

# R Module 3

Connor Gibbs\*

05 Mar, 2021, 01:59 PM

## Contents

<b>1 Welcome!</b>	<b>1</b>
1.1 Special Boxes . . . . .	2
<b>2 R Markdown</b>	<b>3</b>
2.1 Background . . . . .	3
2.2 Creating an R Markdown Document . . . . .	5

## 1 Welcome!

Hi, and welcome to the R Module 3 course at Colorado State University!

This course is the third of three 1 credit courses intended to introduce the R programming language, specifically R Markdown.

Through this Module (course), we'll explore how R can be used to do the following:

1. Create reproducible reports so you can explain your work in a narrative form
2. Track, accept, and revert changes through version control
3. Share code and collaborate with others

In addition, you'll also be exposed to broader concepts, including:

- 1.
- 2.
- 3.
- 4.

### 1.0.1 How To Navigate This Book

To move quickly to different portions of the book, click on the appropriate chapter or section in the the table of contents on the left. The buttons at the top of the page allow you to show/hide the table of contents, search the book, change font settings, download a pdf or ebook copy of this book, or get hints on various sections of the book. The faint left and right arrows at the sides of each page (or bottom of the page if it's narrow enough) allow you to step to the next/previous section. Here's what they look like:

---

\*Department of Statistics, Colorado State University, [connor.gibbs@colostate.edu](mailto:connor.gibbs@colostate.edu)

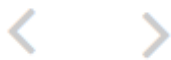


Figure 1: Left and right navigation arrows


### 1.0.2 Associated CSU Course


This bookdown book is intended to accompany the associated course at Colorado State University, but the curriculum is free for anyone to access and use. If you're reading the PDF or EPUB version of this book, you can find the “live” version at <https://csu-r.github.io/Module3/>, and all of the source files for this book can be found at <https://github.com/CSU-R/Module3>.


If you're not taking the CSU course, you will periodically encounter instructions and references which are not relevant to you. For example, we will make reference to the Canvas website, which only CSU students enrolled in the course have access to.


## 1.1 Special Boxes

Because I have a poor memory, I am putting the boxes here as a reference while I write this.

 **Reflect** This box will prompt you to pause and reflect on your experience and/or learning. No feedback will be given, but this may be graded on completion.

 **Assessment** This box will signify a quiz or assignment which you will turn in for grading, on which the instructor will provide feedback.

 **Progress Check** This box is for checking your understanding, to make sure you are ready for what follows.

 **Video** This box is for displaying/linking to videos in order to help illustrate or communicate concepts.

 **Caution** This box will warn you of possible problems or pitfalls you may encounter!

 **Bonus** This box is to provide material going beyond the main course content, or material which will be revisited later in more depth.

● **Feedback** This box will prompt for your feedback on the organization of the course, so we can improve the material for everyone!  
Any of the boxes may include hyperlinks like this: [I am a link](#) or code like this `This is code`.

## 2 R Markdown

“The success of your presentation will be judged not by the knowledge you send but by what the listener receives.” —Lilly Walters, Author

Most scientific disciplines require the investigator to relay her findings through reports or presentations. These reports often consist of human readable text, such as the investigator’s description of methods. Tables and figures are often used throughout to support the investigator’s description or perhaps to convey the results. Sometimes the investigator may even be required to provide code, perhaps illustrating how her methods can be implemented. Of course, all of these text, graphics, and code are well organized by headers and subheaders where certain information is highlighted by adjusting the placement (e.g. centering figures or spacing paragraphs) or changing the topography (e.g. switching fonts, bolding definitions, or italicizing jargon).

In this chapter, we will introduce R Markdown, an environment allowing you create reports, presentations, and much more from the comfort of RStudio. This allows you to combine information such as text, code, and figures easily and reliably.

⚠ **Caution** Learning R Markdown may feel daunting at first. Unlike Microsoft Office products, you will not be able to click and drag to move items. Instead, the “look” of your R Markdown document are generated by templates and personalized by the syntax of your R Markdown file. With practice, you will learn how to customize your documents to your liking and integrating information from different mediums (e.g. text and code) will be easier than ever before.

### 2.1 Background

To understand R Markdown, we first need to tackle **Markdown**. According to its website, Markdown is a markup language you can use to format virtually any document. Unlike Microsoft Word and other **WYSIWYG** editors, you make changes to a document by adding **syntax** to text rather than using buttons and menus. In essence, Markdown syntax are code which need to be processed before the content can be viewed as HTML. One such markdown processor is pandoc which we will see again soon.

To simplify things, **R Markdown** is a variant of Markdown created by **RStudio** to incorporate the functionality of Markdown and **R**. With R Markdown, you can combine human readable text, code, and figures to create a document. While R Markdown is incredibly robust, allowing you to make reports, presentations, and even dashboards, we will start by focusing on the simplest case: creating a report. Compiling an R Markdown document requires your computer to process the Markdown code used to create human readable text, R code used to conduct analysis, and the R code output used to portray results. We will reserve the technical details for [Xie et al. \(2020\)](#).

It is easy to confuse R, R Markdown, and RStudio. Try your best to separate these ideas from one another. R is a statistical programming language (proclaimed as a software environment for statistical computing and graphics), whereas R Markdown is a markup language (think of this as an environment for creating a document). RStudio, on the other hand, is the most popular **integrated development environment** (or IDE, for short) for R. RStudio, like good IDEs should, makes it easier for you to code in R by providing a user friendly landscape to write, run, and debug code.

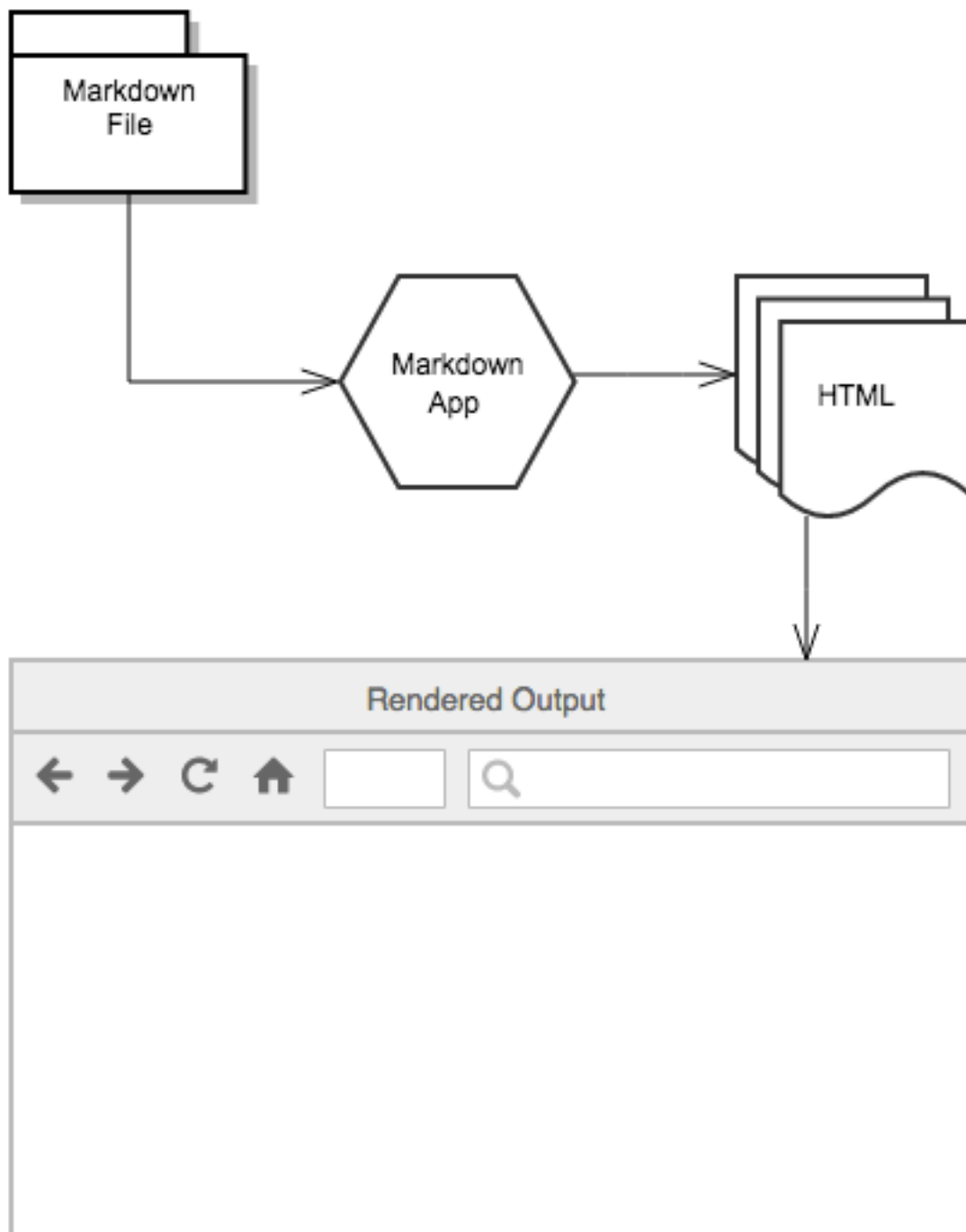


Figure 2: schmatic diagram describing how markdown is rendered

## 2.2 Creating an R Markdown Document

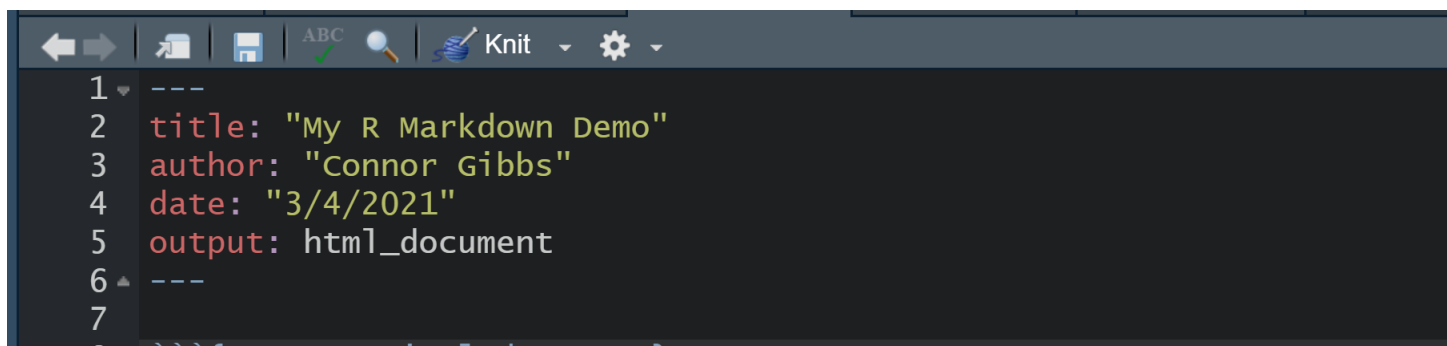
The easiest way to create an R Markdown document is to leverage RStudio by navigating to **File > New File > R Markdown**. You will be asked to populate a few basic options including the **Title**, the **Author**, and the **Default Output Format**. For now, feel free to name your R Markdown document **My R Markdown Demo** with your name as the author. Keep the default format as **HTML**. After clicking **OK**, you will be provided with a default in RStudio's text editing window. This is where you will create your document.

☀ **Bonus** If you wish to create a PDF using R Markdown, you will need a version of **TeX** on your computer, a typesetting system. For users who have never interacted with **TeX**, we recommend installing the `tinytex` package by running `install.packages('tinytex')` and then running `tinytex::install_tinytex()`. After this, you should be able to create a PDF document with R Markdown.

### 2.2.1 Basic Components

There are three basic components to an R Markdown document: the **YAML header**, **Markdown code**, and **R code chunks**. We will explore each of these components one by one.

**2.2.1.1 YAML Header** The **YAML header** exists at the very top of the R Markdown document. Following my own steps from above, the **YAML header** to my default R Markdown document looks like:

A screenshot of the RStudio text editor showing the YAML header of an R Markdown document. The editor has a dark theme. The top toolbar shows icons for navigation, saving, and the 'Knit' button. The code in the editor is as follows:

```
1 ---
2 title: "My R Markdown Demo"
3 author: "Connor Gibbs"
4 date: "3/4/2021"
5 output: html_document
6 ---
7
```

Figure 3: example of a **YAML header** for the default R Markdown template

By default, the **YAML header** must contain a **title**, **author**, **date**, and **output type**. Once the **YAML header** has been specified, you can now **knit** your document as the specified **output type** by clicking the **Knit** icon.



Figure 4: icon used to compile and format your R Markdown document

Alternatively, you can create the knitted document programmatically with `rmarkdown::render()`. Type `?rmarkdown::render` into your console for more details. Go ahead and knit your document, and take a look at the default document that is generated. The following two components are included in the default document.

**2.2.1.2 R Code Chunks** Code chunks are segments of code throughout your document that will be evaluated sequentially upon knitting. Any output stemming from your code will be included in the final document. An example of an R ‘chunk’ is included below:

```
```{r}
x <- c(1, 2, 3, 4)
```
```

All R code chunks are defined in this way; although, you may choose to specify chunk options in the `{}` of the chunk header. Chunk options allow you to customize how output of a given chunk is portrayed. Any R code can be included within the fences of three back ticks.

While incredibly useful for large blocks of code, code chunks can be an overkill for a small line of code. At times, you may wish to evaluate some R code inline, perhaps to take the sum of `x`. You can do this by

**2.2.1.3 Markdown Code** As we mentioned, markdown is essentially a language used to format a document. When creating your reports in R Markdown, all of your headers, sub-headers, paragraphs, lists, etc. will be included as Markdown code. Thankfully, most of this looks like standard English with a few minor adjustments. Anything that is not your YAML header or contained within an R code chunk (more in a second) is inherently markdown code. We will revisit this topic in future chapters when we discuss fine tuning your R Markdown document.

## References

Xie, Y., Dervieux, C., and Riederer, E. (2020). *R Markdown Cookbook*. CRC Press.