R Markdown & Quarto

Sean Hellingman

Data Visualization and Manipulation through Scripting (ADSC1010)

shellingman@tru.ca

Fall 2024

THOMPSON RIVERS UNIVERSITY

Topics

2. Introduction
3. Creating an R Markdown Document
4. R Markdown Anatomy
5. YAML Header
6. Formatted Text
7. Code Chunk
8. R Quarto
9. Exercises and References

Introduction

- R Markdown is a way to combine R code and results with written commentary in one document
  - Very useful for writing reports or papers.
  - Technically a different language that is easily integrated into R Studio.
  - Able to facilitate the construction of reproducible reports.
- **All coding assignments in ADSC1010 & ADSC1000 will presented as Markdown files**

Initialization

- You will need to install the markdown R package.
  - `install.packages("rmarkdown", dep = TRUE)`
- If we want to generate PDF outputs (we will) we need to install LaTeX.
  - `install.packages("tinytex")`
  - `tinytex::install_tinytex()`
- Once installed we are now able to compile most R Markdown documents.

New R Markdown

- To create a new R Markdown file

  File ⇒ New File ⇒ R Markdown

  - In the pop up window, give the document a *Title*, add an *Author*, and set PDF as the *Default Output Format*.
  - PDF files are smaller and will be easier to submit assignments with.

- The new R Markdown document will contain a header and some example code.
  - It's a good idea to periodically save your file as you work.

Example 1

- Create a new R Markdown file.
- Click *knit* to practice converting your file to a PDF.
- *Notice there is a new R Markdown tab in your console window*.

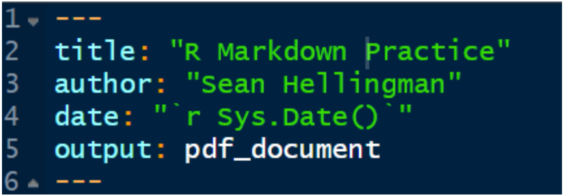
Anatomy

- R Markdown files are generally composed of three components:

1. YAML header
2. Formatted text
3. Code chunk

YAML Header

- YAML stands for ‘YAML Ain’t Markup Language’
  - Automatically created when you create a new markdown file through R studio.
  - Contains the metadata and options for the entire document such as the author name, date, output format, etc.



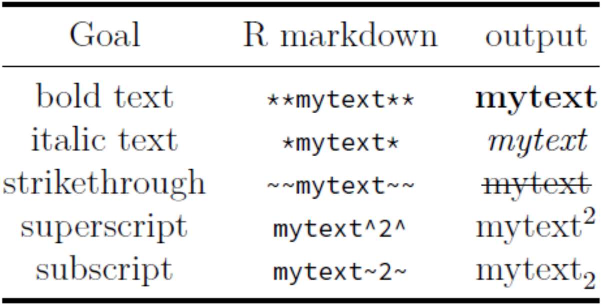
```
---
title: "R Markdown Practice"
author: "Sean Hellingman"
date: "`r Sys.Date()`"
output: pdf_document
---
```

- Many header options: https://bookdown.org/yihui/rmarkdown/html-document.html

Formatted Text

- The advantage of R markdown is that one can add formatted text to generate reports.
- The text format is **NOT** *What-You-See-Is-What-You-Get*
    - R markdown is not like MS Word, the user needs to 'markup' the text formatting.
    - May seem difficult at first (we will try to keep things simple at first).
    - Makes formatting much easier than using R and MS Word together.

Formatted Text Emphasis



Goal	R markdown	output
bold text	`**mytext**`	**mytext**
italic text	`*mytext*`	*mytext*
strikethrough	`~~mytext~~`	~~mytext~~
superscript	`mytext^2^`	mytext^2
subscript	`mytext~2~`	mytext_2

Figure: *source: (1)*

White Space and Line Breaks

- Multiple spaces and carriage returns (enter keys) are generally ignored.
- To start text on a new line then add two blank spaces at the end of the preceding line.
- Can use the `&nbsp;` command to add multiple spaces to text.

Headings

- Headings and subheadings are added to the formatted text by using #.
- Increasing the number of #s decreases the size of the heading.
- Remember to put a space between the # and your heading.

Comments

- The # symbol is used for heading in the formatted text
    - # is still used to comment in a *code chunk*.
- Enclose a comment between `<!--` and `-->`
- Comments in the formatted text will not be included in the final rendered document.

Lists I

- To add a *bullet point* list to the formatted text use the - and + symbols accordingly:

```
- item 1
- item 2
     + sub-item 2
     + sub-item 3
- item 3
- item 4
```

Lists II

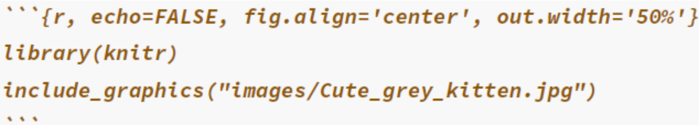
- To add an *ordered* list to the formatted text use the - and + symbols accordingly:

```
1.  item 1
2.  item 2
     + sub-item 2
     + sub-item 3
3.  item 3
4.  item 4
```

Images

- There are two primary ways to add images to your R Markdown document:

1. Including the image in the formatted text section.
    - Example code: `![Cute grey kitten](images/Cute_grey_kitten.jpg)`
2. Within the *code chunk* (allows for easier formatting)
    - Example code:



~~~
```{r, echo=FALSE, fig.align='center', out.width='50%'}
library(knitr)
include_graphics("images/Cute_grey_kitten.jpg")
```
~~~

Figure: *source: (1)*

Links

- Clickable links are easily included in formatted text.
- Code: `[link name](https://www.tru.ca/)`

Example 2

- Create a new R Markdown file with your name and student number in the YAML Header.

- In the formatted text add the following things:
  - A header and sub header for *Thompson Rivers University & Post-Baccalaureate Diploma in Applied Data Science*.
  - The following sentence formatted the same way: *My name is* **your name** not ~~Einstein~~, but I know $E = MC^2$.
  - A list of topics that you are looking forward to learning.
  - A link called *TRU Homepage* to https://www.tru.ca/

- Knit your document to a PDF.

Code Chunks

- To include R code into your R markdown document you place your code into a **code chunk**.

- **All code chunks begin and end with three backticks ```**
  - The backtick key is usually found on the same key as tilde ($\sim$).
  - Windows shortcut: Ctrl + Alt + I
  - MacOSX shortcut: Cmd + Option + I

~~~
```{r}
Any valid R code must go between the backticks (```)
```
~~~

Code Chunk Arguments

- You can use your code chunks to run analysis, generate outputs and tables, and reformat data.

- You can place rules and arguments between the curly brackets {}
  - You can label the code chunk: ` ```{r label} `
  - Display only the output and not the code: ` ```{r label, echo=FALSE } `
  - Display the R code but not the output: ` ```{r label, results="hide" } `

- Please see the link in the *References & Resources* slide for more code chunk arguments.

Arguments for Figures

- When generating reports it can be useful to adjust the figures included in the report.
- To adjust the figure dimensions use `fig.width=` and `fig.height=` (in inches).
- Can adjust the alignment of the figure using `fig.align=`
- You can also add a caption to the figure using `fig.cap="Name of Figure"`

Example 3

- In the R markdown file you have already created, do the following:
  - Initialize a vector called Vector.1, that goes from 1 to 10.
  - Initialize a vector called Vector.2 that goes from 50 to 5 by 5.
  - In a separate code chunk, plot Vector.1 by Vector.2.
  - In a separate code chunk, plot Vector.1 by Vector.2 but adjust the size to be 3 inches high and 4 inches wide.
  - Add a caption to the smaller figure.
- Knit your document to a PDF.

Adding Tables

- Adding results tables or data summaries can be very important to writing reports in data science.
- Tables can be generated using standard markdown context (tedious).
- Use the `kable()` function from the *knitr* package to generate tables.
  - Call the `kable()` function inside code chunks.
  - Can also use the *kableExtra* package.
- To see all possible arguments use `?knitr::kable`
- Or, use: https://bookdown.org/yihui/rmarkdown-cookbook/kable.html

Example 4

- In the R markdown file you have already created, do the following:
- Use `data("iris")` to load the *iris* dataset in base R.
- Create a table using `kable()` of the first 10 observations from this data.
- Rename the columns to something more suitable.
- Knit your document to a PDF.

R Quarto

- *Quarto is a multi-language, next generation version of R Markdown from RStudio*
- It is slightly easier to use than Markdown.
- If you are using the latest version of R and have Markdown installed, Quarto should work.
- You create a new Quarto document the same way as an R markdown script.

Example 5

- Create a new R Quarto script.
- Create a new list of your TRU courses in the formatted text.
- Create a new code chunk to perform some basic calculations.
- Take some time to explore the functionality of the interface.

Final Thoughts

- R Markdown and/or R Quarto will be used for your Assignments in this course and in ADSC 1000.
- It may also be very useful for future projects as you get comfortable with the functionality.

Exercise 1

- In R Studio, create a new R Markdown file.
- In the pop up window:
  - Change the *Title* to "R Markdown Exercises".
  - Change the *Author* to your name and student number.
  - Set PDF as the *Default Output Format*.
  - Click "Okay" to create your new document.
  - Delete all of the irrelevant example code.

Exercise 2

- In the same R Markdown file as Exercise 1 in the formatted text section include:
  - A header and sub header for your previous university and program.
  - Three sentences about your previous education with some *italic*, **bold**, and ~~srtuck through~~ words.
  - A list of previous topics you learned at your last university.
  - An embedded link to the homepage of your last university.

Exercise 3

- In the same R Markdown file as Exercise 1 create a new code chunk and:
  - Use `rep` to create each of the following vectors.
    $(1, 0, -1, 1, 0, -1, 1, 0, -1)$.
    $(2, 0, 2, 0, 2, 0, 2, 0, 2, 0)$.
  - Use `c` and `rep` to create each of the following:
    $(1, 1, 1, 1, 3, 3, 3, 3, 3, 3)$.
    $(1, 1, 1, 7, 2, 2, 2, 2, 2, 3)$.

Exercise 4

- In the same R Markdown file as Exercise 1 create a new code chunk and:
    - Use `data("iris")` to load the *iris* dataset.
    - Use a combination of R markdown commands and `plot` functions to appropriately plot one or more of the variables in *iris*.

Exercise 5

- In the same R Markdown file as Exercise 1 create a new code chunk and:
  - Use the `kable()` function to create a table for the *iris* data.
  - Take some time to play around with the different arguments found in `kable()`.

Exercise 6

- *knit* your R markdown document to a PDF.
- How does it look?

Exercise 7

- Use Quarto to repeat Exercises 1 - 6.
- Does Quarto seem easier to navigate?

References & Resources

1. Douglas, A., Roos, D., Mancini, F., Couto, A., & Lusseau, D. (2023). *An introduction to R*. Retrieved from https://intro2r.com/

- Markdown Cheat Sheet
- R Markdown
- R Quarto