Introduction to R Markdown

STA 610 - Applied Statistics for Health Professions

Andrew DiLernia

## Learning Objectives

* Knitting R Markdown to different formats
* Understanding fundamental Markdown syntax
* Using code chunks in R Markdown
* Using code chunk options
* Including plots & tables in R Markdown

In this activity we will learn about using [R Markdown](https://rmarkdown.rstudio.com/) for reproducible data analyses and report generation. To begin, open up RStudio, which can be accessed in several different ways. The recommended approach will be to navigate to [R Studio Workbench](https://rstudio.gvsu.edu/), a cloud-based version of R Studio provided by GVSU. This will be the first step in completing these activities and other assignments throughout the semester.

## First Steps

➡️ Create a new R Markdown document with *File* > *New File* > *R Markdown…* . Knit it by clicking the  Knit *(or compile)* button (top left).

* Knit it by using the appropriate keyboard shortcut (**Mac**: *Command* + *Shift* + *K*, **Windows**: *Crtl* + *Shift* + *K*).
* Verify that you can modify the input in the YAML specifying the output file format and see the output update after reknitting the document.

➡️ Create one new R Markdown document for each of the three most-common formats: HTML, PDF and Word. Knit  each of the three documents.

* How does the output differ?
* How does the input differ?

**Note**: For PDF output you may need to install LaTeX if knitting to PDF causes an error. This is done by installing the tinytex package and then typing tinytex::install\_tinytex() into the Console and pressing return / enter.

For the rest of the activity, we will focus on the R Markdown document that is knitting to an HTML format.

### Installing and Loading Packages

Let’s install and load some R packages we will use in this activity by submitting the code below to the Console.

*# Install packages if needed (note: this may take some time to run)*  
**for**(pack **in** c('flextable', 'tidyverse')) {  
**if**(!(pack %in% installed.packages())) {  
 install.packages(pack)  
}  
}  
  
*# Load necessary packages*  
library(tidyverse)  
library(flextable)

## Markdown

Markdown is a lightweight set of conventions for formatting plain text files. It is designed to be easy to read and write, so not very customizable but can be learned quickly.

### Headings

Headings and subheadings can be included in a document to organize the document.

* # 1st Level Header
* ## 2nd Level Header
* ### 3rd Level Header

➡️ Include a first-level header, second-level header, and third-level header in your R Markdown document naming them Main Section, Subsection, and Sub-Subsection, respectively.

*# Header 1*

*## 2*

*### 3*

*#### 4*

*##### 5*

*###### 6*

*####### 7*

*######## 8*

*################## unknown*

*### try*

*\*\*Bold\*\* text.*

*### try*

*\*Italicized\* text.*

*### try*

*`code` style text.*

*### try*

*subscript~PM~*

*### try*

*superscript^AM^*

### Links and Images

Hyperlinks and external images / GIFs can be included in R Markdown documents as well.

* <https://www.gvsu.edu>
* [linked phrase](https://www.gvsu.edu)
* ![optional caption text](path/to/img.png)

➡️ Include a hyperlink in the R Markdown document linking to the Google homepage.

### Text Formatting

Basic formatting of text, such as italicizing, bolding, superscripts and subscripts can be implemented as well.

INSERT TABLE HERE

➡️ Include an italicized word or phrase, a bolded word or phrase, a word styled showing it is code, a superscript, and a subscript in your R Markdown document.

## Code Chunks

The real power of R Markdown comes from the ability to add **code chunks** to the document, which allow us to run R code inside our document:

*# Sophisticated calculation*  
pi \* 2

## [1] 6.283185

There are three main ways to insert a code chunk:

1. The keyboard shortcut *Cmd*/*Ctrl* + *Alt* + *I* (recommended)
2. The “Insert” button icon  in the editor toolbar (top right).
3. By manually typing the chunk delimiters ```{r} and ```.

➡️ Add a new code chunk at the bottom of the R Markdown document, naming the chunk uptownChunk. In the chunk, use the include\_graphics() function from the knitr package to include the image at the following URL: <https://github.com/dilernia/STA418-518/blob/main/uptownFunk.png?raw=true>

Code chunks have multiple options which control how they are displayed. Some of the most commonly used:

* eval
* echo
* include
* warning and error
* fig.width and fig.height
* fig.align

Submit str(knitr::opts\_chunk$get()) to the Console in R for a list of all default chunk options.

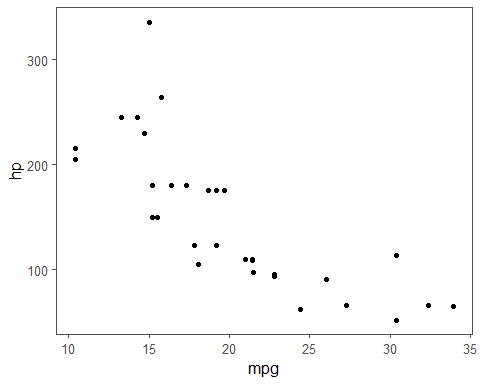
➡️ Modify uptownChunk by setting the echo chunk option to be FALSE.

Toggle the include and eval chunk options for the uptownChunk chunk and see what happens.

### Including Plots

Using R code inside chunks that produces plots allows graphics to be directly included in the output document.

*# Load the tidyverse package*  
library(tidyverse)  
  
*# Set ggplot theme for visualizations*  
theme\_set(ggthemes::theme\_few())  
  
*# Create a scatter plot*  
mtcars %>%   
 ggplot(aes(x = mpg, y = hp)) +   
 geom\_point()



➡️ Include the code to create a scatter plot above in a new chunk at the bottom of the R Markdown document.

➡️ Toggle the fig.height, fig.width, and fig.align chunk options, exploring their effect on the output.

### Inline Code

R Markdown also allows you to call R within a paragraph of text to do calculations and output values:

xbar <- 2  
se <- 1.3

The following R Markdown syntax:

The 95% confidence interval for the mean is (`r xbar - 1.96 \* se `, `r xbar + 1.96 \* se `)

produces:

The 95% confidence interval for the mean is (-0.548, 4.548).

➡️ Include a chunk of R code creating the xbar and se objects, and after that use in-line R code at the bottom of the document to reproduce the sentence containing the confidence interval results above.

### Tables

R Markdown allows you to produce nice tables that also work in Microsoft Word. There are several functions available to include tables, one of the nicer functions being the flextable() function from the flextable package:

*# Load the flextable package*  
library(flextable)  
  
*# Nicely format column names*  
my.table <- mtcars[1:5, 1:4]  
colnames(my.table) <- c("MPG", "Cylinders",   
 "Displacement", "Horsepower")  
  
*# Output the flextable*  
flextable::flextable(my.table) %>% autofit()

| MPG | Cylinders | Displacement | Horsepower |
| --- | --- | --- | --- |
| 21.0 | 6 | 160 | 110 |
| 21.0 | 6 | 160 | 110 |
| 22.8 | 4 | 108 | 93 |
| 21.4 | 6 | 258 | 110 |
| 18.7 | 8 | 360 | 175 |

### Word Templates

* Organizations, such as the [US Department of Agriculture](https://www.nass.usda.gov/Statistics_by_State/Michigan/Publications/Crop_Progress_&_Condition/2023/cw3023mi.pdf), can have weekly or monthly reports that change as data / other inputs are updated.
* R Markdown to Word can use [Word doc templates](https://bookdown.org/yihui/rmarkdown/word-document.html) for consistent formatting, headers, etc. while updating charts & tables

### Much More!

* For an R programming class I took in graduate school at Minnesota, I created an HTML document, and made it available online [here](https://rpubs.com/dilernia/HW3).
* This document was created with R Markdown.
* This [R Markdown Cheat Sheet](https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf) describes additional features and fundamentals of R Markdown.