

Quarto and RMarkdown Statistics 506

Quarto and RMarkdown

Both Quarto and RMarkdown are tools which convert documents (.qmd and .Rmd respectively) which contain a mix of text and R code into other types of documents, such as web pages (html), PDF, or Word documents. By placing the code in special **code chunks**, these tools will execute the code and place the results in the documents as well.

(All course material is written in Quarto.)

Quarto vs RMarkdown

RMarkdown has been the standard for years. Quarto is a newer version that removes the dependence on R. RMarkdown supports other languages such as python and Julia, but still required R. Quarto supports these languages without R installed.

To read more about the shift from RMarkdown to Quarto:

- [Quarto: FAQ for R users](#)
- [With Quarto Coming, is R Markdown Going Away? No.](#)

Quarto can process RMarkdown documents in almost all cases and the two are essentially interchangeable. However, the future is clearly Quarto, so we will be covering it in this class.

Markdown

[Markdown](#) is a plain text formatting language designed to create html documents using plain text and indentation to specify the desired format while remaining easy to read in its raw format. Thanks to the magic of [Pandoc](#) Markdown can also be used to create PDF, MS Word, and other types of documents. There are many “flavors” of markdown, but they all implement the core idea of using simple plain text formats to specify the desired markup.

Some commonly used markups include **bold** or *italics* via double ** or single * surrounding the text. Sections titles are prefaced with #, with an increasing number of #'s to indicate

subsections. Lists can be created ordered (“1.” or “a.”) or unordered(“-”). Tables are both easy and frustrating to make, I recommend using an online tool to generate the markdown syntax for a table.

There are a ton of resources online for basic Markdown guides; I like this site for a quick reference of the basics: <https://www.markdownguide.org/cheat-sheet/>.

Note that there are different **flavors** of markdown, of which RMarkdown may not be fully compatible. If you find something online suggesting how to accomplish something in markdown, make sure its in markdown or RMarkdown and not another flavor.

Code chunks

A Quarto document contains any text you want, with **code chunks** such as the following:

```
```{r}
data(mtcars)
m <- mean(mtcars$wt)
m
```
```

After processing the file, that chunk produces:

```
data(mtcars)
m <- mean(mtcars$wt)
m
```

```
[1] 3.21725
```

The R session in the chunks are persistent, so we can refer to objects created in earlier chunks:

```
z <- m/sd(mtcars$wt)
z
```

```
[1] 3.288084
```

Chunk options

You can pass options to code chunks in the form of special comments. For example,

```

```{r}
#| echo: false
y <- rnorm(10)
z <- rnorm(100)
mean(y) + mean(z)
```

```

produces only:

```
[1] -0.1568799
```

because the `#| echo: false` option suppressed echoing the code out.

The available chunk options are discussed [here](#) and [here](#).

Equations

Equations can be entered via LaTeX syntax:

```

\begin{align}
{n \choose k} &= \frac{n!}{(n-k)!k!} \\
&= \frac{\prod_{i=1}^n i}{\left(\prod_{i=1}^{n-k} i\right) \left(\prod_{j=1}^k j\right)} \\
&= \frac{\prod_{i=(n-k+1)}^n i}{\prod_{j=1}^k j} \\
&= \frac{(n-k+1) \times \dots \times n}{2 \times \dots \times k}
\end{align}

```

produces

$$\binom{n}{k} = \frac{n!}{(n-k)!k!} \tag{1}$$

$$= \frac{\prod_{i=1}^n i}{\left(\prod_{i=1}^{n-k} i\right) \left(\prod_{j=1}^k j\right)} \tag{2}$$

$$= \frac{\prod_{i=(n-k+1)}^n i}{\prod_{j=1}^k j} \tag{3}$$

$$= \frac{(n-k+1) \times \dots \times n}{2 \times \dots \times k} \tag{4}$$

A few references for learning the LaTeX equation syntax:

- <https://en.wikibooks.org/wiki/LaTeX/Mathematics>
- <http://tug.ctan.org/info/undergradmath/undergradmath.pdf>
- <https://math.meta.stackexchange.com/questions/5020/mathjax-basic-tutorial-and-quick-reference> (MathJax is the tool that displays LaTeX equations in web browser)

Note: In most situations, $\$$ $/$$$ $$$$ and $\backslash(\backslash)/\lbracket \rbracket$ are interchangeable, but not always. If one fails, try the other. The dollar signs tend to work better in Markdown, whereas the brackets tend to work better in raw LaTeX.

YAML Header

Adding a YAML header to a Quarto file provides metadata that the processing engine can use. For example, the header to this very file is

```
---
title: "Quarto and RMarkdown<br /><a href='./index.html'>Statistics 506</a>"
format:
  html:
    css: style.css
    toc: true
    toc-depth: 5
    toc-expand: 2
---
```

This is written in a language called [YAML](#). It consists of a series of metadata key-value pairs, using indentation to denote the structure, similar to python.

In the header, I specify the title of the document, the format of the document, then give some options to that format.

You can find a list of options for [html documents](#) or [pdf documents](#). Other documents may have documentation as well.

Rendering Quarto documents

Opening a Quarto file (.qmd) in RStudio allows you to hit the “Render” button to convert it into the requested format.

You can also use the Quarto [package](#) to render documents. This may be handy in scripts.

If you aren’t writing an R document, you’ll likely want to switch to running `quarto` from the [command line](#).