Module 2 – lesson 06: A Book Format

In this lesson, we will be working with the bookdown demo repository. Go to this site: (<https://github.com/rstudio/bookdown-demo>). And download the repository as a ZIP file. Click on the green Clone or Download button and Download as a ZIP file into your local directory for this course.

C:\RepResearchCourse

Unzip the file into this folder. When you’re finished, you should have all of the files from the bookdown-demo repository in this directory.

C:\RepResearchCourse\bookdown-demo-master

There is ALREADY an .Rproj file – so this repository already comes with an RStudio project ready to go.

Start RStudio and Click Open Project and select bookdown-demo.Rproj in this directory

C:\RepResearchCourse\bookdown-demo-master

Note in the upper right window, there is now a TAB that says Build. This TAB is used to “Build” the final book. However, there is NO GIT TAB – this is because we need to re-establish a link using GIT back to your Github account. To do this you may want to refer to Jenny Bryan’s Happy Git and Github for user book – Chapter 18 on how to connect to Github when you have an Existing Project and you are connecting to Github last.

(<http://happygitwithr.com/existing-github-last.html> )

To turn GIT on in RStudio, click on Tools/Project options – click on Git/SVN. At the moment, the Version Control system says “none” – click on the arrow and choose Git. Another window will pop open asking if you want to initialize a new GIT repository for this project – click YES. You’ll then need to restart R-Studio.

After restarting, you now have the GIT TAB in the window at the upper right next to the Build TAB. But we’re not done – we still need to establish a connection between your local “repository” (the directory with all of the files and RStudio project for the bookdown-demo) and your Github cloud account.

Before we connect to the cloud, you need to create a new repository in your Github account. Log into your Github account and Click New Repository. To avoid confusion, go ahead and name your new repository the same as your RStudio project “bookdown\_demo”. However, DO NOT initialize it with a README. Click Create Repository. Your repository is created but there are NO FILES in it – yet.

Let’s connect using GIT. Open Git Bash and change directory to

C:\RepResearchCourse\bookdown-demo-master

Let’s stage – add all of the files

git add .

Next let’s commit these files

git commit –m “add files for bookdown demo”

BUT BEFORE we can PUSH these up to the cloud, we need to connect this directory to your cloud account. To do this – go back to the Github repository you just created and click the “copy to clipboard” button. This will copy the URL for the repository.

Now go back to GIT BASH and type in “git remote add origin” followed by the URL you just copied to your clipboard.

More information on git remote add origin is here:

(https://github.com/melindahiggins2000/bookdown\_demo.git)

The next step is to PUSH your content. However, the syntax is slightly different since this is our first time connecting the Github account to this repository. According to Jenny Bryan we need to “cement the tracking relationship between your GitHub repository and the local repo by pushing and setting the “upstream” remote”. To do this, type in the following GIT command.

git push -u origin master

This will take a minute or to two to run. When it finishes, go back and refresh your Github repository and you should now see all of the bookdown\_demo files in your cloud account.

Now that you have the basic files and have everything backed up to your Github account and synched up, let’s dive into all of these files and see how everything works.

Go back to RStudio. The .gitignore file “specifies intentionally untracked files that Git should ignore.” See the GIT documentation here (<https://git-scm.com/docs/gitignore> )

RStudio created a .Rhistory file (which is empty at the moment – that’s ok).

Click on the bookdown-demo.Rproj. This opens a window showing the various Project Settings for this RStudio project. This is the same window that pops up when you click on Tools/Project Options. The key elements to notice in this window are the settings in the Git/SVN tab which should show that YES you are using Version Control and the URL of your Github repository is now listed. Also see the settings under the Build Tools. Here the Project Build tools should show the Website and we’ll keep all of the other default settings for (Project Root) for the Site Directory, all formats for book output formats, and the box is checked for preview book after building and re-knit current preview when supporting files change.

There are 3 YAML files. YAML header information can be kept in a separate file instead of at the top of an R Markdown file, which is what has been done here.

.travis.yml – this YAML file provides some settings if you choose to set-up TRAVIS CI for “continuous integration” – learn more about TRAVIS CI here: (<https://travis-ci.org/>) - this is beyond the scope of this course, but may be worthwhile learning more if you plan to write computer programs and scripts for data analysis.

\_bookdown.yml – this YAML file contains information about the book, like the filename and chapter names.

\_output.yml – this file has various options for compiling the book in multiple formats like gitbook (which is an HTML format), PDF book and even an EPUB book format.

There are 2 “SHELL” script files (\_build.sh and \_deploy.sh) we will ignore for now.

There are 2 CSS (cascading style sheets) files (style.css and toc.css) that contain codes for formatting the resulting HTML format for the book. We will not change these files for now.

There is a DESCRIPTION and LICENSE file we will ignore for now. Feel free to view these files and learn more about them in the Bookdown book (<https://bookdown.org/yihui/bookdown/>) You may want to update the LICENSE file in the future if you plan on distributing your book online under a different LICENSE.

There is also a README.md file which you can edit later.

The file called preamble.text has some important formatting and setup information for compiling the book to PDF format using LaTeX. We can ignore this for now.

The remaining 7 files are all R Markdown RMD files. The introductory chapter is defined by index.Rmd. Go ahead and open this file.

At the beginning of this index.Rmd file there is another YAML header. Change the author to your name.

Notice the YAML keyword for bibliography – this line sets up the references used or created when this book is compiled or rendered. There are 2 BIB files listed book.bib which is a file in your repository and packages.bib which does not exist yet. Packages.bib will be created when the book is built. These BIB files are BibTeX formatted files – learn more here: (<http://www.bibtex.org/>)

After this introductory chapter there are 6 more - each named with a number and short description. Take a few minutes and open and look at each of these RMD files. You’ll notice that there is NO YAML at the top of these files. And each RMD file starts with a level 1 header indicated by 1 hastag #.

Let’s try building an HTML book. Go to the Build TAB at the top right. Click the down arrow for Build Book and choose bookdown::gitbook. You may be prompted to install a package as needed. Say yes.

If all goes well, your book will appear in the Viewer window. Let’s open a File Explorer window to see the new files.

You’ll notice that the file packages.bib was created. This file now has the BibTeX for all of the citations for each R package used to build this book.

There are also 2 new folders.

\_book – this directory has all of the HTML viewing files. If you go to this folder and double click on index.html, you will be able to see the book in a browser window.

\_bookdown\_files – this folder has a few more subfolders and an image that was produced by one of R code chunks that made a plot in the book.

If you have LaTeX installed, try making a PDF copy of the book – go to Build Book and click bookdown::pdf\_book

When this finishes compiling, there will be a new file in the \_book directory called bookdown-demo.pdf.

Let’s try one more change. Open the \_bookdown.yml YAML file. After the chapter\_name line, add a line with this option for the output directory

output\_dir: docs

Click Build Book to bookdown::gitbook to rebuild the book into an HTML format. This time instead of all of the HTML files being placed in the \_book directory, a new docs directory is created and all the HTML and supporting files are here. There is a reason we created the docs directory - I’ll show you why we want to save these HTML files into the docs directory in just a minute.

Go ahead and back up all of these changes and sync everything up to your Github repository.

Open Git Bash and change to the directory for your Github repository created for “bookdown-demo” – go to:

C:\RepResearchCourse\bookdown-demo-master

Once in that directory, type in the following 4 Git commands to check the status of your local files compared to your Github cloud repository; add or stage the modified files; commit your changes; and then push the changes to your Github cloud repository.

git status

git add .

git commit –m “compiled bookdown demo”

git push

Open your browser and go to your bookdown\_demo repository. Check to make sure your changes were pushed to your Github cloud repository. At the top of your repository at the far right there is a gear-shaped icon for the settings of your repository. Click on this icon.

Scroll about half way down the page to the section titled Github Pages. We’re going to set up a “webpage” for your book. Click on the button for Source – change it from “None” to “master branch/docs folder” and click Save. After you click save, the page will refresh. Scroll back down to the Github Pages section of the setting page, you’ll see the URL for your book’s website. It should look something like

https://USERNAME.github.io/bookdown\_demo/

https://melindahiggins2000.github.io/bookdown\_demo/

Click on this URL and you should see your book online! Congratulations you have now created a book and served it online. Plus, since we have everything connected using Git from your local drive to your Github cloud account, AND we have it setup to publish to your /docs directory - any time you make changes to your book and sync the files up using Git, your online book will be updated also!!

Take a few moments and try making changes to the different RMD files for each chapter by adding text, inserting some R code, adding a plot or an image and recompile the book, resync the files to Github and view the changes.

We’ve barely scratched the surface on creating a book using the bookdown package. I encourage you to learn more because a BOOK format is a wonderful way to organize multiple related documents together in one nice neat package even if your intention was not to write a “book”, but to simply organize a multi-component report.