An introduction to markdown basics

The A Team :)

2021-06-11

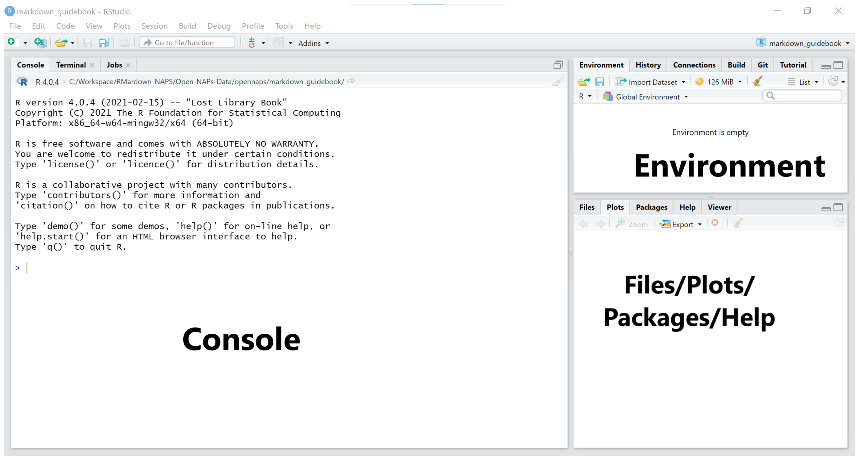
Contents

Chapter 1

Prerequisites

1. You have installed R (https://www.r-project.org/)
2. You have installed RStudio (https://www.rstudio.com/products/rstudio/download/)

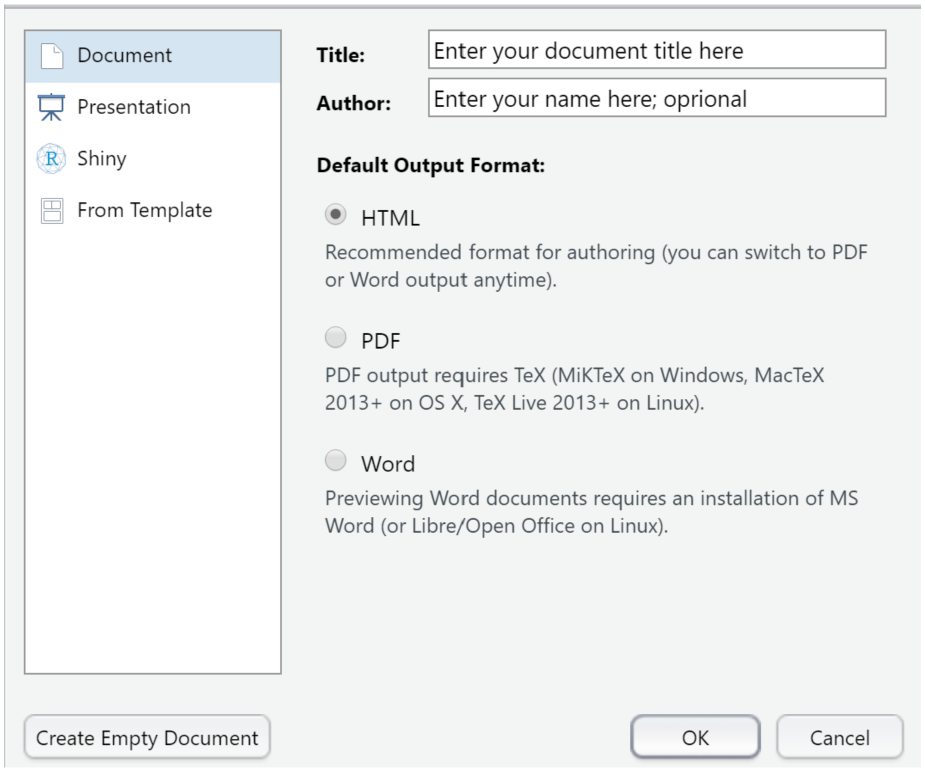
1.1 Opening a markdown file

1. Launch your r/studio app. Your rstudio workspace should look more or less similar to this: 

```
R version 4.0.4 (2021-02-15) -- "Lost Library Book"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

markdown_guidebook - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
markdown_guidebook
Console Terminal Jobs
R 4.0.4 · C:/Workspace/RMardown_NAPS/Open-NAPs-Data/opennaps/markdown_guidebook/
Environment History Connections Build Git Tutorial
Import Dataset
126 MiB
List
Global Environment
Environment is empty
Environment
Files Plots Packages Help Viewer
Zoom
Export
Files/Plots/
Packages/Help
Console

If your rstudio opens up with 4 windows instead of 3, skip next step.
2. From the main menu, go to File-»New File-»R Markdown. You shoukd have a

window like this: 

Title: Enter your document title here

Author: Enter your name here; oprional

Default Output Format:

[✓ HTML]
Recommended format for authoring (you can switch to PDF or Word output anytime).

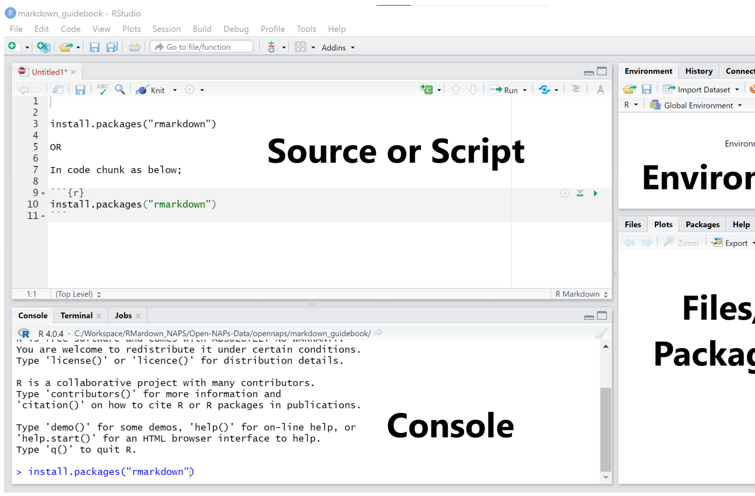
PDF
PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

Word
Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

3. Fill in the details for your document title and author. Leave the rest as default.

4. Click Ok

This opens up your 4th rstudio window with an rmarkdown script.


markdown_guidebook - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Untitled1*
Knit
Run
Source or Script
Environment History Connect
Import Dataset
Global Environment
Enviror
Files Plots Packages Help
Zoom Export
Files,
Packag
Console Terminal Jobs
Console
1:1 (Top Level) R Markdown

~~~

install.packages("rmarkdown")

OR

In code chunk as below;

```{r}
install.packages("rmarkdown")
```
~~~

```
R 4.0.4 · C:/Workspace/RMardown_NAPS/Open-NAPs-Data/opennaps/markdown_guidebook/
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> install.packages("rmarkdown")
```

Now your app area should look more or less like this;
The arrangement may however be different. If so, do not worry, it's just semantics.
As you might have seen, you may open other scripts such as R, python, SQL etc in similar manner.

1.1.1 Rstudio windows explained:

- **The Source**
  This is your scripting area.
  You may write, edit and run your code from here.

- **The Console**
  All processing in r happens here. You may this is the r kitchen. When you run your code, this is where it is evaluated.
  You may also use the console to write and execute your code. However, you can not save your code, nor can you edit it (you will have to rewrite it). Thus, it is always advisable to write your code in the script/source area.

- **The Environment**
  This panel holds your data objects and their metadata. As you create or add new objects into your project, they will appear in this window. You may view and remove objects. It also contains the 'History' tab from which you may view all your r processing history.

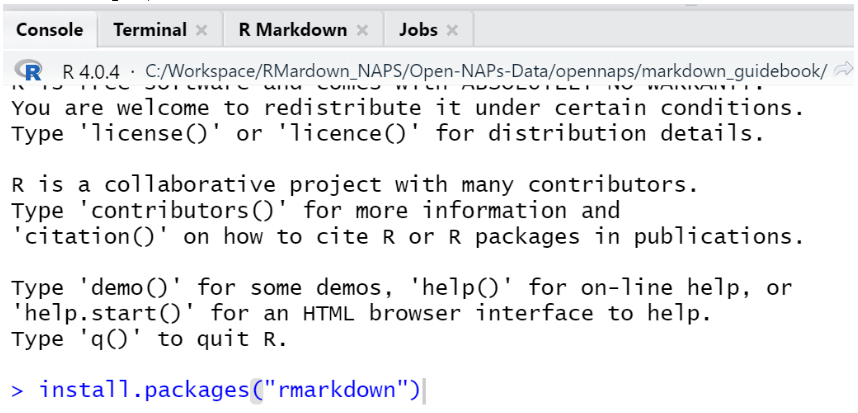
- **The Files/Plots/Packages/Help Panel**
  This panel provides a shortcut/alternative to different functions such as to access your project files, to view plots from your code evaluations, install

packages and access the Help resource. You may add/remove items here but we will learn that much later. For now we leave it as is.

1.2 Installing packages

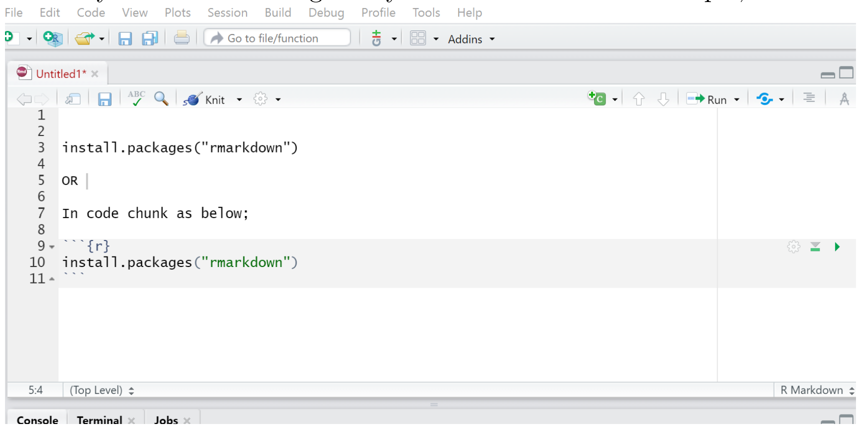
Options:

1. From the console;
   After the greater than (>) sign, write command `install.packages("package name")` and hit enter on your keyboard.
   See example;


Console
Terminal
R Markdown
Jobs
R 4.0.4 · C:/Workspace/RMardown_NAPS/Open-NAPs-Data/opennaps/markdown_guidebook/

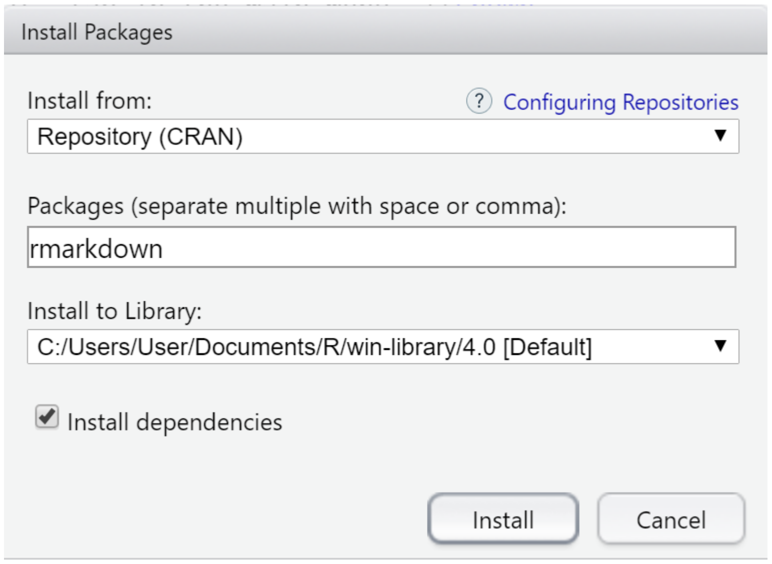
```
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> install.packages("rmarkdown")
```

2. In script window
   Inside your script write command `install.packages("package name")`
   Click on 'Run' (Run button is on the top right of your source panel).
   Alternatively, if your write your command in a code chunk, you may hit the 'Play' button on far right of your code chunk. See example;



~~~

install.packages("rmarkdown")

OR

In code chunk as below;

```{r}
install.packages("rmarkdown")
```
~~~

We see more about code chunks in the next chapter.
3. From the menu bar
From the menu bar, go the the tab 'Tools' and select 'Install Packages'.



Install from: Repository (CRAN)

Packages (separate multiple with space or comma): rmarkdown

Install to Library: C:/Users/User/Documents/R/win-library/4.0 [Default]

✓ Install dependencies

You should see a window like this one

In the 'Install from' field, select 'Repository (CRAN)', then enter the name(s) of packages to be installed.
Leave everything else as default. Click Install.
4. From Files/Plots/Packages/Help Panel
Click on the 'Packages' tab. There are 2 tabs under this, Install & Update.
Click on Install.
If the 'Install from' is not set to 'Repository (CRAN)', set it so, and enter name(s) of packages to be installed.
Leave everything else as default. Click Install.

Note: *Packages can only be installed once. To mean, if you close your r/rstudio app and come back to it the next day/session, you do not need to install the packages you already installed in your previous session. You will only need to load their libraries (see next section)*

1.3 Loading Libraries

To be able to make use of the packages installed, you need to call their libraries. Most libraries take the name of the package. For instance, for the package `rmakdown`, the respective library is `rmarkdown`.To call the rmarkdown library, use command `library(rmarkdown)`. So in general to load/call libraries use command `library(name of library)`.
You may load libraries in the console or in your script.

Note: *Unlike packages,library functions expire when you close a project or end a session. Therefore, each time you open an r session, you have to load/call relevant libraries*

Chapter 2

Getting started

2.1 About rmarkdown

R markdown is a file document that allows you to write, save and execute code, as well as text and figures to help generate reproducible reports that can be shared in several formats. The file extension is .rmd.
A markdown file has three main sections;

1. **YAML header**
   This is where your document metadata go to e.g your document title, author, date, output file type etc. This parameters are set when opening a new .rmd file. And more can be set after. This section MUST always be at the beginning of your document and between a set of three dashes i.e three dashes before section and three dashes after section.

2. **Text**
   Your narration/prose in markdown format. More details about formatting in subsequent sections.

3. **Code chunk(s)**
   They start with “‘{r}

   and end with “‘

2.2 Editing in markdown

2.2.1 Headers

To create headers for your reports/document e.g. chapters, sub-chapters and so on; use the hash sign ‘#’ in front of the title. Sequentially increase the number

of "#' signs to denote subsequent header levels.
Insert blank line before each header (except in the beginning of document).

# Header level 1
## Header level 2
### Header level 3
#### Header level 4

Note: For the purpose of this example, I have added a space before each line so that r doesn't create the headers. Remove the space to create the headers

2.2.2 Bold and italic text

To create emphasis in your markdown texts, use an asterisk before and after text *to italicize* or double asterisk **to make your text bold**.
Alternatively, you may use single underscore *for italics* or double underscore **for bold**.

2.2.3 Create Lists

2.2.3.1 Ordered list

Use numbers to order your list items. and a plus sign '+' to create sub-items on your list.

1. List item 1

2. List item 2

- Sub-item 1

- Sub-item 2

- Sub-item 3

3. List item 3

2.2.3.2 Unordered list

Use an asterisk '*' before list item to create an unordered list.
Use a plus '+' sign for sub-items.
Use tab command or two spaces on your keyboard to indent the list items

- List item
- List item
    - Sub-item
    - Sub-item
        - Sub-sub item
- List item

2.2.4 Manual line breaks

Use two or more spaces at the end of a line
to insert a line break

2.2.5 Insert links

You may insert a link using the plain http address such as https://rmarkdown.rstudio.com/ or insert it as a linked phrase using square brackets and parenthesis such as our link phrase goes here.

2.2.6 Insert figures/images

To insert images to our document, we use the same syntax as links, but start with an exclamation mark'!' before syntax. For an image from a url use; `![text to accompany your image e.g a caption](your https link)`
or for an image file in your local directory use, `[your image text](path to local image file)`.
For instance, I downloaded the UN Climate logo and saved it as a .jpg in my working directory as exact name `unfccc_logo.jpg` The following syntax will insert the logo into my document: `![UN Climate logo](unfccc_logo.jpg)`. However, this draws the logo on entire page width. Therefore to specify the drawing width of the image to x inches, we add the the command '{width=xin}' at the end.

When inserting images from local file, it is strongly recommended to have the file in your working directory.

2.2.7 Insert block quotes

To insert a block quote within your text, use the greater than sign '>' in the beginning of quote. The quote must begin on a new line, and remember to insert blank line before and after. For instance;

> This exercise may seem complicated at first, but trust me, it is not.
> You will agree with me sooner than later :)

2.2.8 Insert code chunks

To write your code use open code chunk with three backticks and the curly brackets {insert your code language}, hit enter on your keyboard, insert your code and hit enter, close the code chunk with another three backticks.
For code in r language;

```
# your code here
```

For code in python;

```
# your code here
```

2.2.9 Create tables

2.2.9.1 Option 1

Create table headers with dashed lines below the header title. Separate headers with tab or space between the headers and corresponding dashed lines.
Type in row values below the dashed lines. The row value length may exceed the dashed line length but MUST not extend into the next header's dashed line.
Column alignment is based on the position of the header/column title relative to the dashed line below it.
To insert a caption or alt text to your table use, full colon ':' followed by your caption text at the end of the table.
Alternatively, use 'Table: your caption or text'.

Table 2.1: you may insert table caption here
Table: Alternative caption option

Header 1	Header 2	Header 3	Header 4
12343	895	0.5867891011	1
Name	Rank	score	remark

Table 2.2: My first simple table with kable

X	Y	Z
20	1.4	yes
30	4.3	no
10	5.9	true
50	2.7	false

Header 1	Header 2	Header 3	Header 4
Type	3	TRUE	12.1
Left align	Right align	Center	Default

2.2.9.2 Option 2

You may also create simple tables using a `knitr` function called `kable`.
The code below tells r that we want to create a data set with 3 columns, X, Y & Z, assigning them the values enclosed in the letter c. The letter c used together with brackets indicates a list of elements. Thus, in the example below, we tell r that our column X, will contain a list of 4 elements i.e 20, 30, 10 & 50.
Then we tell r to create the data set by combining all the columns X, Y & Z into a data frame. Finally, we call the function `kable` and enter the data we created. This function converts or data frame into a table format.
Optionally, you may add a caption to the table and specify cell alignment.

```
X<-c(20,30,10, 50)     # create variable X
Y<-c(1.4, 4.3,5.9,2.7)     # create variable Y
Z<-c("yes","no","true","false")    # create variable Z
mydata<-data.frame(X,Y,Z) # combine variables into table format
knitr::kable(mydata, caption = "My first simple table with kable", align = 'c') # plot table wit
```

Note that our code chunk is labelled 'kable'. You can label code chunks as in the brackets below

(open chunk "'{r your chunk name}

your code

close code chunk with "')

This will be useful for cross-referencing.

2.2.10 Page breaks

Use three or more asterisks or dashes to insert a page break.
Remember to add a blank line before the asterisks or dashes

2.2.11 Process a markdown document to desired output

To create the desired output file document from the markdown format, use the function `render ("your .rmd file name)`.
Alternatively,and most commonly used, is the `Knit` button from the markdown script environment. The button is a blue ball of yarn around a crotchet, and is labeled 'Knit' When a document is rendered, rmarkdown saves the results/output file into your working directory, giving it the same name as your .rmd file, but with relevant extension (e.g. as html if output type was set to html)

2.2.12 References

2.2.12.1 Citations/Bibliography

To add citations to our document, we need to add our reference document details to a text file saved in the .bib format e.g myrefences.bib. We also need to add this file to our bibliography parameter in the yaml section of the index.rmd file. The references for this exercise are saved in the file 'book.bib'.
To add your reference documents to the .bib file, use the BibTeX citation style i.e

```
?{documentname,
title={document title},
author={author name(s)},
year={publication year},
publisher={publisher name},
url={document url}
}
```

Sites such as Google Scholar have ready to use/ formatted bibliography styles such as BibTeX, EndNote etc. You may copy-paste the BibTeX text into your .bib file in r.

You may have as many documents listed in the .bib file but rmarkdown will only include those that have been referenced within your markdown document. To reference documents in your narrative and thus, have them included in the reference section, use the format `[@documentname]`.
For instance, (Xie, 2015) and (Team et al., 2013) are the only references in my book.bib file.
The references will appear at the end of the chapter where they are referenced, as well as in the overall 'References' section.

2.2.12.2 Chapter References

To reference chapter, type in the chapter title inside square brackets. For instance, this command references our first chapter Prerequisites.
The same applies to section headers. Just type [section header name] e.g Page breaks.

2.2.12.3 Table & Figure References

To reference figures use syntax @ref(fig:code chunk label) or @ref(tab:code chunk label) for tables. For instance, we created our kable table in the code chunk named kable. To reference this table anywhere in our document, just type in command 2.2.
This will insert a reference link to table 2.2.

2.2.13 Pandoc & Knitr

`Pandoc` is a universal document converter designed to convert thousands of markup languages. So when we create our document in markdown and want to output it as a pdf, pandoc does the work.
`Knitr` on the other hand, is an r package that enables the integration of yaml, text and code evaluations into an output document. `Knitr` contains the `Knit` function through which we render our rmarkdown documents to our desired output format. When you render a document in rmarkdown (or call the knit function), the rmarkdown document is converted to a basic markdown language (.md) which is then converted by pandoc to say html, pdf, word, etc as per user specifications. `Knitr` and `pandoc` come in bundled with rmarkdown, and thus, there is no need to install them separately.
However, should you need to install pandoc as standalone, you may do so from the Pandoc homepage. In this regard, it is important to note that in as much as standalone installations may provide much higher versions of the software than what is already bundled in r, they are often not streamlined for use in r, and may thus cause some compatibility issues.

Chapter 3

Bookdown

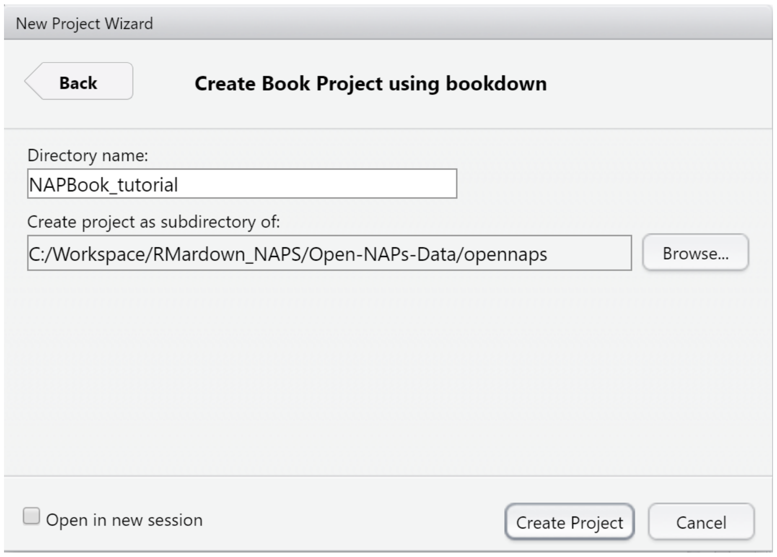
3.1 Setting up an Open NAP document

We will use the package `bookdown` to generate NAP document in a book format. Journal articles or reports can be produced in the same way. We will also use the package `tinytex` to build pdf format of the book document.

1. First, launch the rstudio app in your pc.

2. Install the bookdown and tinytex packages. Use any of the methods shown in the Installing packages section.

3. Then from the menu bar go to `File>->New Project->>New Directory->>Book Project using bookdown`.
   A new window appears



New Project Wizard

Back | **Create Book Project using bookdown**

Directory name: NAPBook_tutorial

Create project as subdirectory of: C:/Workspace/RMardown_NAPS/Open-NAPs-Data/opennaps

Browse...

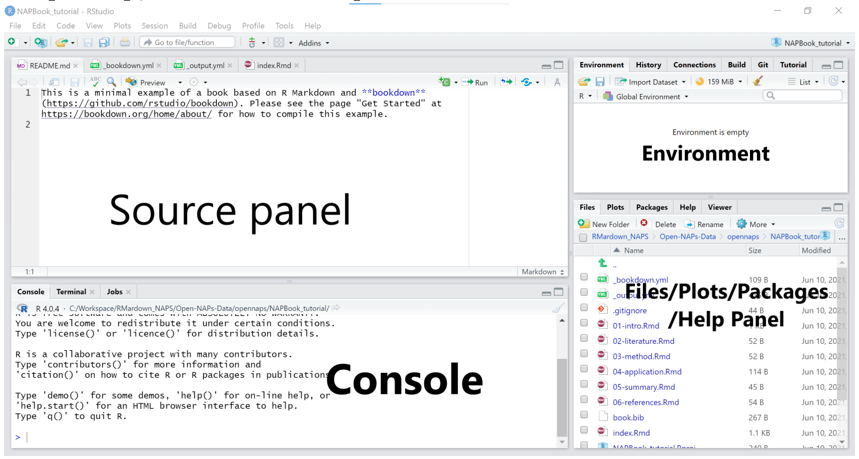
☐ Open in new session

Create Project | Cancel

4. Give your bookdown directory a name 'NAPBook_tutorial'. This is the name of a new folder that will be created to store your bookdwon project files.

5. In the 'Create project as subdirectory of' field, use the browse button to navigate to where you want to save your bookdown project folder.

6. After you select the directory location, you are returned to the 'Create Book project...' window. Click 'create project'.
A new project session opens up, with skeleton chapters and other sections



```
This is a minimal example of a book based on R Markdown and **bookdown**
(https://github.com/rstudio/bookdown). Please see the page "Get Started" at
https://bookdown.org/home/about/ for how to compile this example.
```

R 4.0.4 · C:/Workspace/RMardown_NAPS/Open-NAPs-Data/opennaps/NAPBook_tutorial/

```
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Environment is empty

RMardown_NAPS > Open-NAPs-Data > opennaps > NAPBook_tutorial

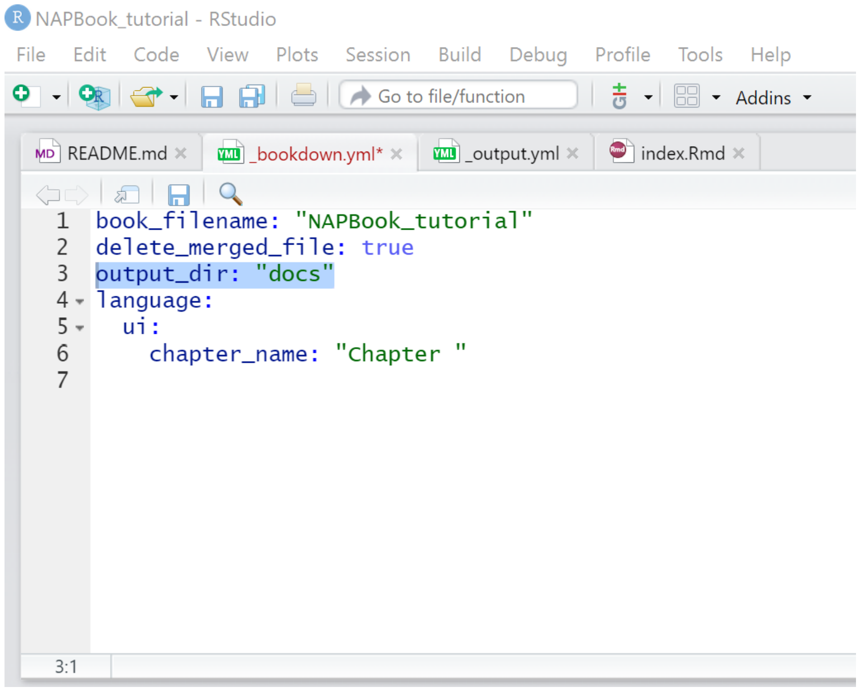
Name	Size	Modified
..		
_bookdown.yml	109 B	Jun 10, 2021
_output.yml	[illegible]	Jun 10, 2021
.gitignore	44 B	Jun 10, 2021
01-intro.Rmd	[illegible]	Jun 10, 2021
02-literature.Rmd	52 B	Jun 10, 2021
03-method.Rmd	52 B	Jun 10, 2021
04-application.Rmd	114 B	Jun 10, 2021
05-summary.Rmd	45 B	Jun 10, 2021
06-references.Rmd	54 B	Jun 10, 2021
book.bib	267 B	Jun 10, 2021
index.Rmd	1.1 KB	Jun 10, 2021

and metadata files.
Your project window might appear with panel arrangements different from mine/in screenshot. Your project files can be accessed from the Files

tab in bottom right of the project window. Each chapter is compiled from a single .rmd file, and always starts with a first level header sign '#'. From the files pane, you can open any chapter, and edit it to your liking. Lets take a look at the '_bookdown.yml' file.

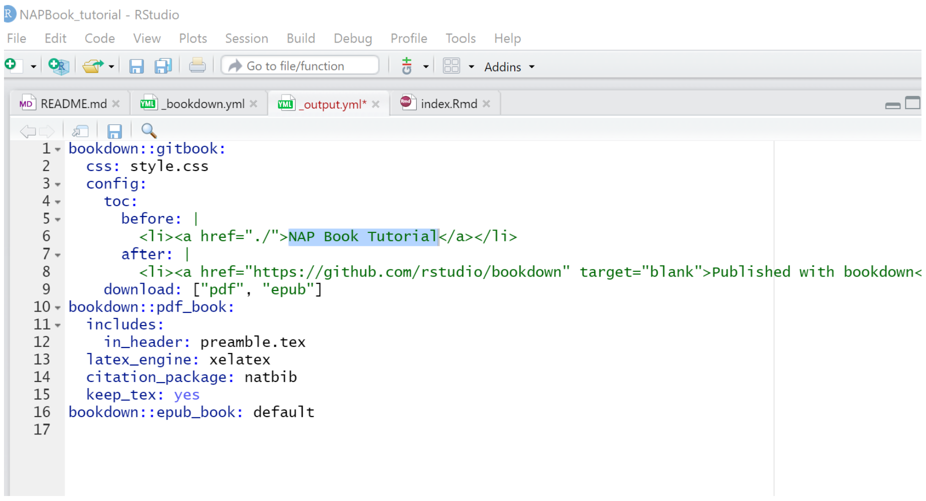
7. From the 'Files' tab, click on the _bookdown.yml' file.
   The file open up in your script/source pane. Notice, the book_filename takes the name of the folder we created when opening the project.

8. Let's add the line `output_dir: "docs"` to this file. See screenshot below;


NAPBook_tutorial - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
README.md
_bookdown.yml*
_output.yml
index.Rmd
3:1

```
book_filename: "NAPBook_tutorial"
delete_merged_file: true
output_dir: "docs"
language:
  ui:
    chapter_name: "Chapter "
```

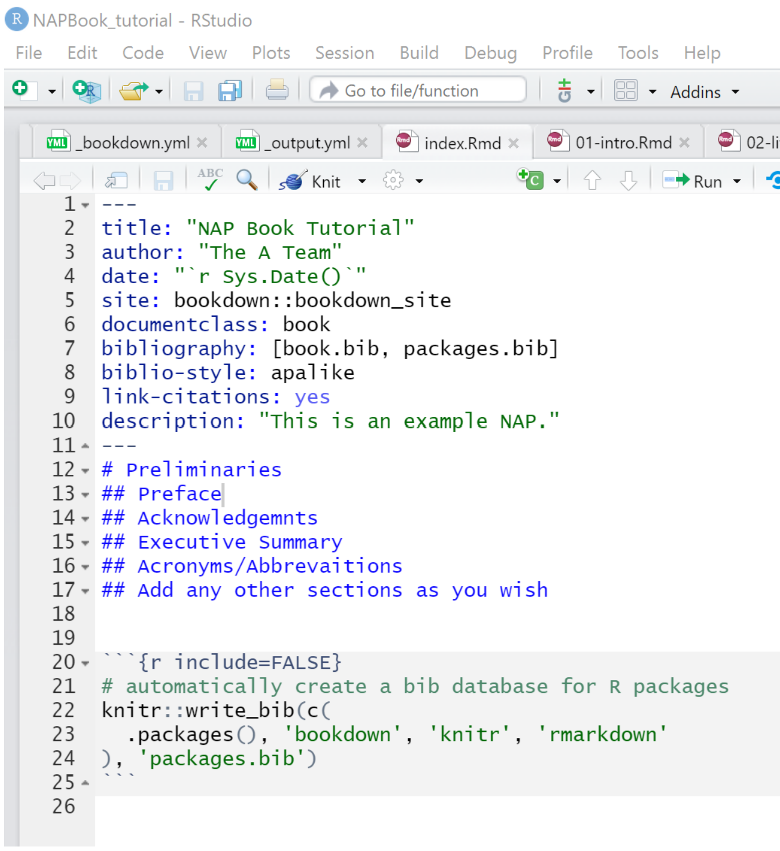
This command creates an additional folder called 'docs' and specifies that we want to store our outputs here. This folder is necessary for creating webpages from our book project.
9. Save changes. Use the 'Save' icon on the script window or Ctrl + S on your keyboard.
10. Next, click to open the '_output.yml' file.
11. Let's edit the line that contains the title to our book. Change the text 'A Minimal Book ...' to 'NAP Book Tutorial'.


NAPBook_tutorial - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
README.md
_bookdown.yml
_output.yml*
index.Rmd

```
bookdown::gitbook:
  css: style.css
  config:
    toc:
      before: |
        <li><a href="./">NAP Book Tutorial</a></li>
      after: |
        <li><a href="https://github.com/rstudio/bookdown" target="blank">Published with bookdown<
    download: ["pdf", "epub"]
bookdown::pdf_book:
  includes:
    in_header: preamble.tex
  latex_engine: xelatex
  citation_package: natbib
  keep_tex: yes
bookdown::epub_book: default
```

Should look like this;
Ignore everything else for now :)
12. Save.
13. Scroll down your files to find the 'index.rmd' file. Click to open.
14. Edit the title, author & description fields in the yaml header section.
15. Delete all text, except the last code chunk.
16. Add a chapter named 'Preliminaries' before the code chunk.
Chapters are created using the first level header #.
17. Add sections to your chapter using second level header ##.

Your script shoud look more or less similar to this;


NAPBook_tutorial - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
_bookdown.yml
_output.yml
index.Rmd
01-intro.Rmd
Knit
Run

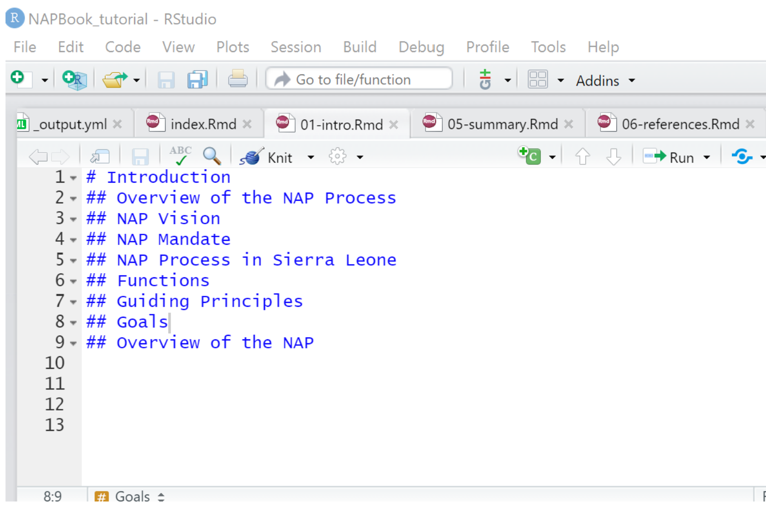
~~~
---
title: "NAP Book Tutorial"
author: "The A Team"
date: "`r Sys.Date()`"
site: bookdown::bookdown_site
documentclass: book
bibliography: [book.bib, packages.bib]
biblio-style: apalike
link-citations: yes
description: "This is an example NAP."
---
# Preliminaries
## Preface
## Acknowledgemnts
## Executive Summary
## Acronyms/Abbrevaitions
## Add any other sections as you wish


```{r include=FALSE}
# automatically create a bib database for R packages
knitr::write_bib(c(
  .packages(), 'bookdown', 'knitr', 'rmarkdown'
), 'packages.bib')
```

~~~

18. Next, open the 'README.md' file and edit the text to this 'This is an example NAP designed to help you get started with creating your NAPs with bookdown.'. Or add a description of your choice.
19. Save your work.

20. Open the '01-Intro.rmd' file.

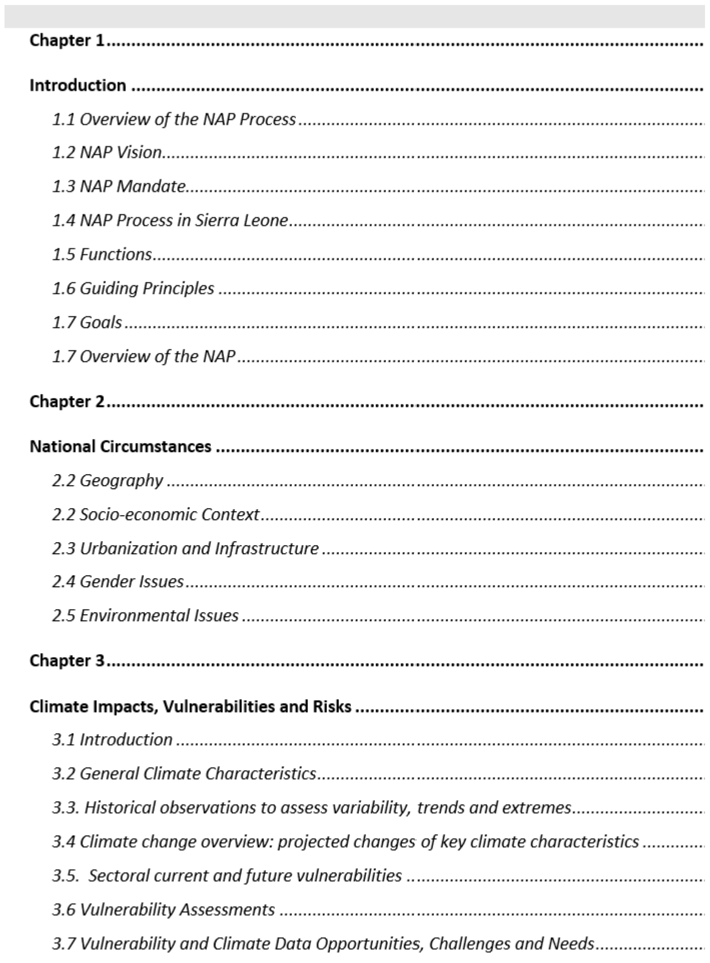
21. Type in the chapter title 'Introduction'

22. Type in sub-chapters with the names 'Overview of the NAP Process, NAP Vision, NAP Mandate, NAP Process in Sierra Leone, Functions, Guiding Principles, Goals, Overview of the NAP'

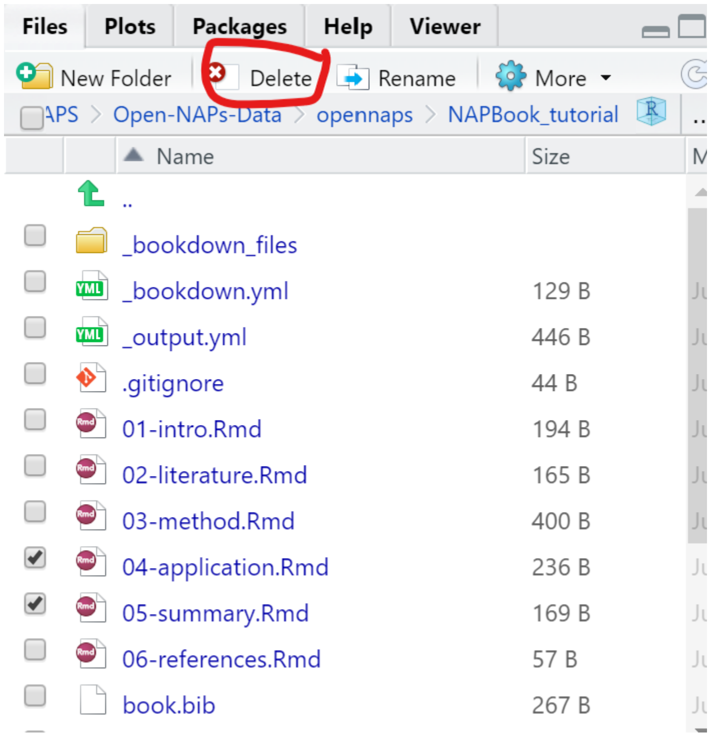


```
# Introduction
## Overview of the NAP Process
## NAP Vision
## NAP Mandate
## NAP Process in Sierra Leone
## Functions
## Guiding Principles
## Goals
## Overview of the NAP
```

Your script should resemble this;
This text has been borrowed from the Sierra Leone NAP. Use the text from the screenshot below to edit the next two chapters (.rmd files). i.e '02-literature.rmd & 03-methods.rmd'.



For the interest of time, let's delete chapters 4 and 5 from our project files.
23. From the files tab (bottom right), click on the check-boxes to the left of the application and summary .rmd files to select them. 24. Click delete

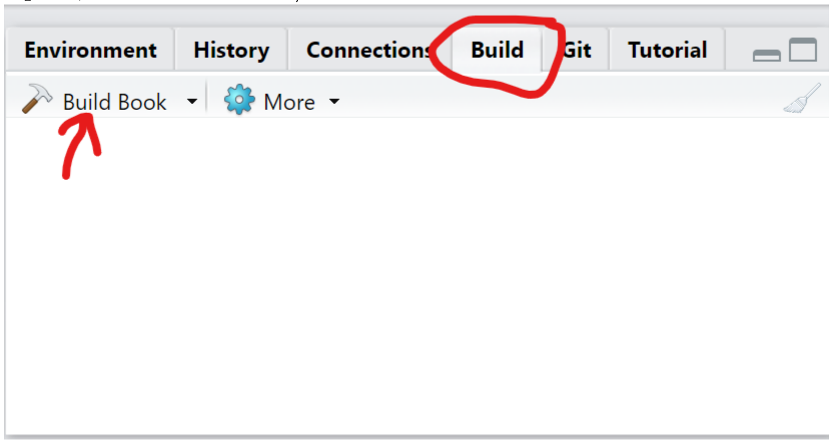


Name	Size
..	
_bookdown_files	
_bookdown.yml	129 B
_output.yml	446 B
.gitignore	44 B
01-intro.Rmd	194 B
02-literature.Rmd	165 B
03-method.Rmd	400 B
✓ 04-application.Rmd	236 B
✓ 05-summary.Rmd	169 B
06-references.Rmd	57 B
book.bib	267 B

25. Save project.

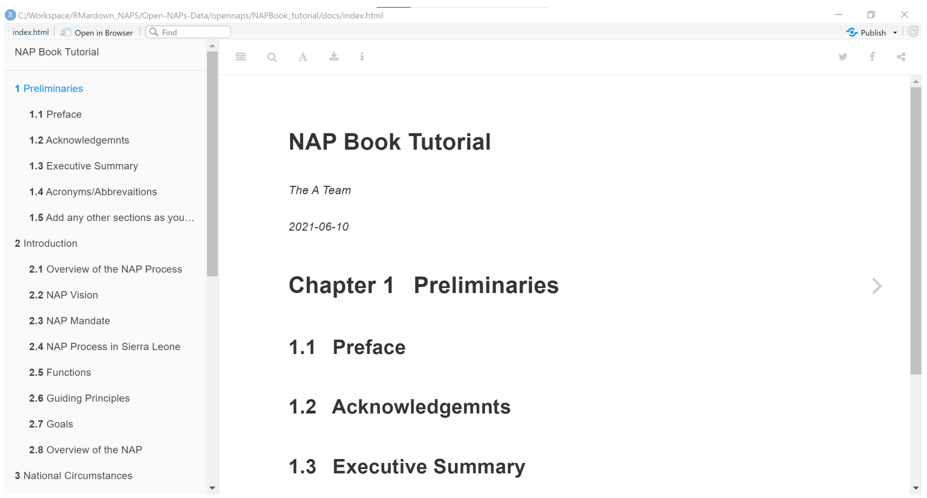
26. From the 'Environment' window, click on the tab 'Build'.

27. On the new window that opens, click the hammer/Build Book button to



start rendering the book.
This process may take a few minutes. You can monitor the progress in the Environment window.
When the process is done, your book should open up in an html window

as below;


C:/Workspace/RMardown_NAPS/Open-NAPs-Data/opennaps/NAPBook_tutorial/docs/index.html
index.html
Open in Browser
Find
Publish

NAP Book Tutorial

1 Preliminaries
1.1 Preface
1.2 Acknowledgemnts
1.3 Executive Summary
1.4 Acronyms/Abbrevaitions
1.5 Add any other sections as you…
2 Introduction
2.1 Overview of the NAP Process
2.2 NAP Vision
2.3 NAP Mandate
2.4 NAP Process in Sierra Leone
2.5 Functions
2.6 Guiding Principles
2.7 Goals
2.8 Overview of the NAP
3 National Circumstances

NAP Book Tutorial

The A Team

2021-06-10

Chapter 1 Preliminaries

1.1 Preface

1.2 Acknowledgemnts

1.3 Executive Summary

Figure 3.1: Your book

28. Scroll down your chapters to find the 'References' chapter. Oops! No reference! Don't sweat.

29. Back to our project, from the Files tab in bottom right window, open the file named 'book.bib'. This is the text file holding our citations.Currently, only one. You will notice that the document type is a book, and the the book name is 'xie2015'.

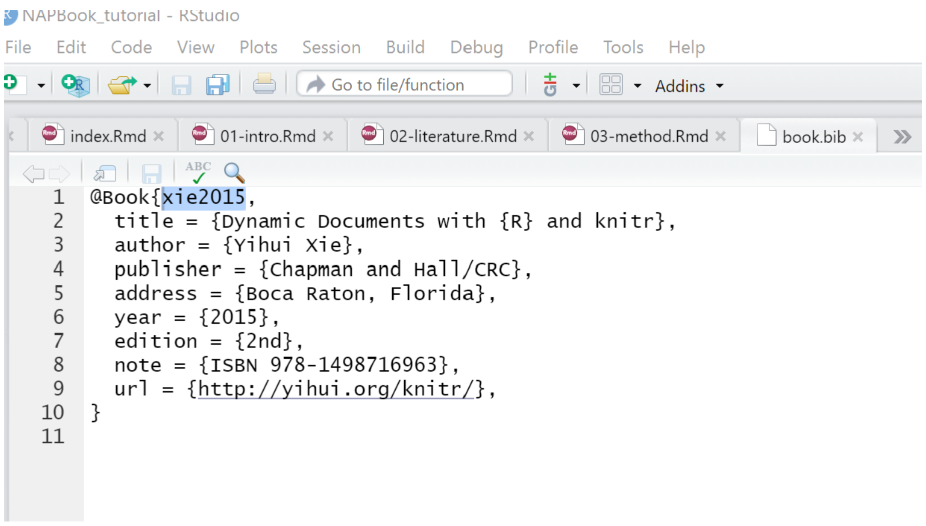
30. To cite this book in our NAP book, open square brackets and type the '@' sign followed by the book name. i.e (Xie, 2015).

31. Let's use this citation say in the Introduction chapter. Open the chapter, and insert the text 'Our only citation at the moment is (Xie, 2015)'. See example:

32. Run the 'Build Book' command again.

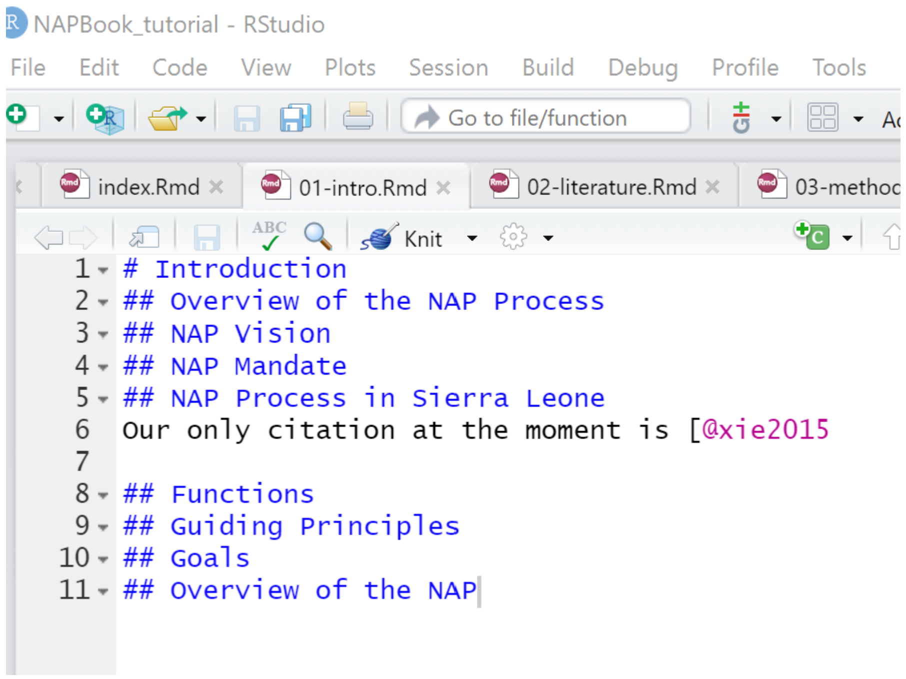
33. Now you should see that your book has been added to the 'References' section, both in the chapter in which you cited it as well as in the overall References Chapter.

Congratulations!!!


NAPBook_tutorial - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
index.Rmd
01-intro.Rmd
02-literature.Rmd
03-method.Rmd
book.bib

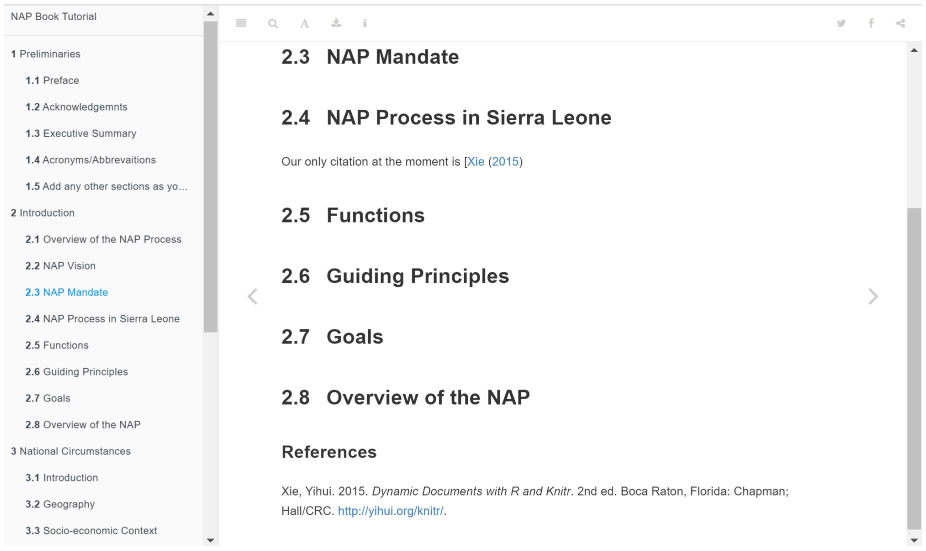
```
@Book{xie2015,
  title = {Dynamic Documents with {R} and knitr},
  author = {Yihui Xie},
  publisher = {Chapman and Hall/CRC},
  address = {Boca Raton, Florida},
  year = {2015},
  edition = {2nd},
  note = {ISBN 978-1498716963},
  url = {http://yihui.org/knitr/},
}
```

Figure 3.2: Adding bibliography


NAPBook_tutorial - RStudio
File Edit Code View Plots Session Build Debug Profile Tools
Go to file/function
index.Rmd
01-intro.Rmd
02-literature.Rmd
03-methoc
Knit

```
# Introduction
## Overview of the NAP Process
## NAP Vision
## NAP Mandate
## NAP Process in Sierra Leone
Our only citation at the moment is [@xie2015

## Functions
## Guiding Principles
## Goals
## Overview of the NAP
```

Figure 3.3: cite book



2.3 NAP Mandate

2.4 NAP Process in Sierra Leone

Our only citation at the moment is [Xie (2015)

2.5 Functions

2.6 Guiding Principles

2.7 Goals

2.8 Overview of the NAP

References

Xie, Yihui. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. http://yihui.org/knitr/.

Figure 3.4: citation added to ref

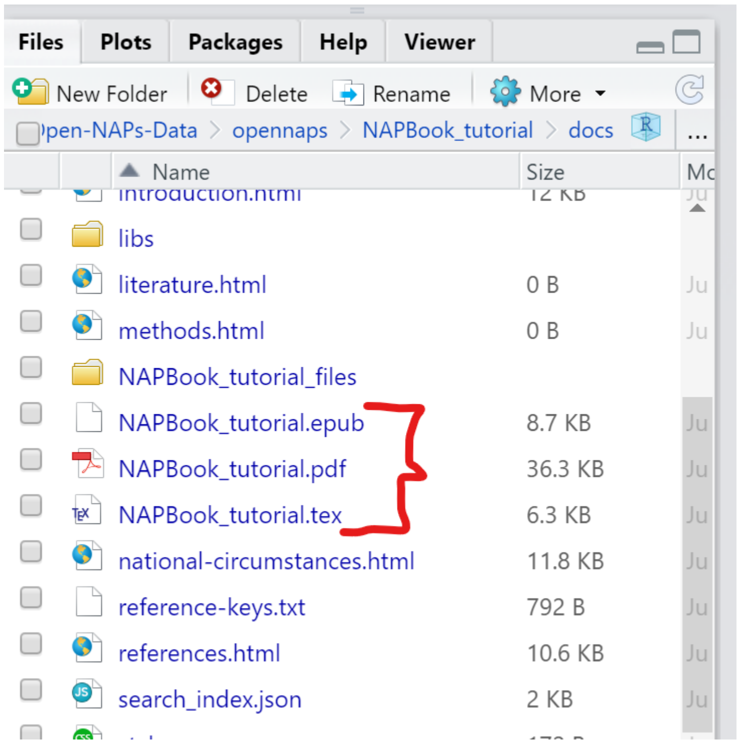
3.1.1 Some things to note

1. To create additional chapters, create a new /empty .rmd file and save it under your bookdown directory. Use the same numbering structure as the default chapters (i.e. 01,02,03, etc).
   We will learn about advanced numbering and re-ordering chapters later.

2. Use the `knit` function to render and preview a single chapter.

3. From our output.yml file, you will notice that we have 3 output options for our file; gitbook, pdf and epub. This is the default. You may select the most preferred output type by deleting the other/unwanted formats or leave this as default and choose a single output format when building the book. Since we left the default 3 options, when we rendered the book, it was compiled in .tex/gitbook, pdf and epub. This can be accessed from the 'docs' folder.

4 When building the book. Here you may choose to build one or all formats. From the Build Book button, click on the drop-down arrow and select your preferred format.



Name	Size
introduction.html	12 KB
libs	
literature.html	0 B
methods.html	0 B
NAPBook_tutorial_files	
NAPBook_tutorial.epub	8.7 KB
NAPBook_tutorial.pdf	36.3 KB
NAPBook_tutorial.tex	6.3 KB
national-circumstances.html	11.8 KB
reference-keys.txt	792 B
references.html	10.6 KB
search_index.json	2 KB

Figure 3.5: Outputs

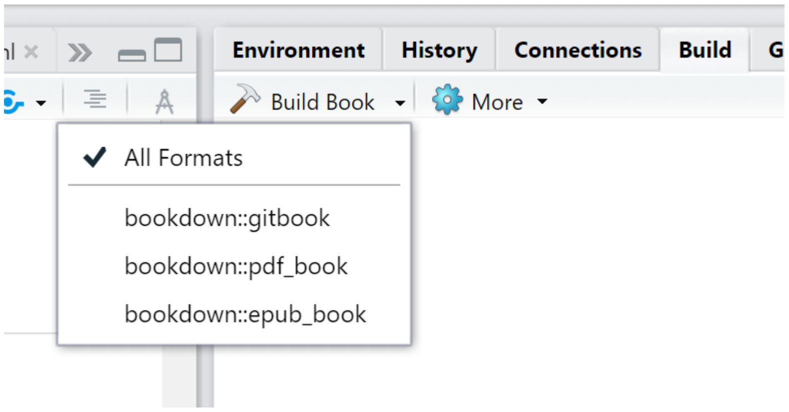


Figure 3.6: Build book format options

3.2 Some Troubleshooting

Creating pdf documents using LaTeX engines and distributions such as TinyTeX is not a straightforward task and may produce errors as the process involves multiple processing activities. However, most errors can be solved using suggestions contained in the error messages produced in the Environment or Console windows.
If, in any case, the LaTeX error generated is not clear, you may use any the commands options below to solve the problem.

Remove the # sign to run code

```
# remotes::install_github('yihui/tinytex') ## 1 install the development version of tinytex

# update.packages(ask = FALSE, checkBuilt = TRUE) #3 2 update your r and
# tinytex::tlmgr_update() ## tinytex packages

# tinytex::reinstall_tinytex()  ## 3 reinstall tinytex

# options(tinytex.verbose = TRUE) ## 4 set this option in an r code chunk.This helps provide more
```

See more on TinyTex Debugging.

Additionally, quite often, LaTeX formatting is not very compatible with other output formats such as html. But with the use of `html widgets` and other advanced formatting options, this problem can be overcome. To produce a pdf document from a document with both LaTeX and html formats, it may be useful to install the package `webshot` from CRAN.

```
# install.packages("webshot")
# webshot::install_phantomjs()
```

Further reading on html widgets here

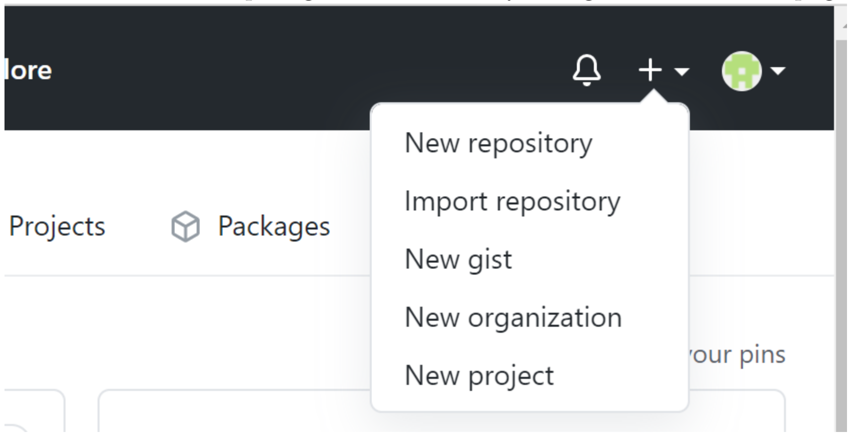
Chapter 4

GitHub & GitHub Pages

4.1 Sharing on GitHub

Github is great for project collaboration, backup and version control. To use github as your repository manager;

1. Create an account at (https://github.com/).
2. Create a repository for your files. Click on the plus sign + located at the top right corner of your github account page.



3. Click on 'New Repository' from the drop-down menu.

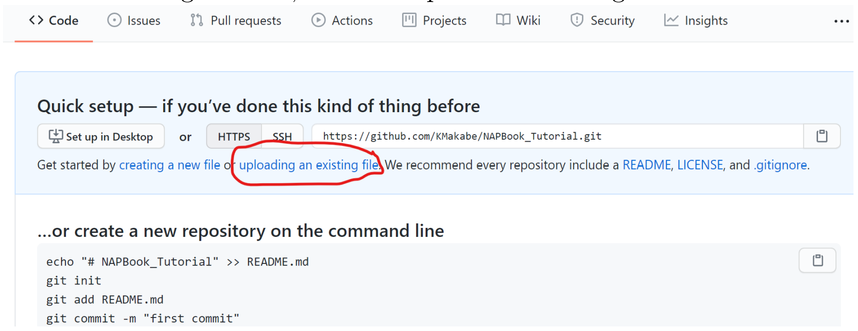
4. On the new window that appears, give the repository a name: NAP-Book_Tutorial

5. Leave the repository as 'Public'

6. Leave everything else as is

7. Scroll down and click on ‘Create Repository’

8. On the resulting window, click on ‘upload an existing file’



Quick setup — if you've done this kind of thing before

https://github.com/KMakabe/NAPBook_Tutorial.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# NAPBook_Tutorial" >> README.md
git init
git add README.md
git commit -m "first commit"
```

This will take you to a new window, from which you can drag-&-drop or browse to your files. 9. Click on ‘choose your files’.
10. Navigate with the file explorer to where we saved our ‘NAPBook_tutorial’ folder in your pc.
11. Select everything within this folder, and click ‘Open’.
Your files will start loading. Give it a minute or so.
If you scroll through the files you will notice that our ‘docs’ folder is missing. Do not worry!
12. To add the folder, lets use the drag and drop method. From your file explorer, navigate to our NAPBook_tutorial folder, and click once on the ‘docs’ folder to select it. 13. Drag and drop this folder into your github repository. The files should start uploading.
14.. After your files finish uploading, scroll down to the ‘Commit changes’ field; here you may enter a short description for your files. Let’s enter the text ‘our first NAP book commit’
When making changes to your files, you may use this field to briefly describe what changes you made, otherwise commonly known as commits.
6. Next, hit the ‘Commit changes’ button at the end. This is called commiting.

Congratulation once again!

To share your repository with colleagues and friends, just copy the link on your browser and share it with them. The link should take a similar identity as below;

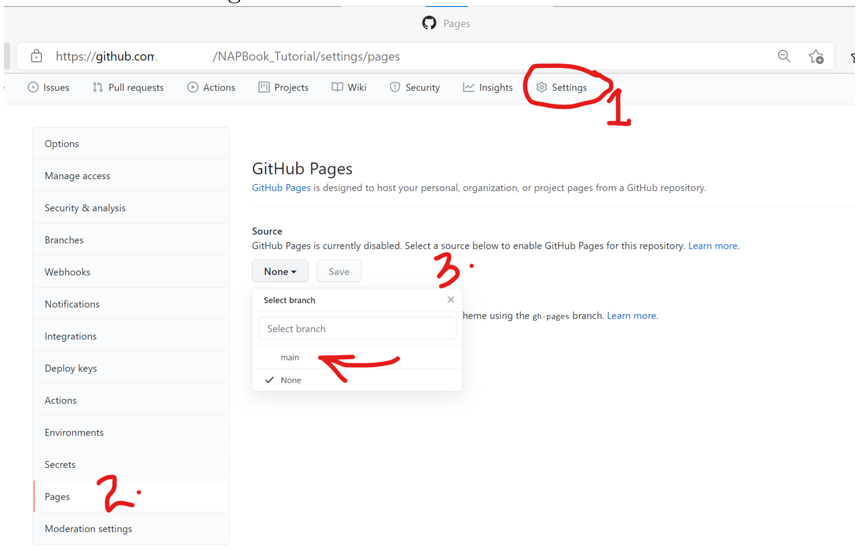
https://github.com/yourusername/NAPBook_Tutorial

4.2 Publishing to GitHub pages

Github pages helps you to create/publish websites in very simple steps. We will publish the NAP book we just created with bookdown into a git-based website. To do this,

1. From the github repository you created in last step, click on the Settings tab (right side of your screen)

2. Scroll down the listed menu items on the left side of the screen until you find menu item 'Pages'. Click on it.



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

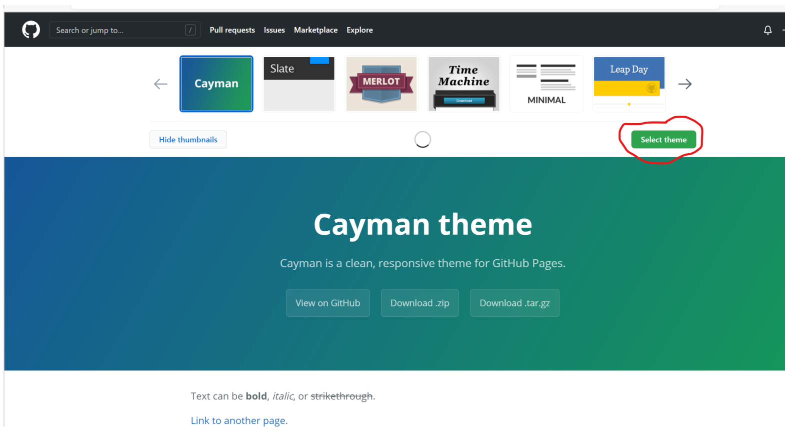
Source

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. Learn more.

heme using the gh-pages branch. Learn more.

3. Scroll down to the 'Source' field. Click on the drop-down arrow and select the **main** branch and **docs** folder as your source files for your website. Click Save.

4. Next, on the 'Theme Chooser' field, click to choose a theme for your website.



Cayman theme

Cayman is a clean, responsive theme for GitHub Pages.

Text can be **bold**, *italic*, or ~~strikethrough~~.

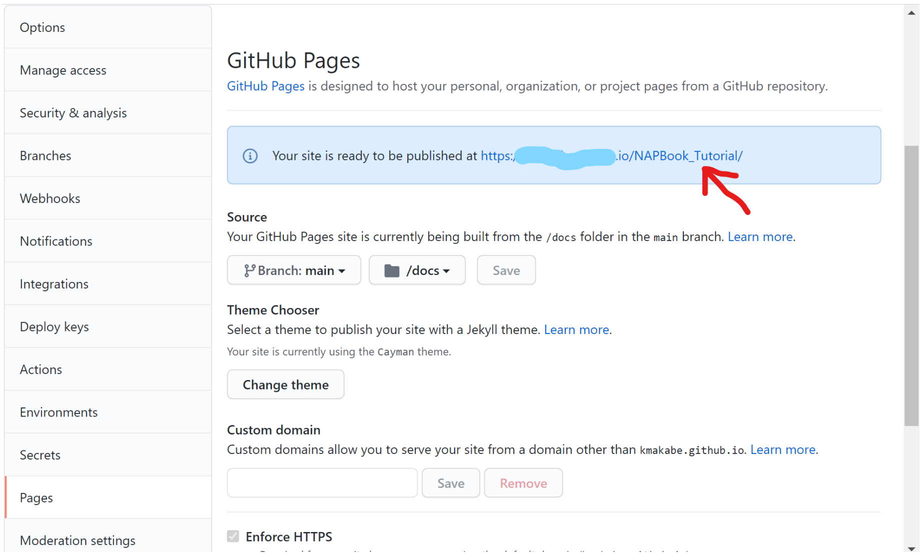
Link to another page.

A page similar to this appears.

You may scroll through to see the options. For this case, lets select the first theme 'Cayman'.

4. Once a theme is selected, a message with a link to your website appears just above your 'Source' field.

   Your site is ready to be published at https://yourusername.github.io/repositoryname/



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is ready to be published at https:[redacted].io/NAPBook_Tutorial/

Source

Your GitHub Pages site is currently being built from the `/docs` folder in the `main` branch. Learn more.

Theme Chooser

Select a theme to publish your site with a Jekyll theme. Learn more.

Your site is currently using the `Cayman` theme.

Custom domain

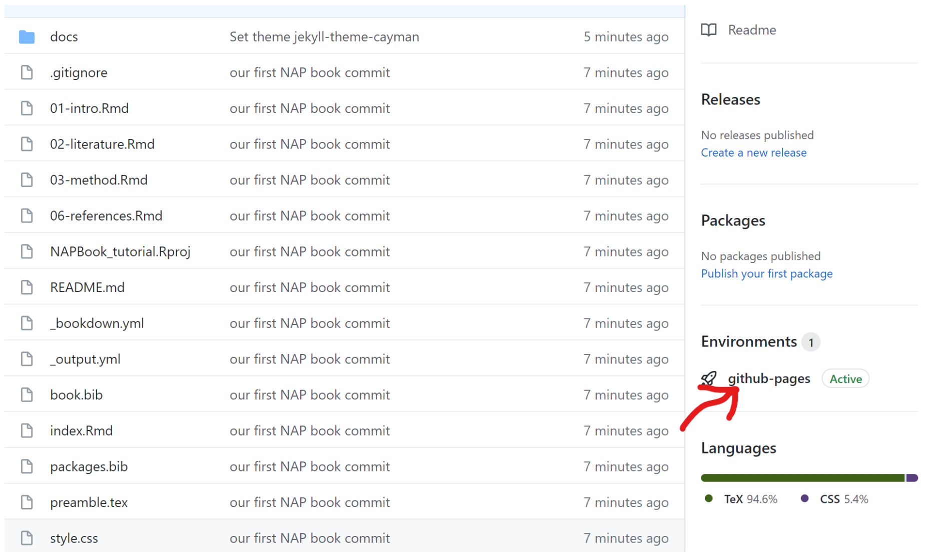
Custom domains allow you to serve your site from a domain other than `kmakabe.github.io`. Learn more.

Enforce HTTPS

or as seen here

Use this link to view your newly created website.
Alternatively, navigate back to your main repository area, scroll down to your right to find your active 'github-pages'. Click to view your website deployments.

Now give yourself a pat on the back!



docs	Set theme jekyll-theme-cayman	5 minutes ago
.gitignore	our first NAP book commit	7 minutes ago
01-intro.Rmd	our first NAP book commit	7 minutes ago
02-literature.Rmd	our first NAP book commit	7 minutes ago
03-method.Rmd	our first NAP book commit	7 minutes ago
06-references.Rmd	our first NAP book commit	7 minutes ago
NAPBook_tutorial.Rproj	our first NAP book commit	7 minutes ago
README.md	our first NAP book commit	7 minutes ago
_bookdown.yml	our first NAP book commit	7 minutes ago
_output.yml	our first NAP book commit	7 minutes ago
book.bib	our first NAP book commit	7 minutes ago
index.Rmd	our first NAP book commit	7 minutes ago
packages.bib	our first NAP book commit	7 minutes ago
preamble.tex	our first NAP book commit	7 minutes ago
style.css	our first NAP book commit	7 minutes ago

Readme

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Environments 1

github-pages Active

Languages

● TeX 94.6% ● CSS 5.4%

Figure 4.1: gh pages deployment

Chapter 5

Suggested Reads

Here we include a list of links to articles and other material that might be useful to your journey'

1. R mardown basics
2. Bookdown basics
3. GitHub for bookdown

Bibliography

Team, R. C. et al. (2013). R: A language and environment for statistical computing.

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.