Quarto Basics

What do we want to be able to do?

Data Science!

- Read in raw data and manipulate it
- Combine data sources
- Summarize data to glean insights
- Apply common analysis methods
- Communicate Effectively

This entire process must be reproducible! Git and github certainly help but we also need to make sure that we document our thoughts and process as we work through cleaning our data, summarizing it, and fitting our models.

There are two major tools to enhance how we program that work very well with RStudio:

- R Markdown:
    - Allows for writing of thought processes, code, and interpretations
    - Easy to create many types of final documents: HTML pages, PDFs, slideshows, and more!
    - Created via a `.Rmd` (R markdown file)
- Quarto: Next generation version of R Markdown!
    - Built to use multiple programming languages (R, python, and Julia) easily
    - Works with Jupyter Notebook format as well
    - Created via a `.qmd` (quarto markdown file)
    - Renders most .Rmd files as is!

We'll go through the basics to get you started. Much more is available on the quarto docs page, the RStudio quarto integration page, and in the R for Data Science book.

Markdown Idea

Markdown is a simpler version of a markup language. HTML is the most commonly known markup language (HTML = Hypertext markup language). With HTML you use tags to specify things that a web browser like chrome interprets. For instance,

```
<h1>My first level header</h1>
<a href = "https://www.google.com">Link to a search engine.</a>
```

is HTML interpreted by your browser to be a header and a link. Markdown is a simpler version of this. There are many markdown languages (include quarto markdown, R markdown, and github markdown) but most have the same base

structure. An equivalent way to write the above would be

```
# My first level header
[Link to a search engine](https://www.google.com)
```

Where R markdown and quarto go beyond is in the ability to weave R code into the equation!

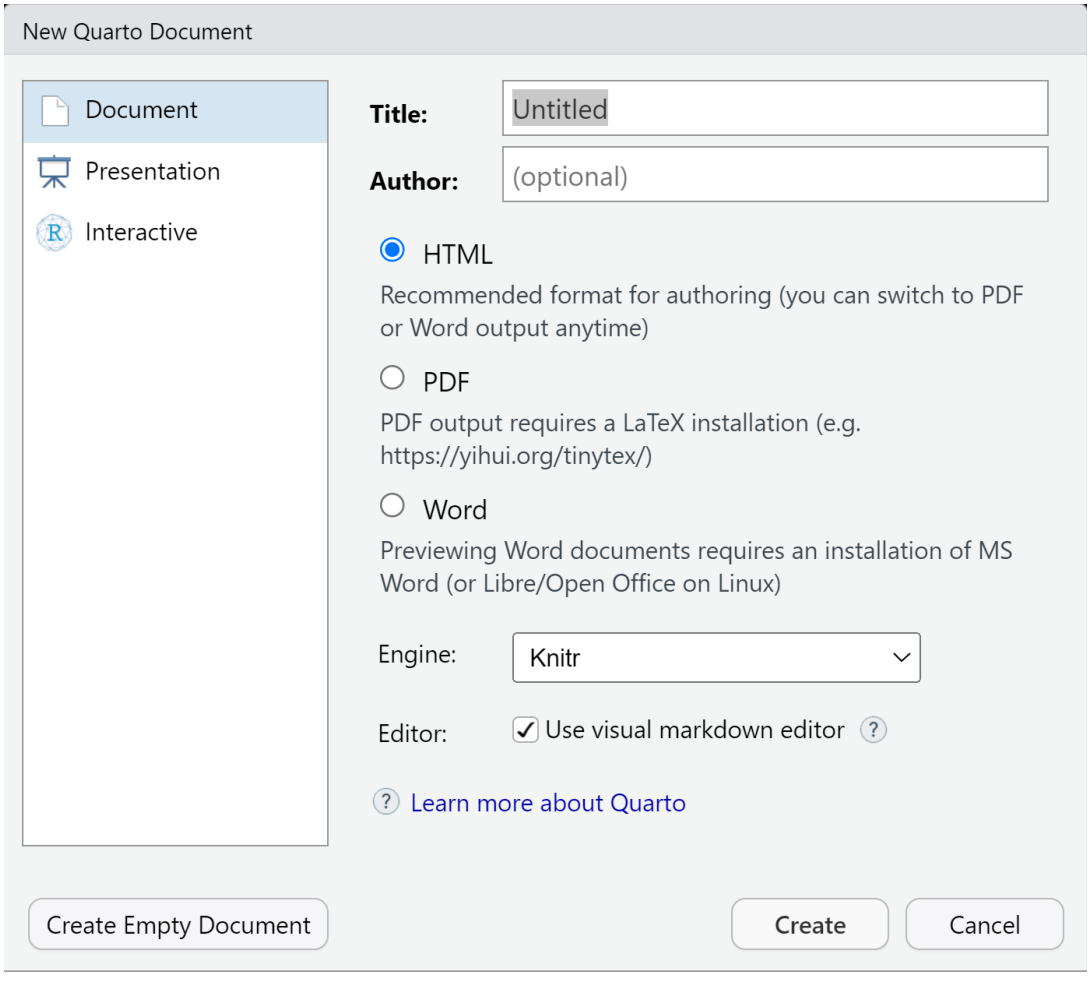
- You can include **code chunks** in your markdown.
- You then render the markdown through RStudio (or the command line).
- The R code runs and output is included in the final document!
- It is very awesome.

Documenting with Markdown via quarto

Designed to be used in three ways (R for Data Science)

- Communicating to decision makers (focus on conclusions not code)
- Collaborating with other data scientists (including future you!)
- As environment to do data science (documents what you did and what you were thinking)

It is easy to create many types of documents in quarto! Go to file –> New File –> Quarto Document



Select HTML as that is the easiest to render (build as the output). Save the template file that is produced (this will be a `.qmd` file).

`.qmd` files contain three important types of content:

1. (Optional) YAML header surrounded by `---`s
    - Defines meta data about the document
2. Chunks of R code
    - Code that may evaluate and produce output when *knitting* the document
3. Text mixed with simple text formatting instructions (Markdown syntax)

There is a visual editor in RStudio (similar to a word processing program) and a source editor. I'd recommend starting with the visual editor and trying to move to the source quickly. You'll be way more efficient using the source editor.

YAML Header

YAML stands for "Yet Another Markup Language" or "YAML ain't markup language". This defines settings for the creation process (when you go to render the document).

As we used an HTML for our document, you should see something similar to this in the top part of your document:

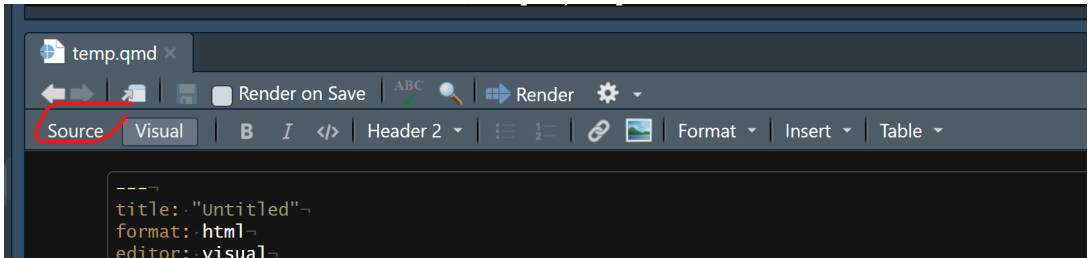
```
---
title: "Untitled"
format: html
editor: visual
---
```

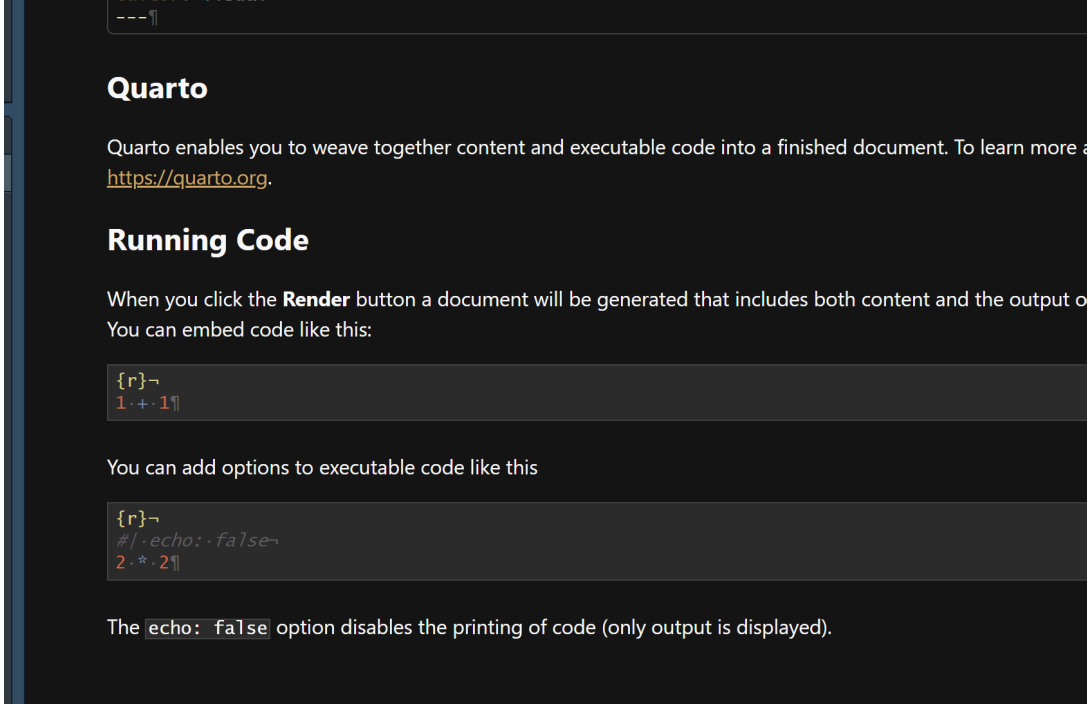
If you render the document (which I'll likely call Knitting as that is what it was called with R Markdown), it obeys these instructions for what to create.

- Try to render the document and see if you can get the HTML output.
    - Do this via the "Render" button or by using hot keys: CTRL/CMD + Shift + k

Markdown Syntax

In the template document created you'll also see some text. Some of it is larger, some of it links, some of it plain text. If you click on the "Source" you'll see the markdown syntax that you can use to spice up your outputted document:





The syntax is really easy to pick up. Below you'll find some commonly used markdown syntax:

- `# R Markdown` → First level header
- `## Next` → Second level header
- `**Knit**` or `__Knit__` → Bold font (**Knit**)
- `*italic*` or `_italic_` → Italic font (*italic*)
- `*__both__*` → Bold and italic (***both***)
- `<https://rstudio.github.io/cheatsheets/quarto.pdf>` → A hyperlink: https://rstudio.github.io/cheatsheets/quarto.pdf
- `[Cheat Sheet link](https://rstudio.github.io/cheatsheets/quarto.pdf)` → Cheat Sheet link

Check this site for markdown basics. (Headers can be used to easily create a table of contents (and is useful for accessibility of documents).)

Code Chunks

The real power of quarto and R markdown is the ability to run R code when rendering and having the output show in the final document. This saves so much time and makes updating reports/documents a breeze!

A code chunk looks like the following:

````
```{r}
#| echo: false
#| fig-cap: "Air Quality"

library(ggplot2)
ggplot(airquality, aes(Temp, Ozone)) +
  geom_point() +
````
````

```
  geom_smooth(method = "loess", se = FALSE)
```

- Start a code chunk by typing out the syntax or with CTRL/CMD + Alt/Option + I
- When rendering:
  - Chunks run sequentially in the document
  - Chunks share objects. Essentially an R environment is created when rendering a document and all objects created in chunks are stored in it.
  - Can specify behavior of each code chunk (show R code or hide it, evaluate or don't evaluate) and set global chunk behavior

To change the behavior of code chunks, we use chunk options:

- Hide/show code with `echo = FALSE/TRUE`
- Choose if code is evaluated with `eval = TRUE/FALSE`
- Have code evaluate, not show, and show no output with `include = TRUE/FALSE`
- Turn on/off displaying of messages or warnings with `message = TRUE/FALSE` and `warning = TRUE/FALSE`

Specifying these options in the top part of the code chunk is the R Markdown way of doing things (still acceptable)

```
```{r ggplot,eval=FALSE}
select(iris, Sepal.Width)
ggplot(iris, aes(x = Sepal.Width, y = Sepal.Length)) +
geom_point()
```
```

However, the new, better, way of doing it is via special comments.

> One important difference between R Markdown documents and Quarto documents is that in Quarto chunk options are typically included in special comments at the top of code chunks rather than within the line that begins the chunk. For example:

```
```{r}
#| echo: false
#| fig-cap: "Air Quality"

library(ggplot2)
ggplot(airquality, aes(Temp, Ozone)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE)
```
```

With quarto, if you want to specify global chunk options the best way to do so is in the YAML header. **Be very careful about spacing in YAML headers!**

Here is an example that would make all code chunks be 'collapsed' by default.

```
---
title: "My Document"
format: html
knitr:
  opts_chunk:
    collapse: true
---
```

You can also still set code chunk options in a *setup* code chunk. This is just a code chunk you put at the beginning of the document that sets options for you. Something like the following in a code chunk:

```
#| include: false
knitr::opts_chunk$set(echo = FALSE, eval = TRUE, warning = FALSE)
```

The `include: false` tells `knitr` not to include the code or output of this chunk in the final document.

Please pop this video out and watch it in the full panopto player!