R Markdown Template

STAE04: Data Visualization

Johan Larsson

2020-08-28

Contents

1 R Markdown

This document is written using R Markdown. R Markdown is a syntax for formatting documents that lets you focus on the content instead of formatting. You write text, including code, in R Markdown syntax and then R Markdown (plus a handful of very useful packages) turn that text into a neatly formatted document.

This document serves as both a template for your submissions during the data visualization course as well as an introduction into R Markdown. To follow along, you should load this file in RStudio.

1.1 Installation

To get started, you are going to need two packages: **rmarkdown** and **knitr**.

```r
install.packages(c("rmarkdown", "knitr"))
```

You will also need a distribution of LaTeX. Installing LaTeX can be a little involved, but it is fortunately very easy to install a lightweight LaTeX distribution using the **tinytex** package (if you don't already have a distribution), like so:

```r
install.packages("tinytex")
tinytex::install_tinytex()
```

1.2 YAML

A R Markdown file ends with `.Rmd` and is best edited using RStudio. Each Rmarkdown file starts with a so called YAML block; here is a bare-bones one:

```yaml
---
title: "An Awesome Title"
author: "Fantastic Me"
date: "2020-09-28"
output: pdf_document
---
```

For this course, please use the YAML block supplied in this template but change `author`, `date`, and `title` fields. You can also remove the `references` item in the YAML block.

1.3 Formatting

Rmarkdown uses Pandoc markdown for formatting text, which uses a special—but very simple—syntax for formatting text. The following are some ways in which you can format text in R Markdown.

Start a new paragraph by surrounding text with blank lines.

Single asterisks italicize text, **double asterisks format text in bold**, and `grave accents formats text in monospace`.

To create lists in markdown, you add a

- dash before each item in the list, and
  - possibly asterisks for sub-items in the list,
- placing items each on its own line, and
- adding blank lines before and after the list.

Numbered lists are

1. created similarly, but
2. using numbers instead of dashes.

Sections are created by prefacing the section title with a hash tag (`#`):

```
# One Hashtag Creates a Section

## Two Hashtags Creates a Subsection

### Three Hastags Creates a Subsubsection
```

R Markdown can do quotation formatting too!

> If you want to quote something, adding a `>` before the text creates a block quote —Johan Larsson

There are many ways to format tables in markdown, but the simplest one is two simply create columns of text with dashes (`---`) separating the title of each column from the cells of the table.

Table 1: A caption for the table can be added like this.

Header 1	Header 2
Cell 1	Cell 2
Cell 3	Cell 4

To add a link in markdown, you can either simply surround the URL with brackets like this link to a R Markdown cheatsheet: https://rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf or you can use brackets and parantheses to provide your own text for the link.

Images can be added with syntax similar to the one for links, with the text inside brackets indicating the caption for the figure. Let's first download an image to our working directory.

```
download.file(
  "https://upload.wikimedia.org/wikipedia/commons/2/29/Minard.png",
  "minard.png"
)
```
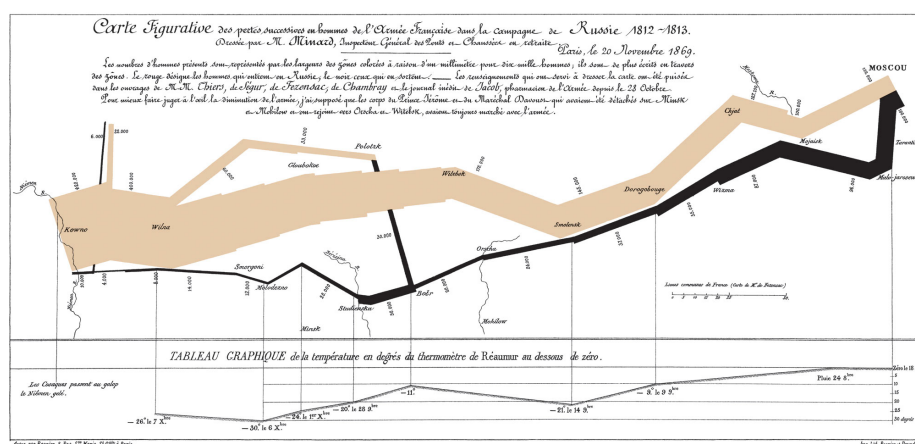

Figure 1: Here is Charle's Minard famous Napoleon chart, for instance. (Public domain.)

Footnotes can be useful to provide additional information[1]. For a longer footnote, it might be better to refer to it with a number[2].

It is also possible to add citations in R Markdown but this is somewhat complicated if you are not familiar with markdown, pandoc, or LaTeX since before. In this course you will usually never need more than one or two citations, so we

---

[1] If the footnote is short, it is often best to write it in-line, like this.

[2] Then you can add your text separately in the document.

suggest you write the citations manually, which means you can skip the next paragraph (unless you are keen on getting deeper into R Markdown and pandoc).

To cite in R Markdown, you will need either 1) a `.bib` file (with bibtex-formatted references) somewhere in your working directory or 2) a `references` field in the YAML block; in this document we use the latter but only to keep this document self-contained. Using a `.bib` file is recommended. Find the key of the reference you are looking for and preface it with an `@`, like the following reference to Wickham (2010), which creates a text reference. If you surround the key with brackets, you instead get a non-text citation (Wickham 2010). If you've done everything right, the final document will get a bibliography at the end (as in this one).

1.4 Code Chunks

So far we've only really talked about features that are made possible by pandoc and its flavor of markdown. But what makes R Markdown special is that is allows us to write code chunks containing R code. This is a code chunk that plots a simple plot using ggplot2:

```r
library(tidyverse)

ggplot(msleep, aes(brainwt, sleep_rem)) +
  geom_point() +
  scale_x_log10() +
  labs(x = "Brain Weight (lbs)", y = "REM Sleep (hours)")
```
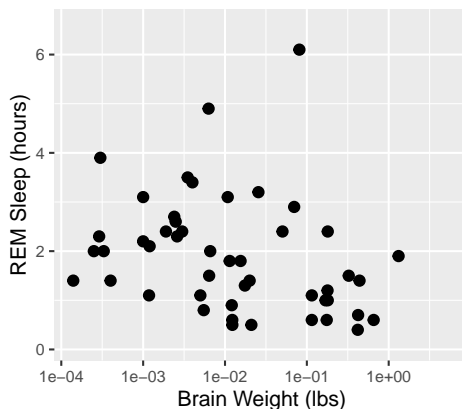

As you can see, we've started the code chunk with ` ```{r} ` and ended it with ` ``` `. Everything in between will be treated as R code, just as if you would have written in in an R script of the R terminal. When you compile this document all this code will be run and if it produces any output (text, plot, tables) then that output will make it into the final document.

You can control many settings by changing the parameters in the headline of the code chunk. In the following chunk we've changed the width and height of the figure as well as added a caption to the figure. These are settings that will be **incredibly** useful to you during the course.

```r
ggplot(msleep, aes(brainwt, sleep_rem)) +
  geom_point() +
  scale_x_log10() +
  labs(x = "Brain Weight (kg)", y = "REM Sleep (hours)")
```
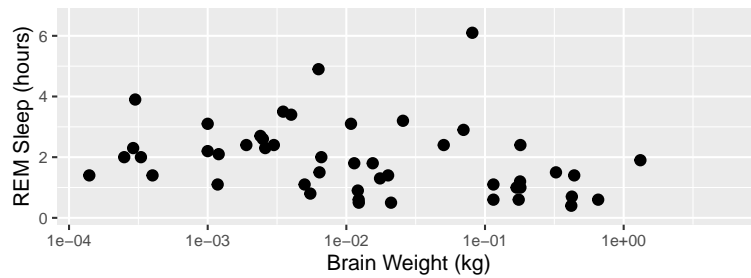

Figure 2: Brain weight and REM sleep duration for mammals.

1.5 Knitting

When you are done writing content, you can render the final document by *knitting* it. In RStudio, the shortcut for this is `CTRL+Shift+K`. When a document is knitted, all your code in the document is first run and then the result is passed on to pandoc, which produces the final document for you.

R Markdown files can be knitted to a wide range of medium, including PDF files, HTML, and word documents. Although in this course you will always produce PDF files from your R Markdown documents.

1.6 Reproducibility

Authoring your documents using R Markdown facilitates reproducibility. Because you need to supply all the code used to produce your paper in the `.Rmd` file, this makes it much easier for other people to re-run your analysis and use your code.

2 Submissions

In this course, you will use R Markdown for all your assignment submissions. For each of these you will need to submit both the `.Rmd` source file as well as the knitted `.pdf` file. Documents from office packages (such as Microsoft Office) will not be accepted.

3 Learning More About R Markdown

If you want to learn more about R Markdown, we recommend the R Markdown Cookbook. If you run into any issues with R Markdown, please use the course's discussion board on Canvas or search stack overflow with the `[r-markdown]` or `[knitr]` tag.

References

Wickham, Hadley. 2010. "A Layered Grammar of Graphics." *Journal of Computational and Graphical Statistics* 19 (1): 3–28. https://doi.org/10.1198/jcgs.2009.07098.