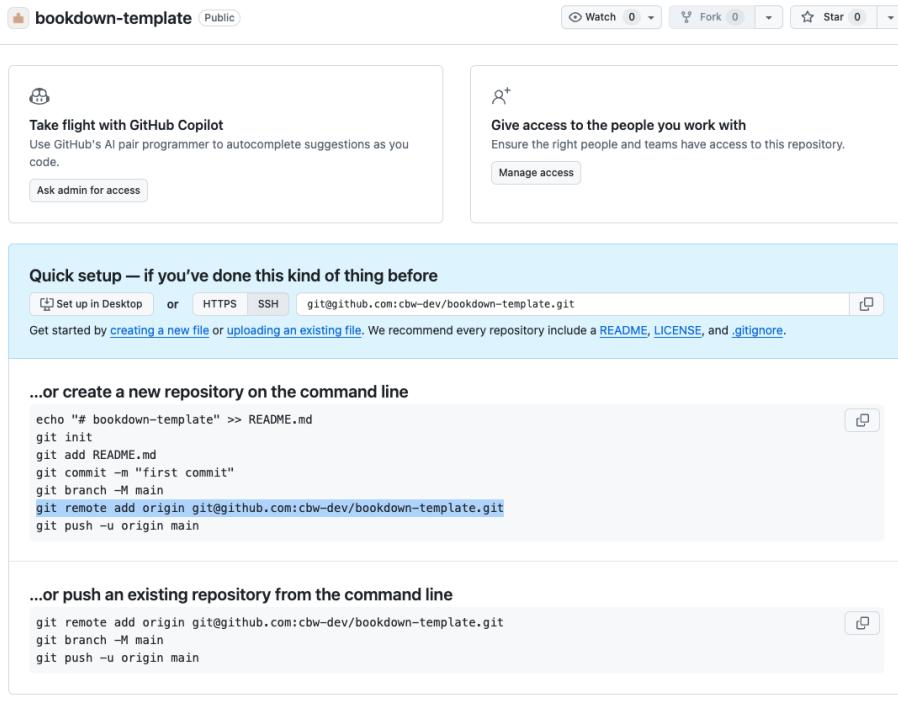


Figure 10.1: landing page after making a git repo

5. Open Terminal (Mac) or Command Prompt/Powershell (Windows).
6. Go to where we created the bookdown project.
7. Once inside the folder with the project. Let's make the git repo. First we initialize: `git init`. (Put this into terminal and press enter.)
8. Let's add all the files: `git add *`
9. Let's commit these files, with a descriptive message to help make it clear to others what we just did. For now, our message can be simple: `git commit -m "first commit"`. (Put this into terminal and press enter.)
10. Next, put this into terminal and press enter: `git branch -M main`.
11. **Important:** This step is why I highlighted that specific text above. Copy that command, and put it into terminal. Gen-

erally, it will look something like this: `git remote add origin git@github.com:cbw-dev/NAME-OF-YOUR-REPO.git`

12. Next, put `git push -u origin main` into terminal and press enter.

All the steps are shown below.

```
[jqliu@ML8034-JQIU ~ % cd Documents/CBWgithub/cbw-dev-templates-docs/bookdown-template
[jqliu@ML8034-JQIU bookdown-template % git init
[Initialized empty Git repository in /Users/jqliu/Documents/CBWgithub/cbw-dev-templates-docs/bookdown-template/.git/
[jqliu@ML8034-JQIU bookdown-template % git add *
[jqliu@ML8034-JQIU bookdown-template % git commit -m "first commit"
[main (root-commit) 44aae1d] first commit
[ 15 files changed, 266 insertions(+)
  create mode 100644 01-intro.Rmd
  create mode 100644 02-cross-refs.Rmd
  create mode 100644 03-parts.Rmd
  create mode 100644 04-citations.Rmd
  create mode 100644 05-blocks.Rmd
  create mode 100644 06-share.Rmd
  create mode 100644 07-references.Rmd
  create mode 100644 README.md
  create mode 100644 _bookdown.yml
  create mode 100644 _output.yml
  create mode 100644 book.bib
  create mode 100644 bookdown-template.Rproj
  create mode 100644 index.Rmd
  create mode 100644 preamble.tex
  create mode 100644 style.css
jqliu@ML8034-JQIU bookdown-template % git branch -M main
[jqliu@ML8034-JQIU bookdown-template % git remote add origin git@github.com:cbw-dev/bookdown-template.git
[jqliu@ML8034-JQIU bookdown-template % git push -u origin main
[Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 11 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (17/17), 5.74 KiB | 2.87 MiB/s, done.
Total 17 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:cbw-dev/bookdown-template.git
 * [new branch]  main -> main
branch 'main' set up to track 'origin/main'.
jqliu@ML8034-JQIU bookdown-template %
```

Figure 10.2: all the git steps typed out into terminal with results

10.2 Updating GitHub via RStudio

Now, close your RStudio session, and reopen it.

Now, we will be able to see a Git window in the top right. Click “Git” to open this window.

Let’s say we only edited `index.Rmd`, now we see the newly edited files. (Do not worry too much about `.DS_Store` and `.gitignore` do.) Let’s try to push this change to GitHub.

13. Select all the edited files.
14. Then, click the Commit button, which appears above your selected items. A window pane will appear (shown below).
15. Add a commit message in the corresponding box, and then press commit below it.
16. A new window will show up, detailing your updates. Close this window and then press Push to push your updates to GitHub.

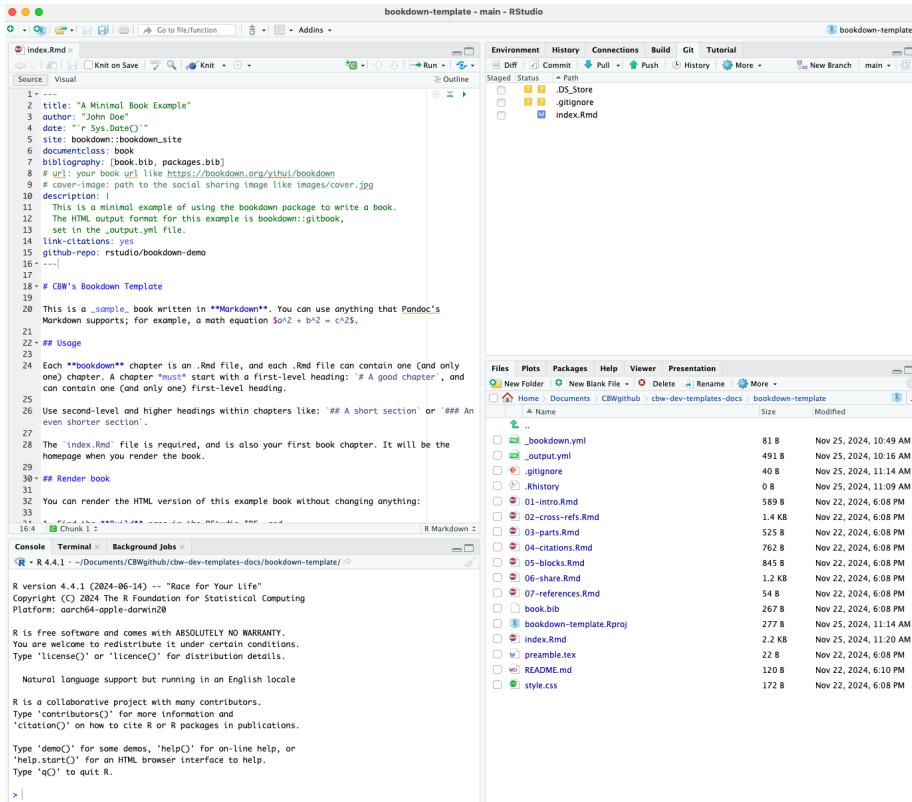


Figure 10.3: RStudio with Git window open

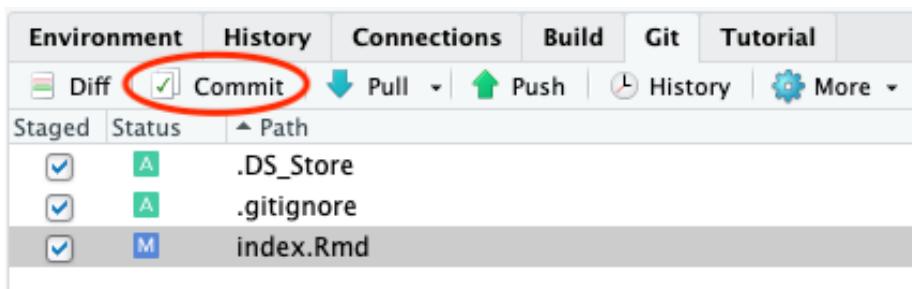


Figure 10.4: selected files in the git window and the commit button highlighted

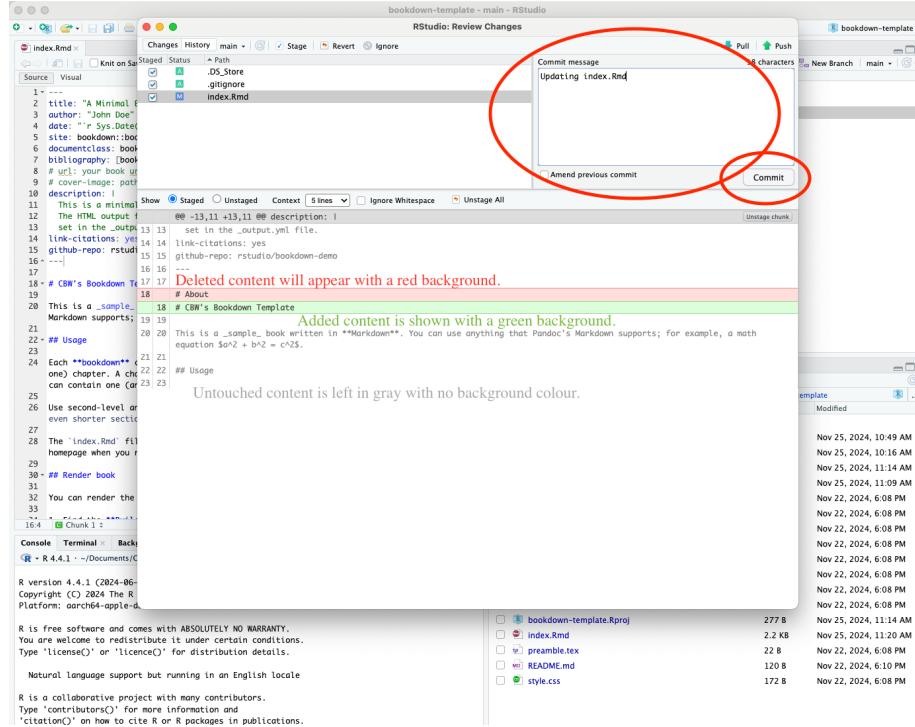


Figure 10.5: git commit window pane

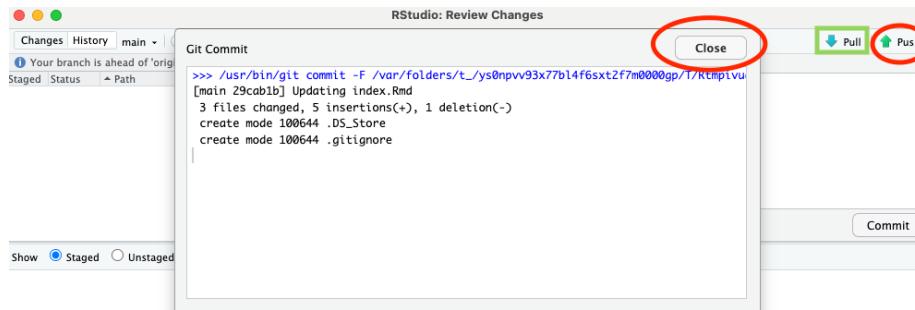


Figure 10.6: post git commit window

Now, we're done! We should see the updates on GitHub now. Also note, if we ever want to pull updates from GitHub, there is also a **Pull** button in the Git window within RStudio!

Chapter 11

How to Deploy Your Workshop Website

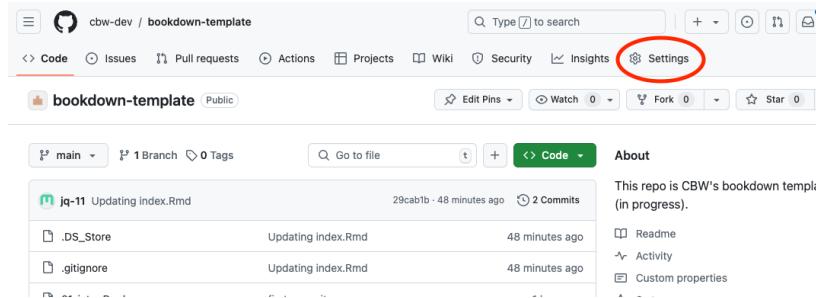
Let's recap.

We've made a bookdown project that builds into a website. We've reconfigured the output to go to a folder called "docs" (output_dir: "docs"). We've pushed our content onto github, and also made a ".nojekyll" file, which we placed into docs.

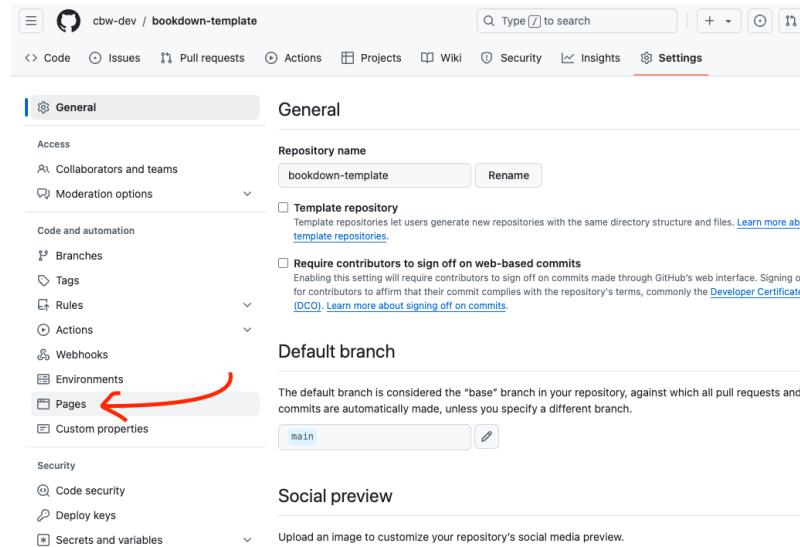
Now in our ./docs folder, we have a bunch of html files that make up our website. We want GitHub to look at these files in the docs folder and host the website for us!

We deploy our website using GitHub pages. GitHub pages uses jekyll, so the .nojekyll file tells it to no longer rely on jekyll. Now, all we need to do is tell GitHub pages to deploy (create/update the website) from our docs folder.

1. Go to your repo on GitHub.

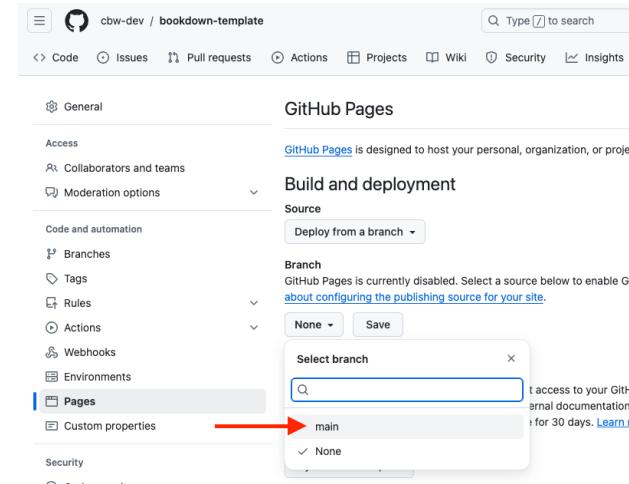


2. In the top navigation bar, select settings.



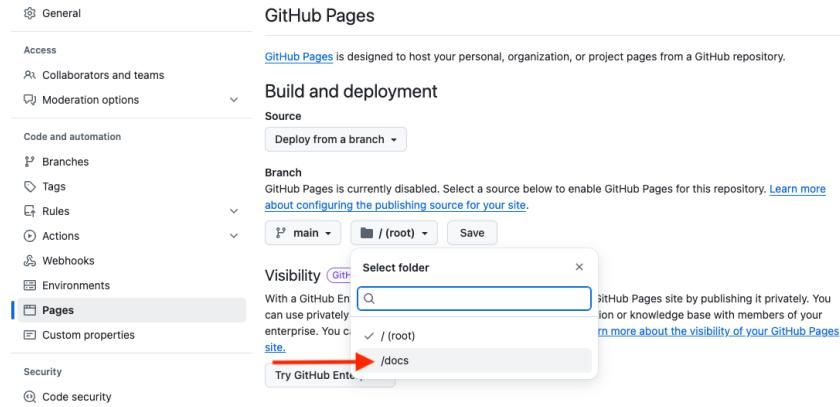
3. Then, go to the pages sidebar.

4. “Deploy from a branch” is already selected, which is what we want. We



must change the branch from “none” to main.

5. Then, change the folder from `/root` to `/docs`. Then press save.



Great! Now we're waiting on the page to build and deploy, which should take less than a minute.

To see updates, go to the **Actions** page (found along the top navigation bar). This will help you understand how the deploy is working, and if it succeeded or failed.

The screenshot shows the GitHub Actions page for the repository 'cbw-dev / bookdown-template'. The top navigation bar includes Code, Issues, Pull requests, Actions (which is selected), Projects, Wiki, Security, Insights, and Settings. The main content area is titled 'All workflows' and shows 'Showing runs from all workflows'. It lists three workflow runs:

Workflow	Event	Status	Branch	Actor
pages-build-deployment #3: by jq-11	now	Queued		...
pages-build-deployment #2: by jq-11	1 minute ago	26s		...
pages-build-deployment #1: by jq-11	4 minutes ago	37s		...

The first run is in progress ('Queued'). The second run was a 'Successful deploy' ('26s'). The third run failed ('Unsuccessful deploy') with the message '(forgot to make the nojekyll)'.

Figure 11.1: Image showing the different possibilities of deploy a github page

You can click **pages build and deployment** for updates. It will give you errors (which may not be very clear) or the link of your deployed page!

pages build and deployment #3

Triggered via dynamic 38 minutes ago Status Success Total duration 29s Artifacts 1

Re-run all jobs ...

Summary

Jobs

- build
- report-build-status
- deploy

Run details

Usage

pages-build-deployment
on: dynamic

```
graph LR; build[build] -- "5s" --> reportBuildStatus[report-build-status]; reportBuildStatus -- "3s" --> deploy[deploy];
```

Artifacts
Produced during runtime

Name	Size
github-pages	383 KB

<https://cbw-dev.github.io/bookdown-te...>

 pages build and deployment #1

 ...

Triggered via dynamic 41 minutes ago
jq-11 -> 29cab1b
Status
Failure
Total duration
37s
Artifacts
-

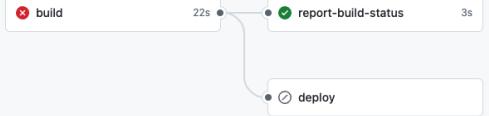
Jobs

-  **build**
-  **report-build-status**
-  **deploy**

Run details

 Usage

pages-build-deployment
on: dynamic



```

graph LR
    build[build] --> report[report-build-status]
    report --> deploy[deploy]
    
```

Annotations
1 error

 **build**

```

Logging at level: debug GitHub Pages: github-pages v232 GitHub Pages: jekyll v3.10.0 Theme: jekyll-theme-primer Theme source: /usr/local/bundle/gems/jekyll-theme-primer-0.6.0 Requiring: jekyll-github-metadata Requiring: jekyll-seo-tag Requiring: jekyll-coffeescript Requiring: jekyll-commonmark-ghpages Requiring: jekyll-gist Requiring: jekyll-github-metadata Requiring: jekyll-paginate Requiring: jekyll-relative-links Requiring: jekyll-optimal-front-matter Requiring: jekyll-readme-index Requiring: jekyll-default-layout Requiring: jekyll-titles-from-headings GitHub Metadata: Initializing... Source: /github/workspace/.docs Destination: /github/workspace/.docs/_site Incremental build: disabled. Enable with --incremental Generating... Generating: JekyllOptionalFrontMatter::Generator finished in 1.2012e-05 seconds. Generating: JekyllReadmeIndex::Generator finished in 6.061e-06 seconds. Generating: Jekyll::Paginator::Pagination finished in 2.875e-06 seconds. Generating: JekyllRelativeLinks::Generator finished in 2.2362e-05 seconds. Generating: JekyllDefaultLayout::Generator finished in 1.074e-05 seconds. Generating: JekyllTitlesFromHeadings::Generator finished in 6.602e-06 seconds. Rendering: assets/css/style.scss Pre-Render Hooks: assets/css/style.scss Rendering Markup: assets/css/style.scss github-pages 232 | Error: No such file or directory @ dir_chdir0 - /github/workspace/docs

```

Show less

Click around to explore more!

Bibliography

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*, 2024. URL <https://github.com/rstudio/bookdown>. R package version 0.40.