

CBW's Bookdown Template Documentation

2024-11-28

Contents

1	CBW's Bookdown Documentation	5
2	Getting Started	7
2.1	Installation	7
2.2	File Setup	7
3	Build Site	9
3.1	How to edit <code>_bookdown.yml</code>	9
3.2	How to edit <code>_output.yml</code> (RC)	9
3.3	Mandatory “index.Rmd” landing page	9
3.4	Build the book:	9
4	Git Instructions	11
4.1	How to Make a Git Repo (RC)	11
4.2	Updating GitHub via RStudio	14
4.3	How to Git Clone (ISC)	17
5	How to Push to GitHub	19
6	How to Deploy Your Workshop Website	21
7	Storing all formatting content	27
7.1	Chapters	27
7.2	A section	27
7.3	Cross-references	27
7.4	Chapters and sub-chapters	28
7.5	Captioned figures and tables	28
7.6	Parts	29
7.7	Footnotes	29
7.8	Citations	30
7.9	Blocks	30
7.10	Equations	30
7.11	Theorems and proofs	30
7.12	Callout blocks	30

7.13	Sharing your book	31
7.14	Publishing	31
7.15	404 pages	31
7.16	Metadata for sharing	31
8	How to Render Code	33
9	Brain Dump / FAQ	35
9.1	Danger Zones	35
9.2	Potential Errors & Bugs	35
9.3	FAQ	36

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 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

2.1 Installation

Install RStudio, get bookdown locally

2.2 File Setup

Explain file setup (tree diagram?)

Chapter 3

Build Site

3.1 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

3.2 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

3.3 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.

3.4 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

3.4.1 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/>.

3.4.2 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 4

Git Instructions

4.1 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

4. 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!)

4.1.1 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.

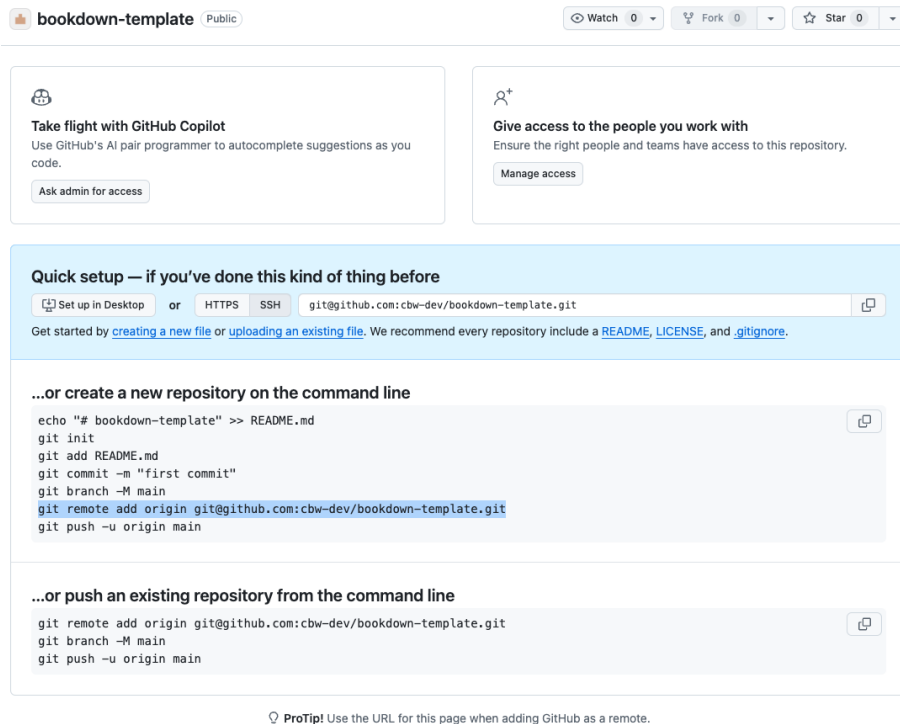


Figure 4.1: landing page after making a git repo

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.

```
jquiu@ML8034-JQIU ~ % cd Documents/CBWgithub/cbw-dev-templates-docs/bookdown-template
jquiu@ML8034-JQIU bookdown-template % git init
Initialized empty Git repository in /Users/jquiu/Documents/CBWgithub/cbw-dev-templates-docs/bookdown-template/.git/
jquiu@ML8034-JQIU bookdown-template % git add *
jquiu@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
jquiu@ML8034-JQIU bookdown-template % git branch -M main
jquiu@ML8034-JQIU bookdown-template % git remote add origin git@github.com:cbw-dev/bookdown-template.git
jquiu@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'.
jquiu@ML8034-JQIU bookdown-template %
```

Figure 4.2: all the git steps typed out into terminal with results

4.2 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.

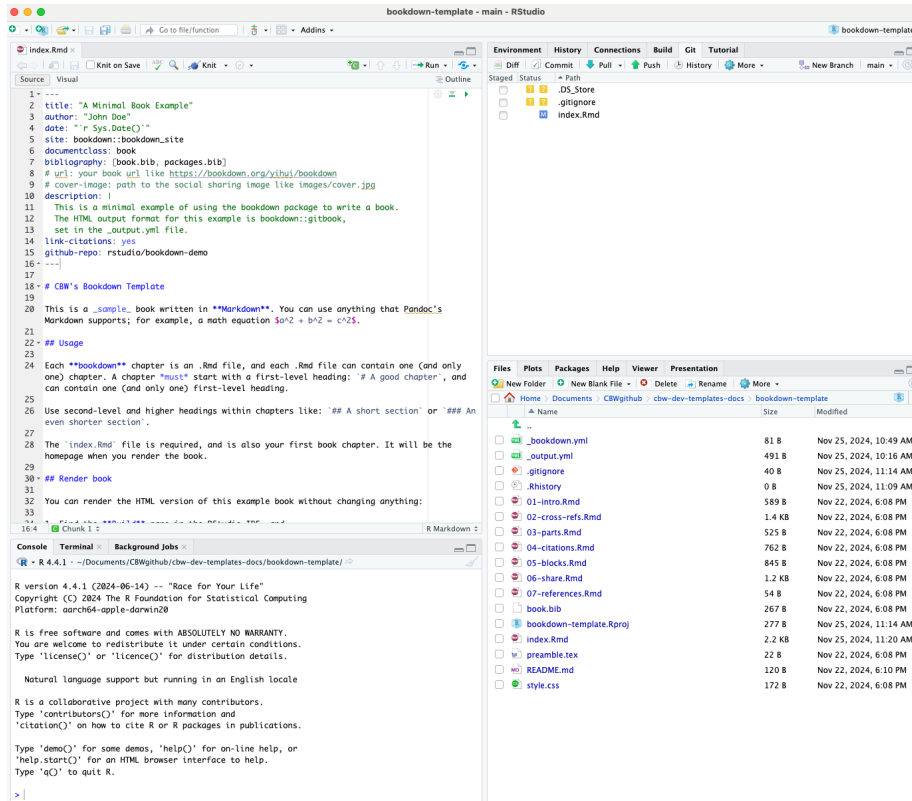


Figure 4.3: RStudio with Git window open

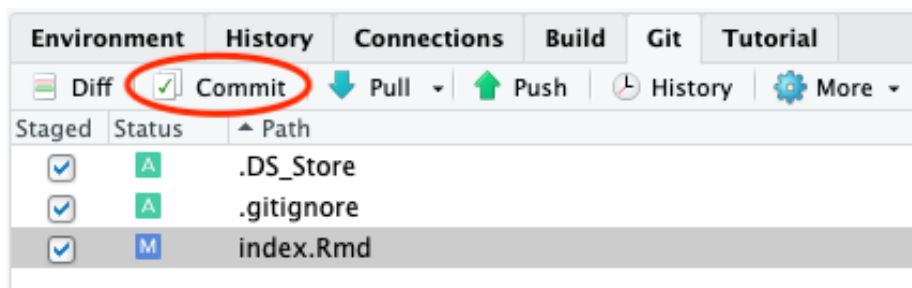


Figure 4.4: selected files in the git window and the commit button highlighted

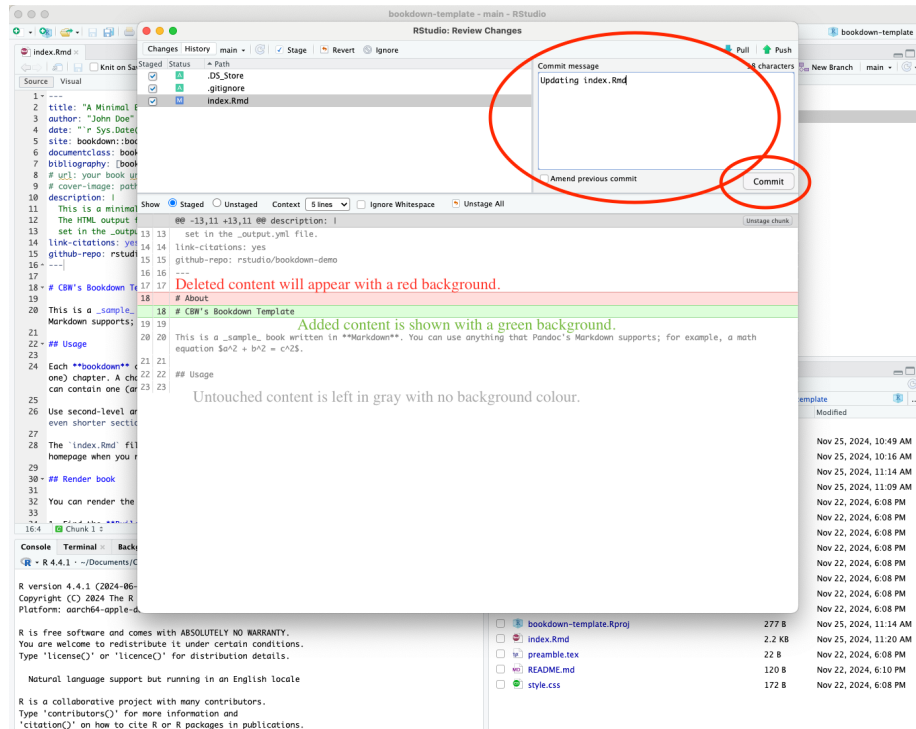


Figure 4.5: git commit window pane

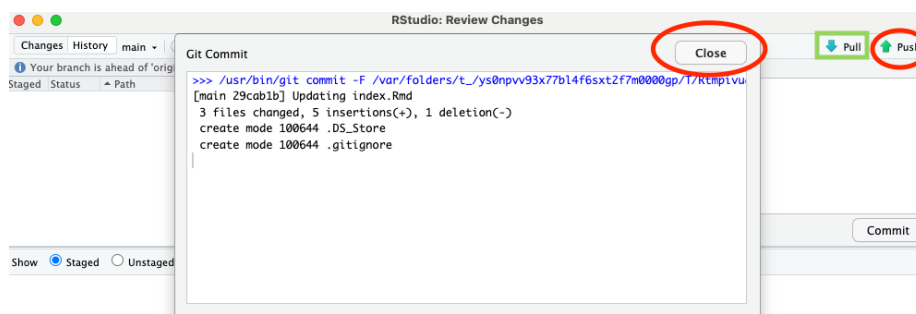


Figure 4.6: post git commit window

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!

4.3 How to Git Clone (ISC)

Chapter 5

How to Push to GitHub

Chapter 6

How to Deploy Your Workshop Website

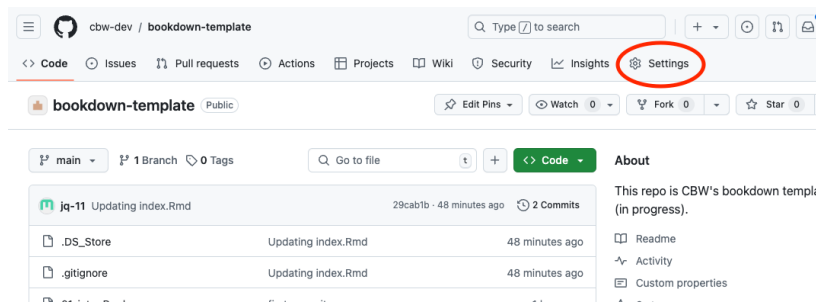
Let's recap.

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.

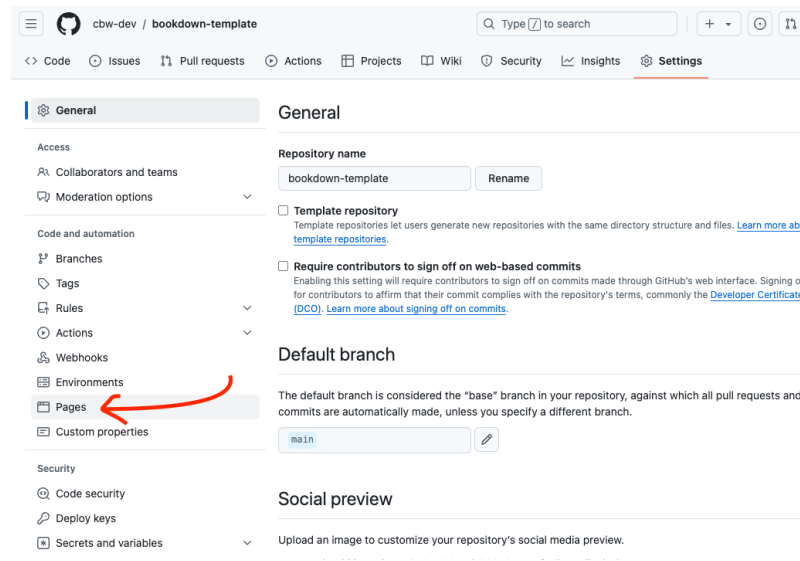
Now in our ./docs folder, we have a bunch of html files that make up our website. We want GitHub to look at these files in the docs folder and host the website for us!

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.

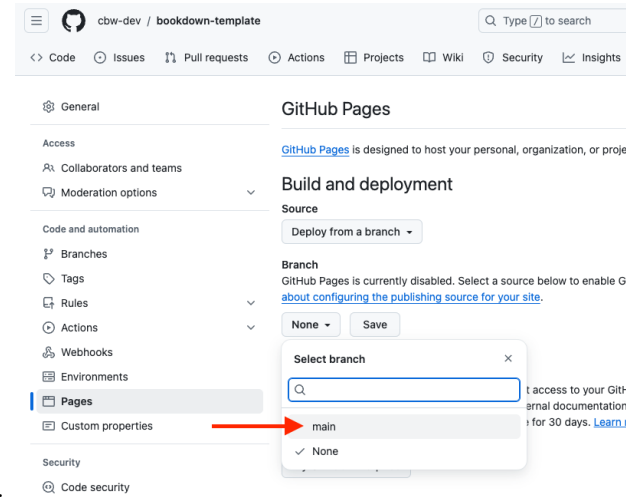


2. In the top navigation bar, select settings.



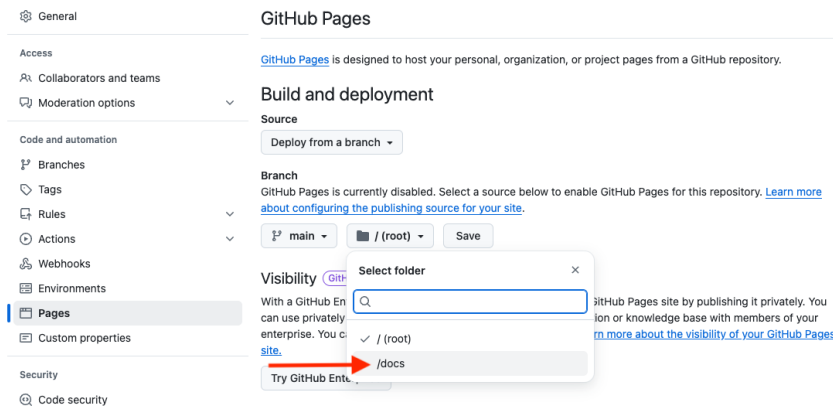
3. Then, go to the pages sidebar.

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.

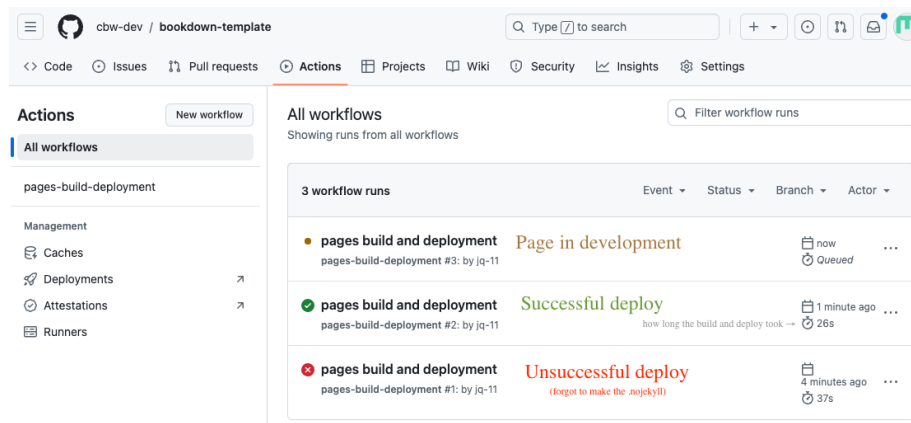


Figure 6.1: Image showing the different possibilities of deploy a github page

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

Re-run all jobs ...

Summary

Jobs

- ✓ build
- ✓ report-build-status
- ✓ deploy

Run details

Usage

Triggered via dynamic 38 minutes ago

jq-11 → 2973937

Status

Success

Total duration

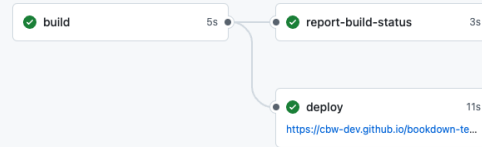
29s

Artifacts

1

pages-build-deployment

on: dynamic



Artifacts

Produced during runtime

Name	Size	
github-pages	383 KB	📄 🗑

pages build and deployment #1

Re-run jobs

Summary

Jobs

build

report-build-status

deploy

Run details

Usage

Triggered via dynamic 41 minutes ago

Status

Total duration

Artifacts

jq-11 -> 29cab1b

Failure

37s

-

pages-build-deployment

on: dynamic

build 22s

report-build-status 3s

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-optional-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::Paginate::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!

Chapter 7

Storing all formatting content

- How to edit index.Rmd

7.1 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.

7.2 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.

7.3 Cross-references

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

7.4 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 7.3.
 - If you prefer text as the link instead of a numbered reference use: any text you want can go here.

7.5 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 7.1.

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

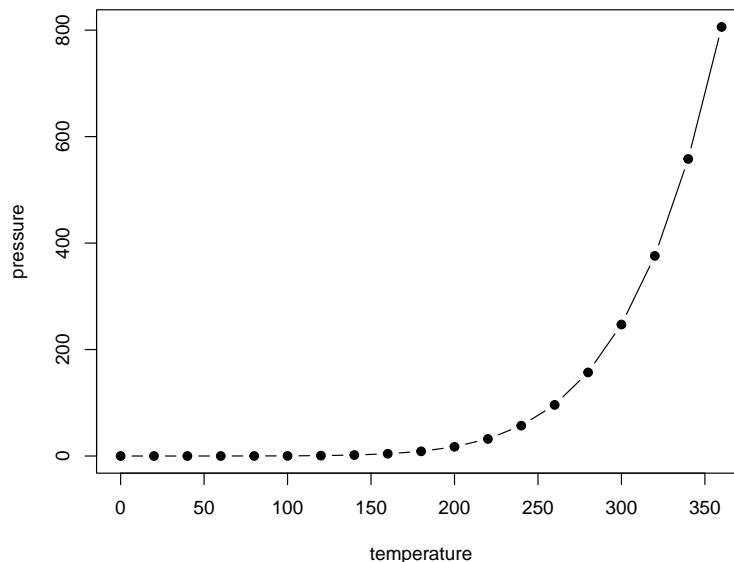


Figure 7.1: Here is a nice figure!

Table 7.1: 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 7.1.

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

7.6 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.

7.7 Footnotes

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

¹This is a footnote.

7.8 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>

7.9 Blocks

7.10 Equations

Here is an equation.

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (7.1)$$

You may refer to using `\@ref{eq:binom}`, like see Equation (7.1).

7.11 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref{thm:tri}`, for example, check out this smart theorem 7.1.

Theorem 7.1. *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>.

7.12 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>

7.13 Sharing your book

7.14 Publishing

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

7.15 404 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.

7.16 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 8

How to Render Code

Chapter 9

Brain Dump / FAQ

9.1 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

9.2 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]

9.3 FAQ

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.