

An introduction to markdown basics

Yihui Xie

2021-06-09

Contents

1	Prerequisites	5
1.1	Opening a markdown file	5
1.2	Installing packages	5
1.3	Loading Libraries	6
2	Getting started	7
2.1	About rmarkdown	7
2.2	Editing in markdown	8
3	Bookdown	13
3.1	How to set up an Open NAP document	13
3.2	Important to note	14
4	GitHub & GitHub Pages	15
4.1	Sharing on GitHub	15
4.2	Publishing to GitHub pages	15
5	Suggested Reads	17

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

Steps;

1. Launch your r/studio app.
2. Install rmarkdown package, use command `install.packages("rmarkdown")`
3. If you intend to create .pdf documents install TinyTeX, use command `install.packages("tinytex")`

1.2 Installing packages

In markdown, packages are installed within a code chunk, i.e, between `{r}` and.

To install a package, use command `install.packages("name of package")`.

Alternatively, from the menu bar, go to the tab 'Tools' and select 'Install Packages'.

In the 'Install from' field, select 'Repository (CRAN)', then enter the name(s) of packages to be installed, and leave the 'Install to Library' field as default.

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 **rmarkdown**, the respective library is **rmarkdown**.

To call the library, use command `library(rmarkdown)`. So in general to load/call libraries use command `library(name of library)`.

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 end with “`”`”

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).

2.2 Editing in markdown

2.2.1 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**.

2.2.2 Create Lists

2.2.2.1 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

2.2.2.2 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

2.2.3 Manual line breaks

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

2.2.4 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.

2.2.5 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 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.

2.2.6 Insert block quotes

To insert a block quote within your text, use the greater than sign '>' in the beginning of quote. For instance;

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

2.2.7 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
```

2.2.8 Create tables

2.2.8.1 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.

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
Type	3	TRUE	12.1
Left align	Right align	Center	Default

2.2.8.2 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.

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

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 our 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 with
```

2.2.9 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

2.2.10 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`)

2.2.11 References

To do later

2.2.12 Pandoc & Knitr

Pandoc is a universal document converter designed to convert thousands of markup languages. So when we create our document in mrakdown and wan 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 rmarkdoen (or call the knit function), the r markdoen 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 rmakrdown, 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 How to set 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.

1. First, we install the bookdown package, use command `install.packages("bookdown")`.
2. Then from the menu bar go to **File>>New Project>>New Directory>>Book Project using bookdown**. Give your bookdown directory a name and click ‘create project’.
3. A new project session opens up, with skeleton chapters and other sections and metadata files. Each chapter is compiled from a single .rmd file. From the files pane, you can open any chapter, and edit it to your liking.
4. To add additional chapters, create a new .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. 5. Use the `knit` function to render and preview a single chapter. 6. 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. I recommend to leave the default 3 options 7. From the menu bar click on the ‘Build’ tab to compile your book. 8. Here you may choose to build one or all formats.

9. For the purpose of publishing our NAP to github pages later, we will need to create an extra docs folder in our bookdown project files. 10. To do so, navigate to and open the ‘bookdown.yml’ file and add the line

```
output_dir: "docs"
```

Remember to keep saving your project.

3.2 Important to note

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 error can be solved using suggestions contained in the error messages produced.

If, in any case, the LaTeX error generated is not clear, check out options provided here for TinyTex Debugging.

```
# 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 p
```

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
3. On the new repository, click on the ‘Add File’ drop-down menu, select ‘Upload files’.
4. This will take you to a new window, from which you can drag-&-drop or browse to your files.
5. After your files finish uploading, scroll down to the ‘Commit changes’ field; here you may enter a short description for your files. When making changes to your files, you may use this field to briefly describe what changes you made. This is commonly known as committing.
6. When done, hit the ‘Commit changes’ button at the end.

4.2 Publishing to GitHub pages

GitHub pages helps you to create/publish websites in very simple steps. We will publish our book/NAP document 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
3. Scroll down to the 'Source' field. Here, select the **main** branch and **docs** folder as your source files for your website. Click Save.
4. Next, choose a theme for your website.
5. 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/>

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

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 markdown basics
2. Bookdown basics
3. GitHub for bookdown