

1.5 – Optimize Workflow

ECON 480 • Econometrics • Fall 2022

Dr. Ryan Safner

Associate Professor of Economics

safner@hood.edu

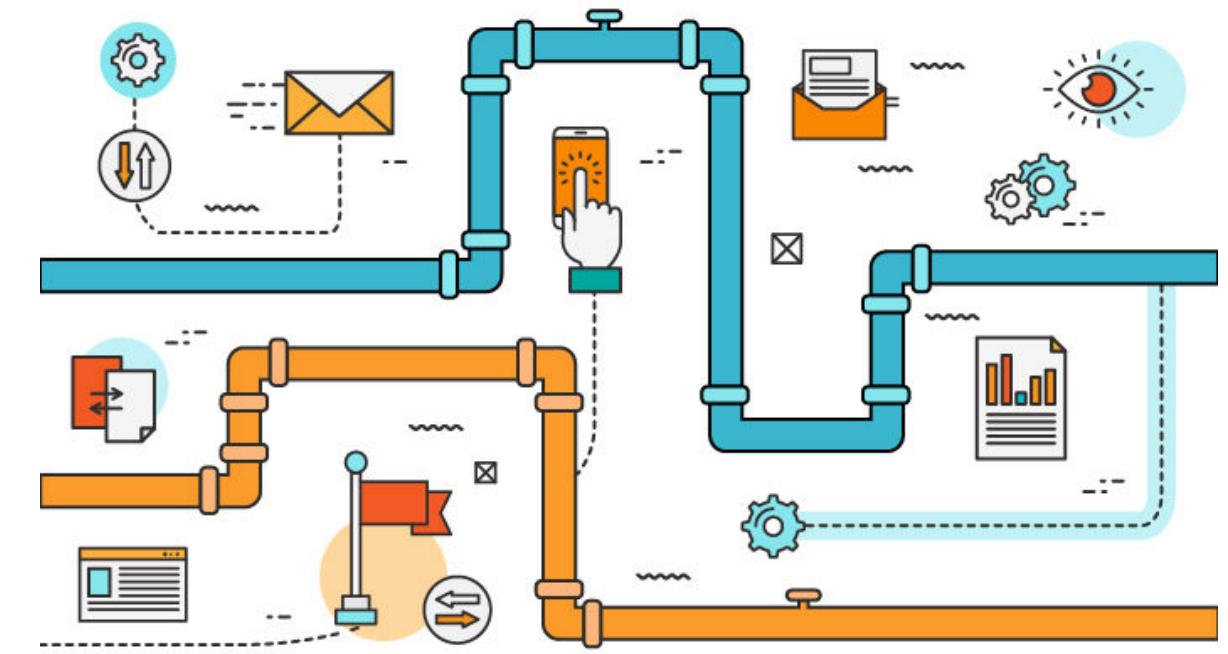
ryansafner/metricsF22

metricsF22.classes.ryansafner.com



Your Workflow Has a Lot of Moving Parts

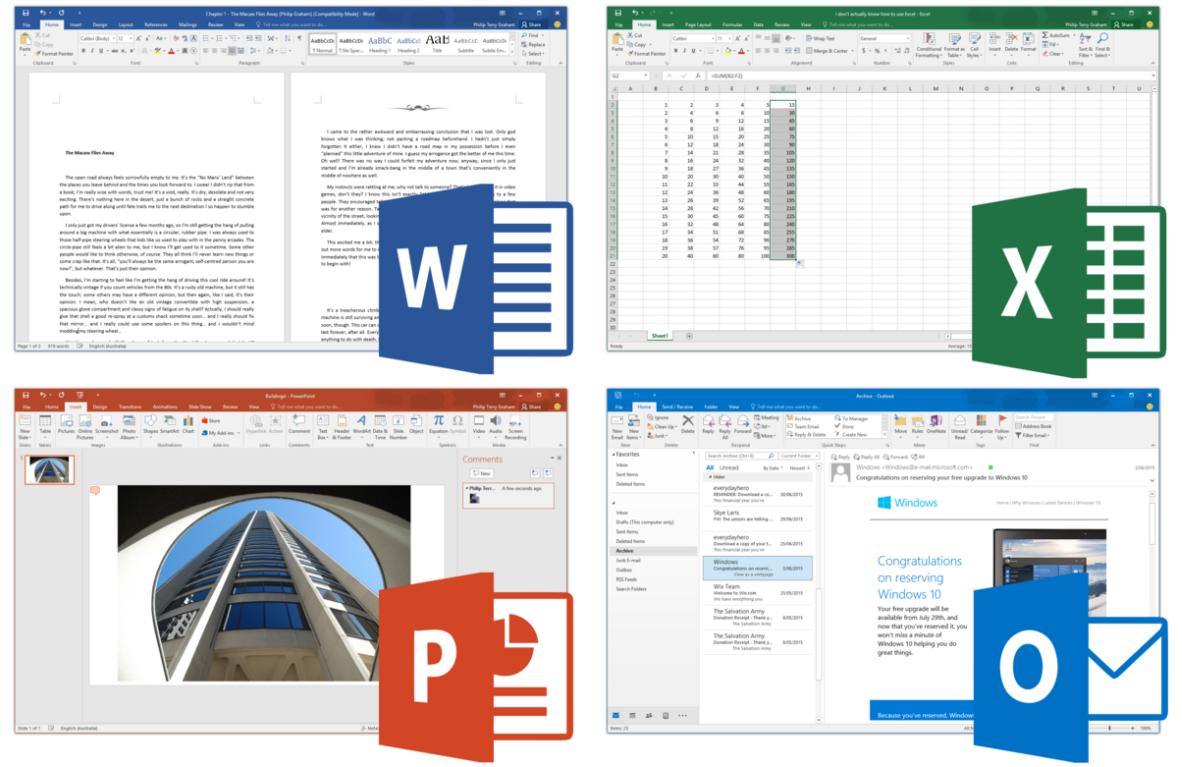
1. Writing text/documents
2. Managing citations and bibliographies
3. Performing data analysis
4. Making figures and tables
5. Saving files for future use
6. Monitoring changes in documents
7. Collaborating and sharing with others
8. Combining into a deliverable (report, paper, presentation, etc.)



The Office Model

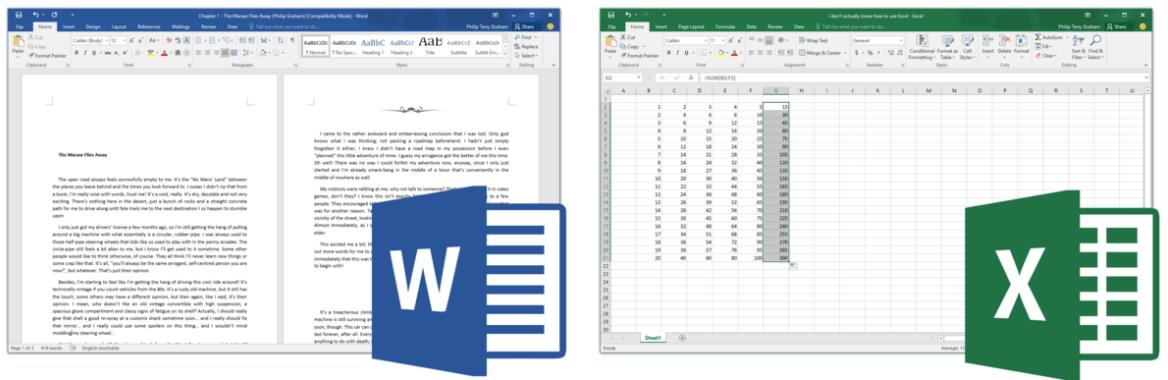
The Office Model I

1. Writing text/documents
2. Managing citations and bibliographies
3. Performing data analysis
4. Making figures and tables
5. Saving files for future use
6. Monitoring changes in documents
7. Collaborating and sharing with others
8. Combining into a deliverable (report, paper, presentation, etc.)

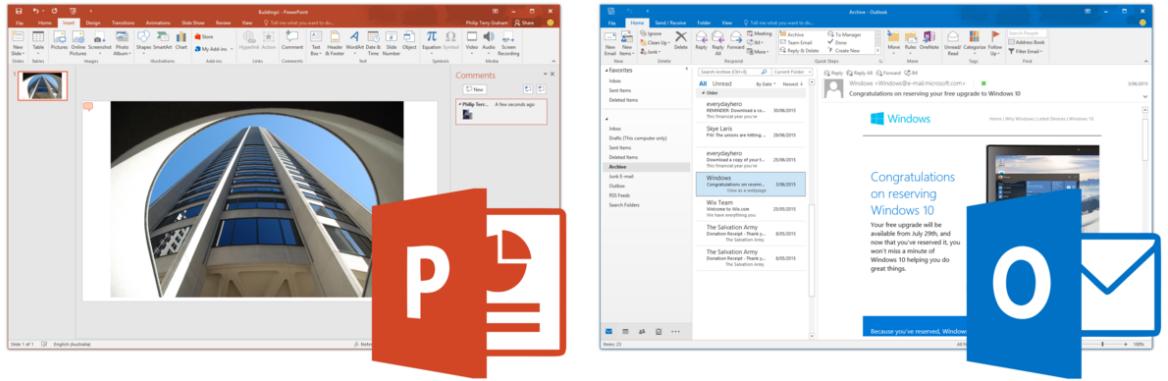
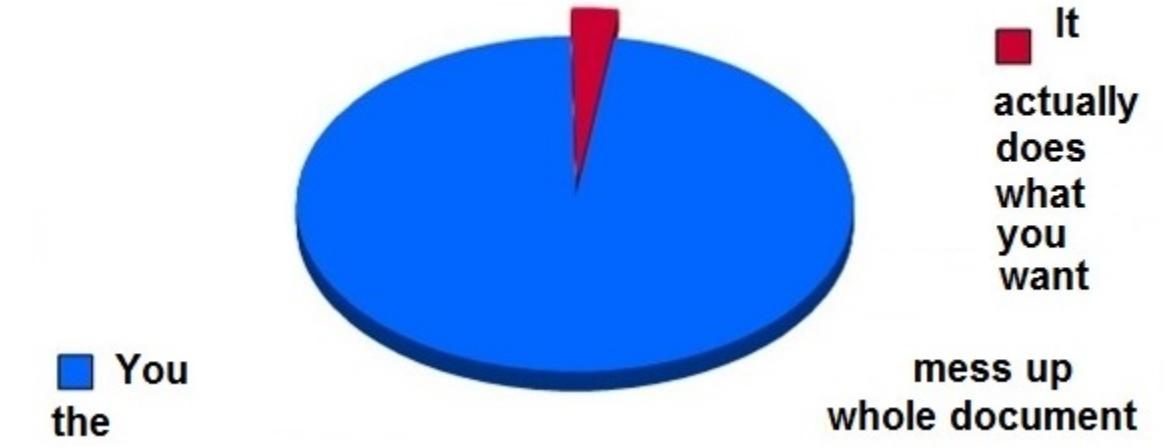


The Office Model II

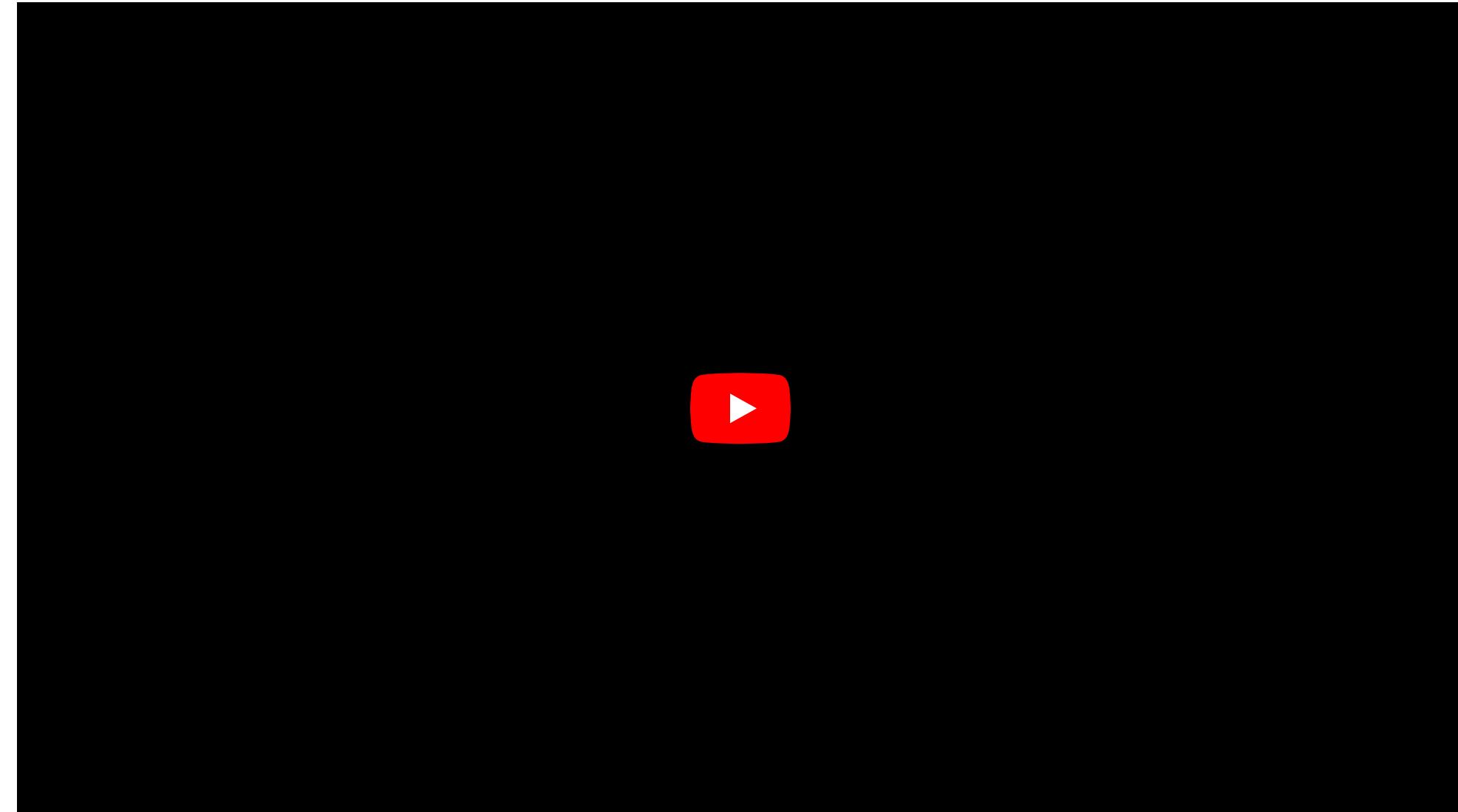
- A lot of **copy/paste**
- A lot of:



Moving a picture in Microsoft Word



The Office Model: A Short Horror Film



The Office Model: Mistakes

Science

Contents News Careers Journals

PLOS ONE PHYLOGENY/Flickr (CC BY 2.0)

Subscribe today. If you have already subscribed, log into your [News account](#).

Got a tip? [How to contact the news team](#)

Latest News

Trending Most Read

1. Crystalline nets harvest water from desert air, turn carbon dioxide into liquid fuel
2. This bird really can fly over Mount Everest, wind tunnel experiments reveal
3. Stalled in Hawaii, giant telescope faces roadblocks at its backup site in the Canary Islands
4. Study challenges idea that autism is caused by an overly masculine brain
5. Genetics may explain up to 25% of same-sex behavior, giant analysis reveals

Related Jobs

Engineer I / II, Sterile Product Technology
Moderna, Norwood, MA

One in five genetics papers contains errors thanks to Microsoft Excel

By Jessica Boddy | Aug. 29, 2016, 1:45 PM

Autoformatting in Microsoft Excel has caused many a headache—but now, a new study shows that one in five genetics papers in top scientific journals **contains errors from the program**, *The Washington Post* reports. The errors often arose when gene names in a spreadsheet **were automatically changed** to calendar dates or numerical values. For example, one gene called *Septin-2* is commonly shortened to *SEPT2*, but is changed to 2-SEP and stored as the date 2 September 2016 by Excel. The researchers, who published their analysis in *Genome Biology*, say the issue can be fixed by formatting Excel columns as text and remaining vigilant—or switching to Google Sheets, where gene names are stored exactly as they're entered.

Source: [Science Magazine](#)

Bloomberg the Company & Its Products | Bloomberg Anywhere Remote Login | Bloomberg Terminal Demo Request

Bloomberg Businessweek

April 18, 2013, 6:31 AM EDT

FAQ: Reinhart, Rogoff, and the Excel Error That Changed History

By Peter Coy

LIVE ON BLOOMBERG Watch Live TV > Bloomberg Television Listen to Live Radio >

PHOTOGRAPH BY GREGOR SCHUSTER

Source: [Bloomberg](#)

The Office Model: Not Reproducible

Kaitlin Thaney 😊 (she/her)
@kaythaney · [Follow](#)

Twitter logo

""Reproducible research' is a redundant term.
'Irreproducible research' just used to be known as
'bullshit'." - [@fperez_org](#) ::slow clap::

7:11 PM · May 8, 2014

57 Reply Copy link to Tweet

[Read 4 replies](#)



Drawing the Rest of the Owl

How to draw an owl

1.



2.



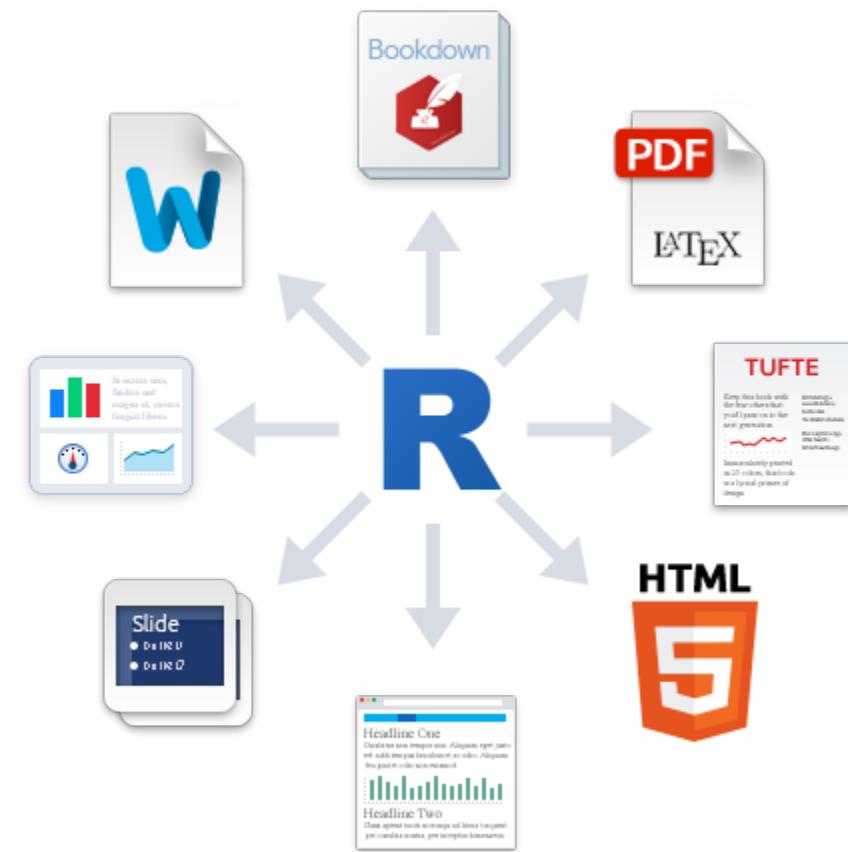
1. Draw some circles

2. Draw the rest of the fucking owl



What I'll Show You

- This is how I make my...
 - Research papers
 - Course documents
 - Websites
 - Slides and presentations
- I have not used any MS Office products since 2011 (good riddance!)
- **This stuff is optional**
 - If you like your office model, you can keep it
 - But this is what most people who take this course continue to use (R is only really if you have data work)



The Plain Text Model

The Plain Text Model II

Meet [Quarto](#), which can do *all of this* in one pipeline

1. Writing text/documents
2. Managing citations and bibliographies
3. Performing data analysis
4. Making figures and tables
5. Saving files for future use
6. Monitoring changes in documents
7. Collaborating and sharing with others
8. Combining into a deliverable (report, paper, presentation, etc.)

Welcome to Quarto

Quarto® is an open-source scientific and technical publishing system built on [Pandoc](#)

- Create dynamic content with [Python](#), [R](#), [Julia](#), and [Observable](#).
- Author documents as plain text markdown or [Jupyter](#) notebooks.
- Publish high-quality articles, reports, presentations, websites, blogs, and books in HTML, PDF, MS Word, ePub, and more.
- Author with scientific markdown, including equations, citations, crossrefs, figure panels, callouts, advanced layout, and more.

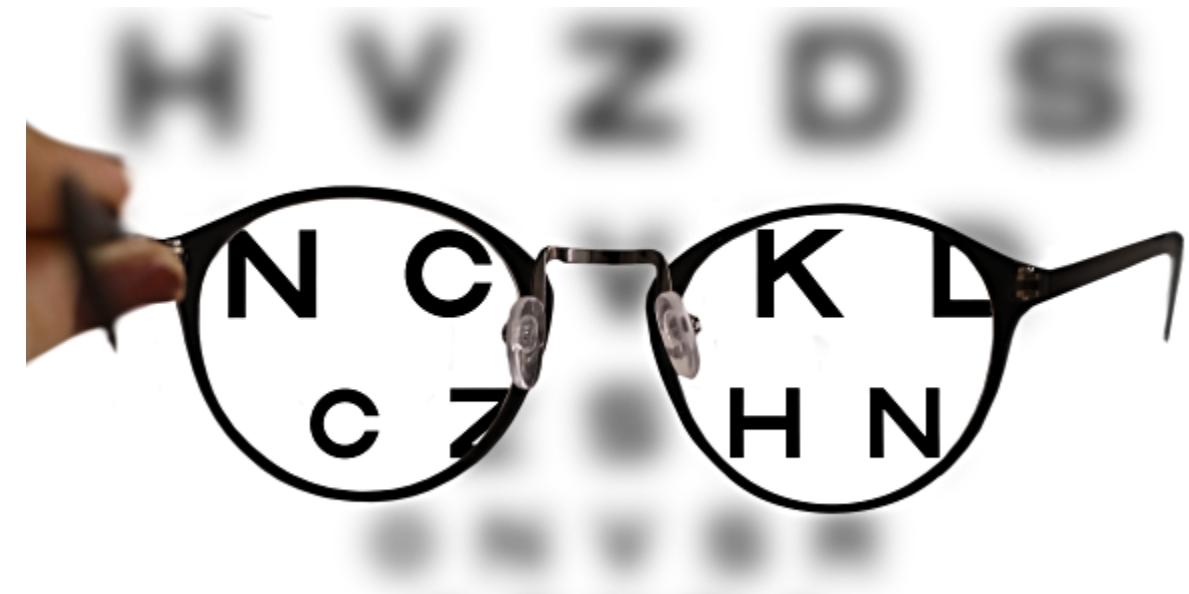
[Get Started](#)[Guide](#)

[Quarto.org](#)



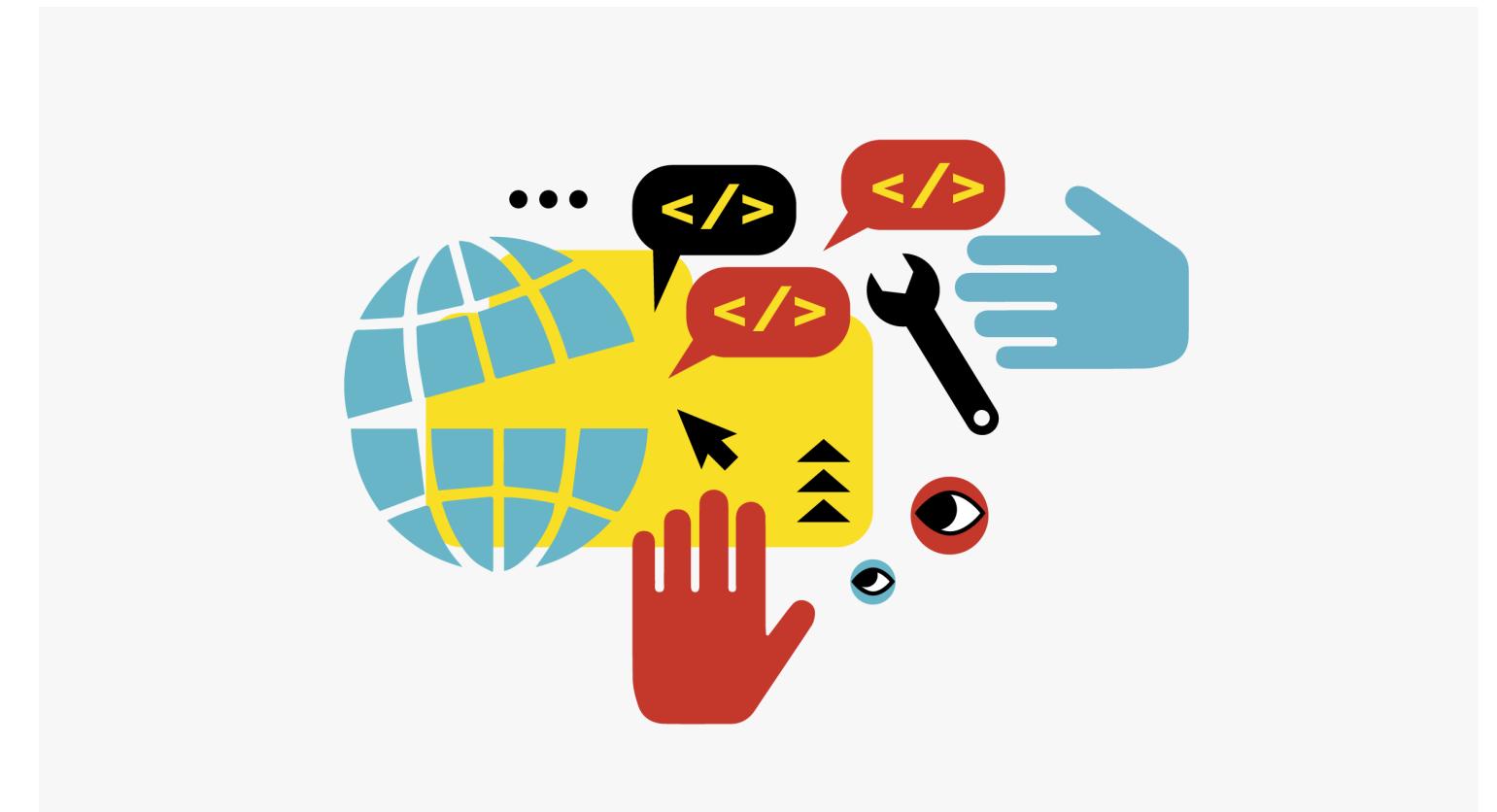
The Plain Text Model II

- Plain text files: readable by *both* machines and humans
 - Understand how a document is structured and formatted via code and markup to text
- Focus entirely on the *actual writing of the content* instead of the formatting and aesthetics
 - You can still customize, but with precise commands instead of point, click, drag, guess, pray



The Plain Text Model III

- **Open Source:** free, useable forever, often very small file size
 - Proprietary software is a gamble - can you still open a `.doc` file from Microsoft Word 1997?
- **Automate and Minimize Errors**, especially in repetitive processes
- Can be used with **version control** (see below)



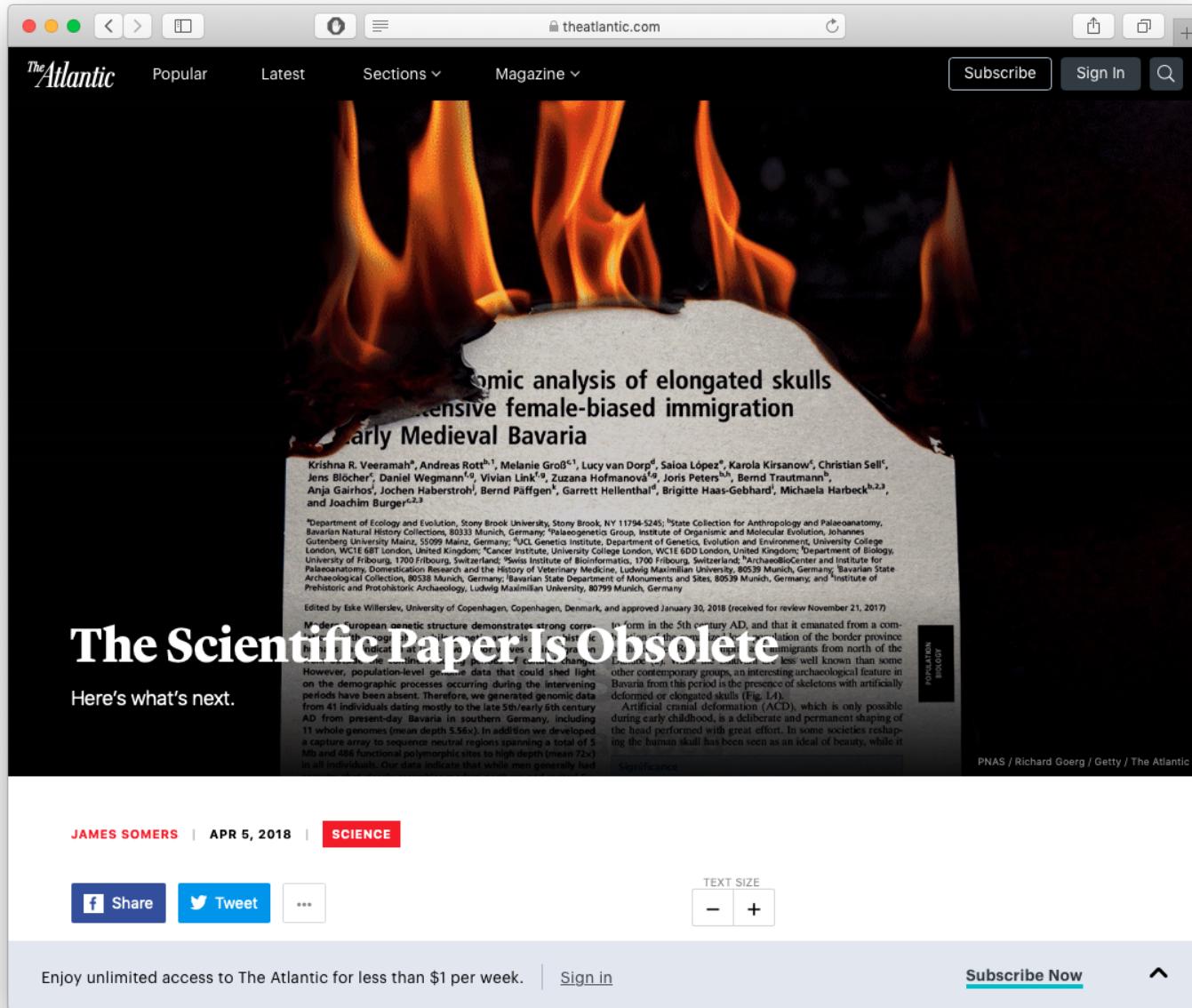
Making Your Work Reproducible



- Quarto file (`.qmd`) is the “real” part of your analysis, *everything* can live in this plain-text file!
- Document text in `markdown`
- `R code` executed in “chunks”
- Plots and tables generated from `R code`
- Citations and bibliography automated with `.bib` file

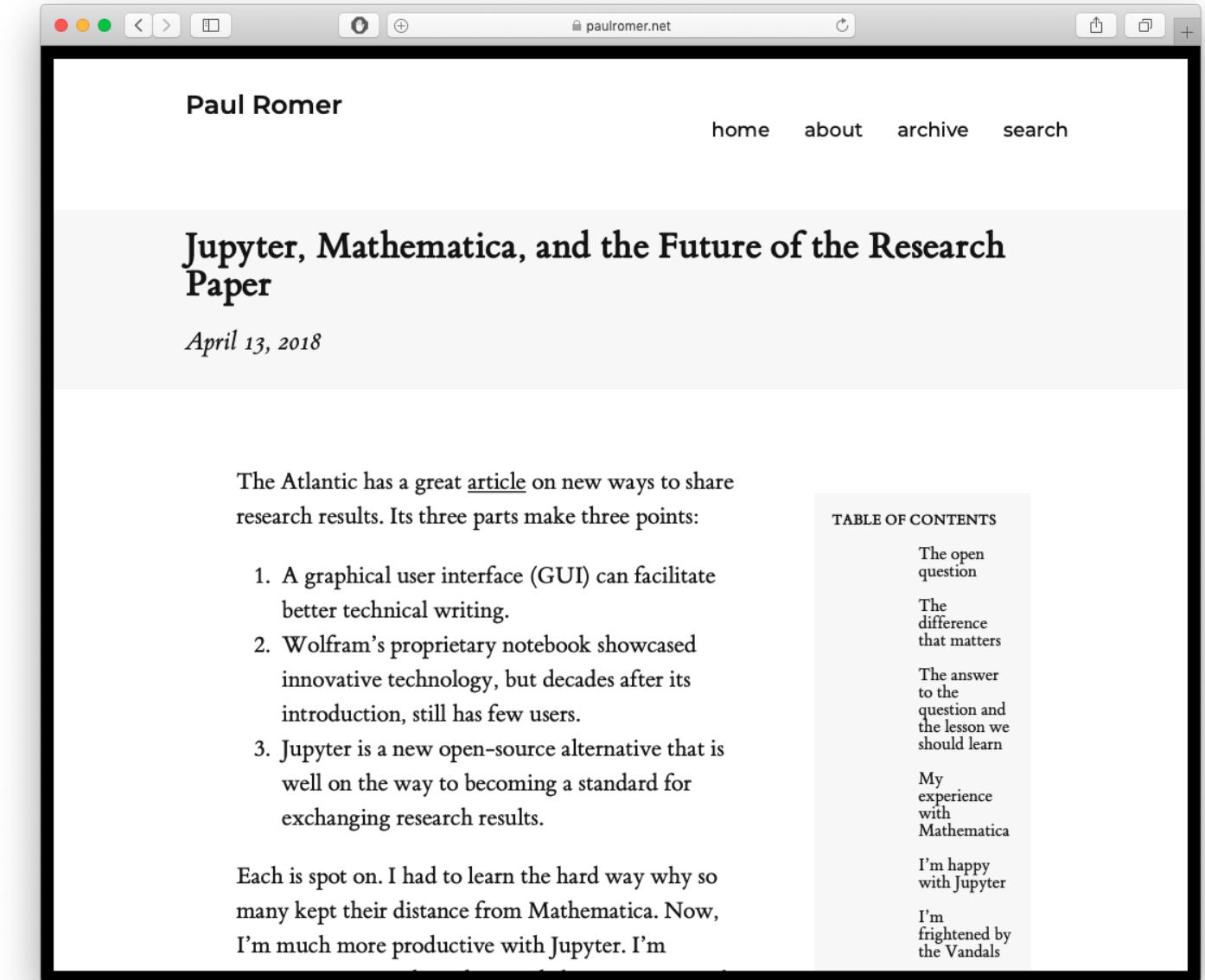


The Future of Science is Open Source Plain Text



The screenshot shows a web browser window for The Atlantic. The main headline is "The Scientific Paper Is Obsolete". Below it, a sub-headline reads "A genomic analysis of elongated skulls suggests extensive female-biased immigration in early Medieval Bavaria". The article features a large image of flames at the top. The text discusses genetic data from 41 individuals, mentioning artificial cranial deformation (ACD) and its history in Bavaria. At the bottom, there's a "Significance" section and a "TEXT SIZE" adjustment tool.

Source: [The Atlantic](#)



The screenshot shows a web browser window for paulromer.net. The title of the post is "Jupyter, Mathematica, and the Future of the Research Paper", dated April 13, 2018. The post discusses the advantages of Jupyter over traditional scientific paper formats. To the right, there's a sidebar titled "TABLE OF CONTENTS" with several sections listed vertically:

- The open question
- The difference that matters
- The answer to the question and the lesson we should learn
- My experience with Mathematica
- I'm happy with Jupyter
- I'm frightened by the Vandals

Below the sidebar, a quote from the post states: "Each is spot on. I had to learn the hard way why so many kept their distance from Mathematica. Now, I'm much more productive with Jupyter. I'm

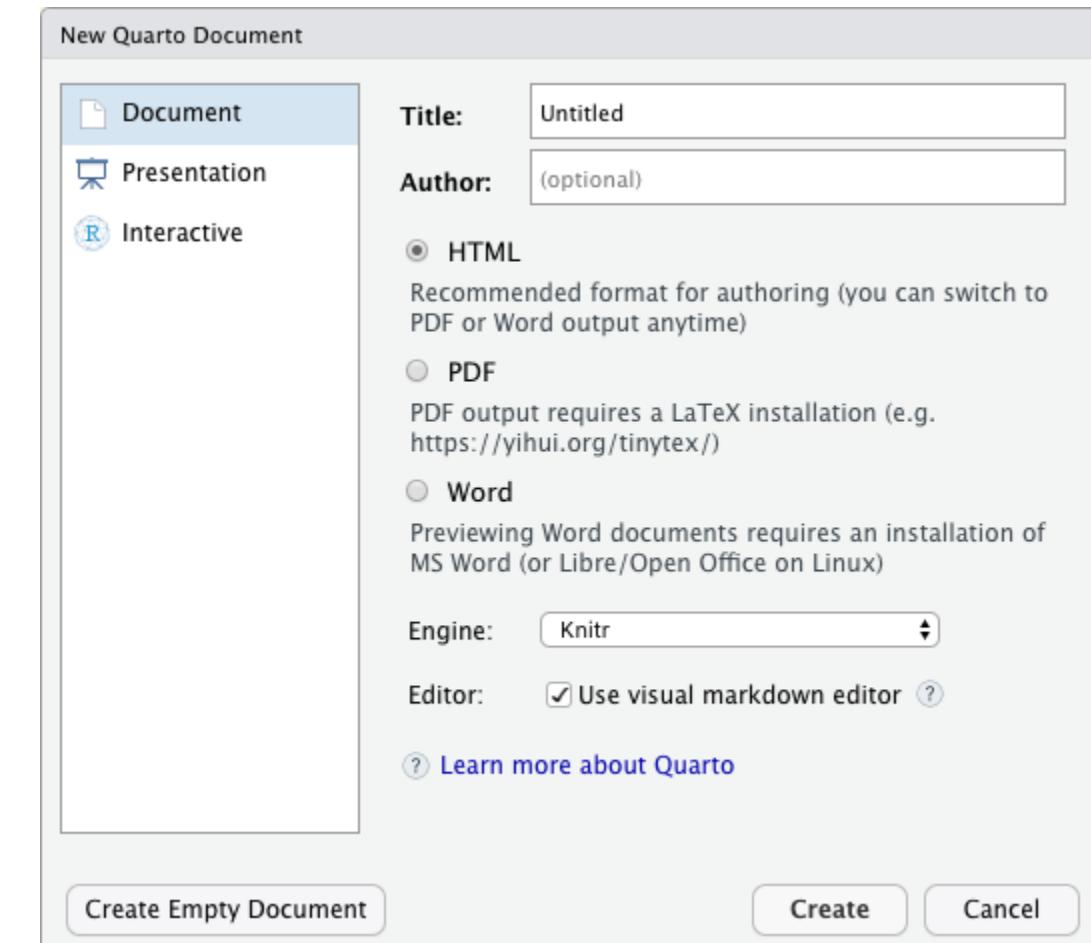
Source: [Paul Romer \(2018 Economics Nobel\)](#)



Creating a Quarto Document I

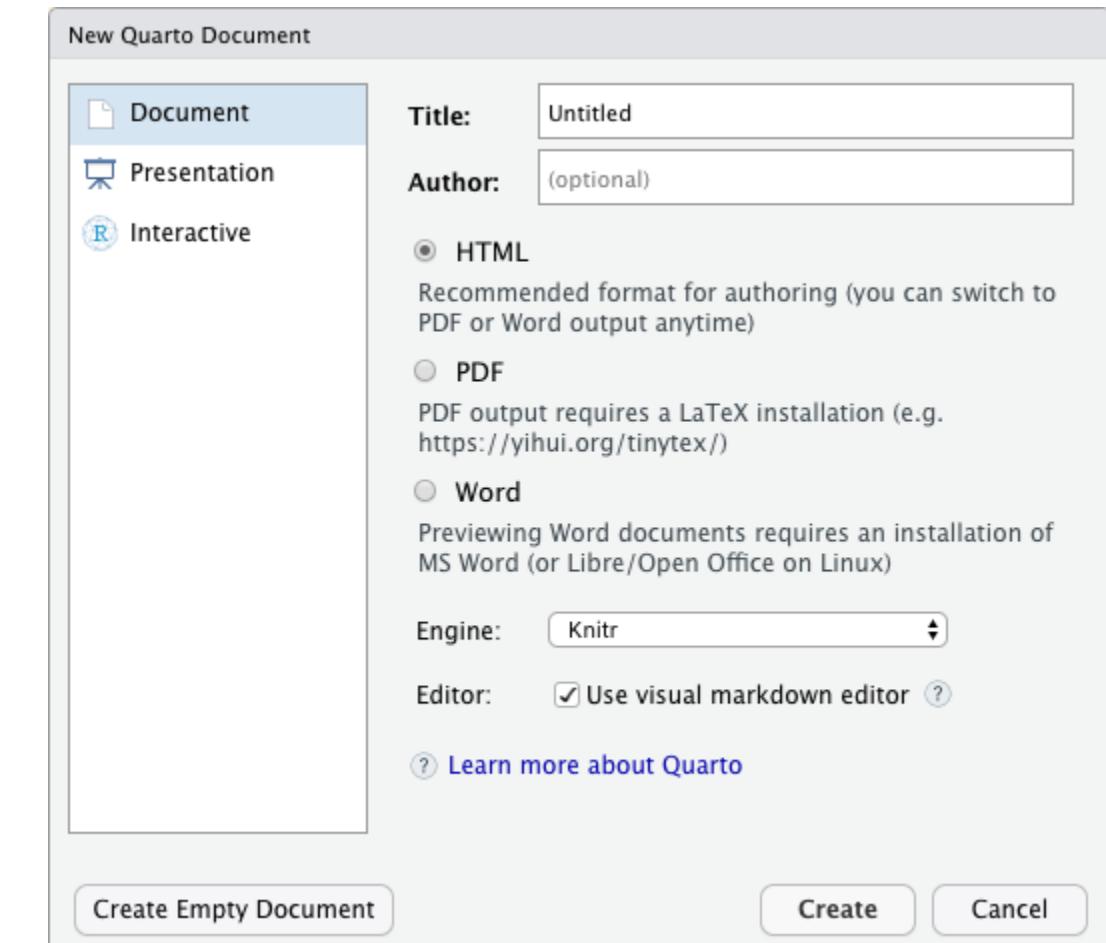
File → New File → Quarto Document...

- Outputs:
 - Document (what you'll use for most things)
 - Presentation (for making slides in various formats)
 - Interactive (an html and R based web app, advanced)



Creating a Quarto Document I

- **html**: renders a webpage, viewable in any browser
 - default, easiest to produce and share
 - can have interactive elements (gifs, animations, web apps)
 - requires internet connection to host and share (*you can view offline*)
- **pdf**: renders a PDF document
 - most common document format around
 - requires **LaTeX** distribution to render (more on that soon)
- **word**: create a Microsoft Word document
 - ...if you must



Structure of a Quarto Document

Entire document is written in a single file with three types of content:

1. YAML header for metadata

```
1 ---  
2 title: "Title"  
3 format: html  
4 ---
```

2. Text of the document written with markdown

```
1 # Header 1  
2 **Bold** and *italic* text.
```

3. R chunks for data analysis, plots, figures, tables, statistics, as necessary

```
1 2+2
```

```
[1] 4
```

The screenshot shows the Quarto editor interface. At the top is a toolbar with various icons for file operations, search, and rendering. Below the toolbar is a menu bar with 'Source' and 'Visual' tabs selected. The main area is divided into sections: 'Quarto' which contains the YAML header, 'Running Code' which contains a brief introduction and an R code chunk, and 'R' which contains an R code chunk with options like 'echo: false'.

Quarto

```
---
```

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

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
{r}
1 + 1
```

You can add options to executable code like this

```
{r}
#| echo: false
2 * 2
```

The `echo: false` option disables the printing of code (only output is displayed).



YAML Header I

- Top of a document contains the **YAML**¹ separated by three dashes **---** above and below
- Contains the **metadata** of the document, such as:

```
1 ---  
2 title: "My Document"  
3 author: "Ryan Safner"  
4 date: "`r Sys.Date()`" # here I'm using R code to generate today's date!  
5 format: html  
6 ---
```

- **format** must be specified, everything else can be left blank, and other options can be added as necessary
- In most cases, you can safely ignore other things in the **yaml** until you are ready

¹ YAML stands for “YAML Ain’t Markup Language.” Nerds love recursive acronyms.



YAML Header Example I

- Example from these slides

```
1 ---  
2 format:  
3   revealjs:  
4     theme: [default, custom.scss]  
5     logo: "../images/metrics_hex.png"  
6     footer: "[ECON 480 – Econometrics]({{ site.url }}metricsF22.classes.ryansafner.com)"  
7     height: 900  
8     width: 1600  
9     #df-print: paged  
10    slide-number: c  
11 overview: true  
12 execute:  
13   echo: false  
14   warning: false  
15   freeze: auto  
16 ---
```



YAML Header Example II

- Example from one of my papers:

```
1 ---  
2 title: Distributing Patronage^I would like to thank the Board of Associates of Hood College...]  
3 subtitle: Intellectual Property in the Transition from Natural State to Open Access Order  
4 date: \today  
5 author:  
6 - Ryan Safner^Hood College, Department of Economics and Business Administration; safner@hood.edu]  
7  
8 abstract: |  
9   "This paper explores the emergence of the modern forms of copyright and patent in ...  
10  *JEL Classification:* O30, O43, N43  
11  *Keywords:* Copyright, intellectual property, economic history, freedom of the press, economic development  
12  
13 bibliography: patronage.bib  
14 geometry: margin = 1in  
15 fontsize: 12pt  
16 mainfont: Fira Sans Condensed  
17 output:
```



R Chunks

- You can create a “**chunk**” of R code with **three backticks**¹ above and below your code
- After the first pair of backticks, signify the **language** of the code² inside braces, e.g:

Input

```
1 ````{r}
2 2+2 # code goes here!
3 ````
```

Output

```
[1] 4
```

1. The key to the left of the #1 key on your keyboard.

2. Yes that does mean you can use other coding languages!



R Chunks

Input

```
1 `~`{r}
2 gapminder %>%
3   head()
4 `~`
```

Output

```
# A tibble: 6 × 6
  country    continent year lifeExp      pop gdpPercap
  <fct>      <fct>     <int>   <dbl>     <int>     <dbl>
  1 Afghanistan Asia      1952    28.8     8425333    779.
  2 Afghanistan Asia      1957    30.3     9240934    821.
  3 Afghanistan Asia      1962    32.0    10267083    853.
  4 Afghanistan Asia      1967    34.0    11537966    836.
  5 Afghanistan Asia      1972    36.1    13079460    740.
  6 Afghanistan Asia      1977    38.4    14880372    786.
```



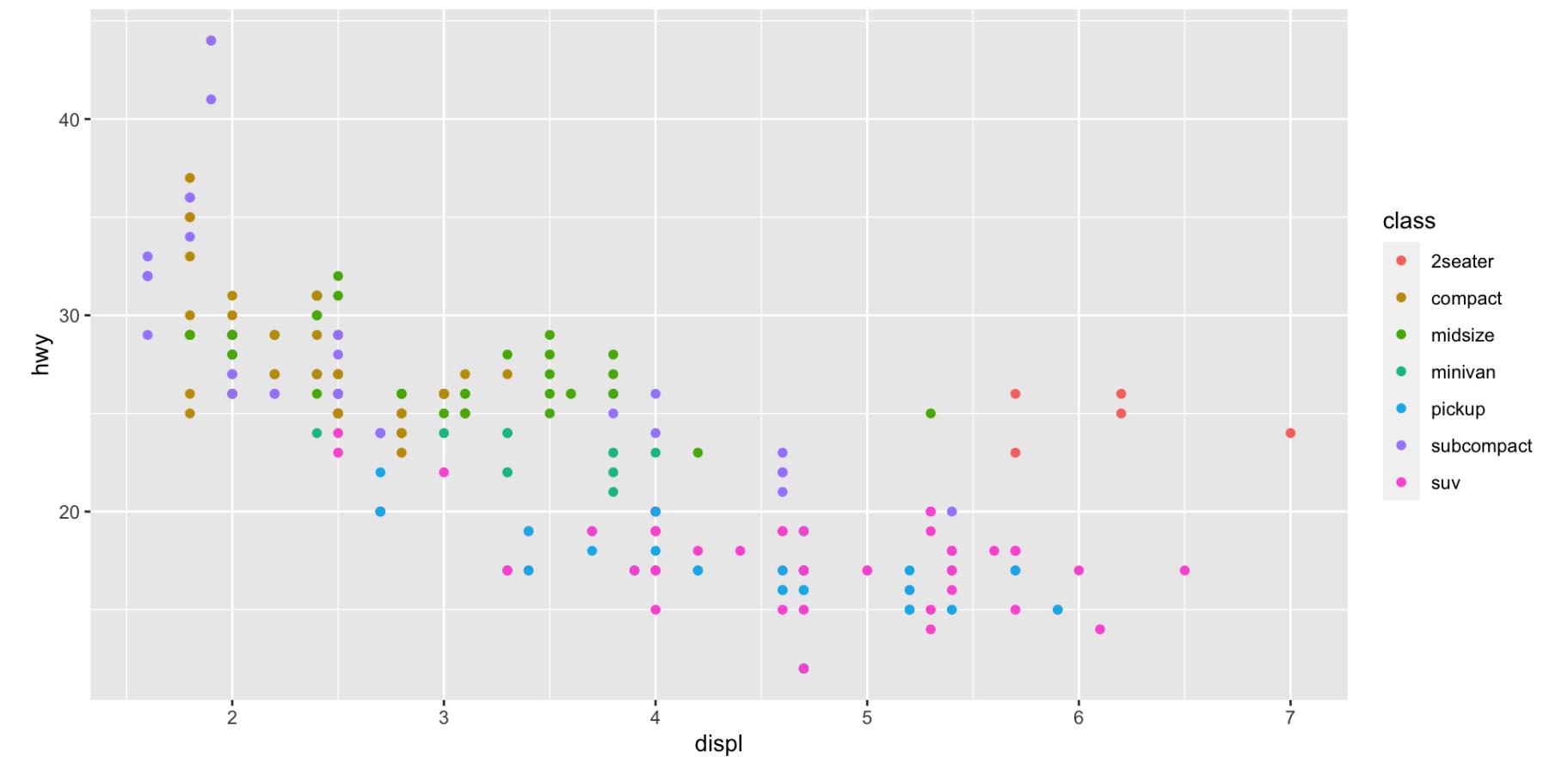
R Chunks

Input

```

1  ````{r}
2  library("ggplot2") # load ggplot2
3  ggplot(data = mpg) +
4    aes(x = displ,
5         y = hwy) +
6    geom_point(aes(color = class))
7  ````
```

Output



R Chunk Options

- Chunks can have options with the “hash pipe” `#|` at the top of the chunk

```
1 #| label: my_chunk_title # give chunk a name
2 #| eval: true # run the code?
3 #| echo: true # display code?
4 #| warning: true # display warnings?
5 #| message: false # display messages?
6 #| fig.width: 6 # width for figures
```

- In [R Markdown](#) (the predecessor to Quarto...that I know better), you put options inside the braces at the top of a chunk. This is still valid in [Quarto](#):

```
```{r my_chunk_title, eval = T, echo = F, warning = F, message = F, fig.width = 6}
```

```



Global Chunk Options

- You can set default options for all chunks in the **YAML** header:

```
1 execute:  
2   echo: false # hide all input code  
3   warning: false # hide all output warnings  
4   message: false # hide all output messages
```

- Learn more [here](#) and [here](#)



R Inline Code

- If you just want to display some code (or at least format it like code) in the middle of a sentence, **place between a single backtick on either side.**
 - e.g. if I mention `tidyverse` or `gapminder`, it formats the text as **in-line code**.
- To actually **execute** R code to output something in the middle of a sentence, put **r** as the first character inside the backticks, and then run the actual code such as pi is equal to 3.1415927.

Input

pi is equal to `r pi`.

Output

pi is equal to 3.1415927.



Or Like This

Input

```
The average GDP per capita is `r  
dollar(mean(gapminder$gdpPercap))`  
with a standard deviation of `r  
dollar(sd(gapminder$gdpPercap))`  
`.
```

Output

The average GDP per capita is \$7,215.33 with a standard deviation of \$9,857.45.



Writing Text with Markdown: Formatting

- **Markdown** is a lightweight markup language geared towards HTML (i.e. the internet)
 - **Markup languages** used to add commands about how to display plain text

| Markdown Syntax | Output |
|-------------------------------|---|
| *italics* and **bold** | <i>italics</i> and bold |
| superscript^2^ / subscript~2~ | superscript ² / subscript ₂ |
| ~~strikethrough~~ | strikethrough |
| `verbatim code` | verbatim code |

- Comment your document with `<!-- Unprinted comments here -->` (will not print in output; this comes from `html`)



Writing Text with Markdown: Lists

| Markdown Syntax | Output |
|---|--|
| | <ul style="list-style-type: none"> • unordered list <ul style="list-style-type: none"> ▪ sub-item 1 ▪ sub-item 2 ○ sub-sub-item 1 |
| * unordered list + sub-item 1 + sub-item 2 - sub-sub-item 1 | |
| | <ul style="list-style-type: none"> • item 2 <p>Continued (indent 4 spaces)</p> |
| * item 2 <p>Continued (indent 4 spaces)</p> | |
| | <ol style="list-style-type: none"> 1. ordered list 2. item 2 <ol style="list-style-type: none"> 1. sub-item 1 <ol style="list-style-type: none"> 1. sub-sub-item 1 |
| 1. ordered list 2. item 2 i) sub-item 1 A. sub-sub-item 1 | |



Writing Text with Markdown: Headings

Markdown Syntax

```
# Header 1
```

```
## Header 2
```

```
### Header 3
```

```
#### Header 4
```

```
##### Header 5
```

```
###### Header 6
```

Output

Header 1

Header 2

Header 3

Header 4

Header 5

Header 6



Writing Text with Markdown: Links

Markdown

```
1 You can embed  
2 [named hyperlinks](https://metricsF22.classes.ryan  
3 direct urls like <https://metricsF22.classes.ryans  
4 and links to  
5 [other places – like the previous slide](#writing-  
6 in the document.
```

Output

You can embed **named hyperlinks**, direct urls like <https://metricsF22.classes.ryansafner.com/>, and links to **other places** – like the previous slide in the document.



Writing Text with Markdown: Images



Markdown

```
1 ! [Caption: tidyverse](images/tidyversel.png)
2
3 ! [](https://dplyr.tidyverse.org/logo.png)
```

Output



The tidyverse



dplyr logo



Writing Text with Markdown: Making Tables

Markdown

| Right | Left | Default | Center |
|--------|--------|---------|--------|
| -----: | -----: | -----: | -----: |
| 12 | 12 | 12 | 12 |
| 123 | 123 | 123 | 123 |
| 1 | 1 | 1 | 1 |

```
: Table Example {tbl-colwidths=" [25,25,25,25] "}
```

Output

Table Example

| Right | Left | Default | Center |
|-------|------|---------|--------|
| 12 | 12 | 12 | 12 |
| 123 | 123 | 123 | 123 |
| 1 | 1 | 1 | 1 |

- See the Quarto Documentation for more help on [tables](#)



Writing Text with Markdown: Printing Tables

- Sometimes we want to print tables from our data
- The `kableExtra` package is great for this [see Documentation](#)

```

1 library(kableExtra)
2 mtcars %>%
3   head() %>%
4   kbl()

```

| | mpg | cyl | disp | hp | drat | wt | qsec | vs | am | gear | carb |
|-------------------|------------|------------|-------------|-----------|-------------|-----------|-------------|-----------|-----------|-------------|-------------|
| Mazda RX4 | 21.0 | 6 | 160 | 110 | 3.90 | 2.620 | 16.46 | 0 | 1 | 4 | 4 |
| Mazda RX4 Wag | 21.0 | 6 | 160 | 110 | 3.90 | 2.875 | 17.02 | 0 | 1 | 4 | 4 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 | 2.320 | 18.61 | 1 | 1 | 4 | 1 |
| Hornet 4 Drive | 21.4 | 6 | 258 | 110 | 3.08 | 3.215 | 19.44 | 1 | 0 | 3 | 1 |
| Hornet Sportabout | 18.7 | 8 | 360 | 175 | 3.15 | 3.440 | 17.02 | 0 | 0 | 3 | 2 |
| Valiant | 18.1 | 6 | 225 | 105 | 2.76 | 3.460 | 20.22 | 1 | 0 | 3 | 1 |



Writing Text with Markdown: Printing Tables

- Sometimes we want to print tables from our data
- The `kableExtra` package is great for this [see Documentation](#)

```

1 library(kableExtra)
2 mtcars %>%
3   head() %>%
4   kbl()

```

| | mpg | cyl | disp | hp | drat | wt | qsec | vs | am | gear | carb |
|-------------------|------------|------------|-------------|-----------|-------------|-----------|-------------|-----------|-----------|-------------|-------------|
| Mazda RX4 | 21.0 | 6 | 160 | 110 | 3.90 | 2.620 | 16.46 | 0 | 1 | 4 | 4 |
| Mazda RX4 Wag | 21.0 | 6 | 160 | 110 | 3.90 | 2.875 | 17.02 | 0 | 1 | 4 | 4 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 | 2.320 | 18.61 | 1 | 1 | 4 | 1 |
| Hornet 4 Drive | 21.4 | 6 | 258 | 110 | 3.08 | 3.215 | 19.44 | 1 | 0 | 3 | 1 |
| Hornet Sportabout | 18.7 | 8 | 360 | 175 | 3.15 | 3.440 | 17.02 | 0 | 0 | 3 | 2 |
| Valiant | 18.1 | 6 | 225 | 105 | 2.76 | 3.460 | 20.22 | 1 | 0 | 3 | 1 |



Writing Text with Markdown: Printing Tables

```
1 mtcars %>%
2   head() %>%
3   rmarkdown::paged_table()
```

| | mpg
<dbl> | cyl
<dbl> |
|-------------------|--------------|--------------|
| Mazda RX4 | 21.0 | 6 |
| Mazda RX4 Wag | 21.0 | 6 |
| Datsun 710 | 22.8 | 4 |
| Hornet 4 Drive | 21.4 | 6 |
| Hornet Sportabout | 18.7 | 8 |
| Valiant | 18.1 | 6 |

6 rows | 1-3 of 12 columns



Writing Math

- Add beautifully-formatted math with the `$` tag before and after the math, two `$$` before/after for a centered equation
- In-line math example: `$1^2=\frac{\sqrt{16}}{4}$` produces $1^2 = \frac{\sqrt{16}}{4}$ in my text
- Centered-equation example:

Input

```
$$
\hat{\beta}_1 = \frac{\displaystyle \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\displaystyle \sum_{i=1}^n (X_i - \bar{X})^2} $$

```

Output

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2}$$



Writing Math

- Math uses a (much older) language called \LaTeX , used by mathematicians, economists, and others to write papers and slides with perfect math \LaTeX formatting
 - I used to use for everything before I found R and markdown
 - Producing pdf output actually converts markdown files into first! \LaTeX
 - Much steeper learning curve, a good cheatsheet
 - An extensive library of mathematical symbols, notation, formats, and ligatures, e.g.
- A great resource: Wikibooks LaTeX Mathematics chapter



Writing Math

Input

`α`

`π`

`$\frac{1}{2}$`

`\hat{x}`

`\bar{y}`

`$x_{1,2}$`

`x^{a-1}$`

`$\lim_x \rightarrow \infty$`

`$A=\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$`

Output

α

π

\hat{x}

\bar{y}

$x_{1,2}$

x^{a-1}

$\lim_{x \rightarrow \infty}$

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$$





Citations, References, & Bibliography

- Manage your citations and bibliography automatically with `.bib` files
- First create a `.bib` file to list all of your references in
 - You can do this in R via: File → New File → Text File (and save with `.bib` at the end)
 - See `examplebib.bib` in this repository used in this document
 - At the top of your YAML header in the main document, add bibliography: `examplebib.bib` so R knows to pull references from this file
 - For each reference, add information to a `.bib` file, like so:



An Example .bib File

```
1 @article{safner2016,
2   author = {Ryan Safner},
3   year = {2016},
4   journal = {Journal of Institutional Economics},
5   title = {Institutional Entrepreneurship, Wikipedia, and the Opportunity of the Commons},
6   volume = {12},
7   number = {4},
8   pages = {743-771}
9 }
10 }
```

- A **.bib** file is a plain text file with entries like this
- Classes for **@article**, **@book**, **@collectedwork**, **@unpublished**, etc.
 - Each will have different keys needed (e.g. **editor**, **publisher**, **address**)
- First input after the **@article** is your **citation key** (e.g. **safner2016**)
 - Whenever you want to cite this article, you'll invoke this key



Citations

- Whenever you want to cite a work in your text, call up the **citation key** with @, like so: `@safner2016 []`, which produces (Safner, 2016)
- You can customize citations, e.g.:

| Write | Produces |
|---------------------------------------|---------------------------|
| <code>[@Safner2016]</code> | (Safner, 2016) |
| <code>@Safner2016</code> | Safner 2016 |
| <code>-@Safner2016</code> | (2016) |
| <code>@Safner2016 [p. 743–744]</code> | (Safner, 2016, p.743-744) |

- BibTeX will automatically collect all works cited at the end and produce a **bibliography** according to a style you can choose



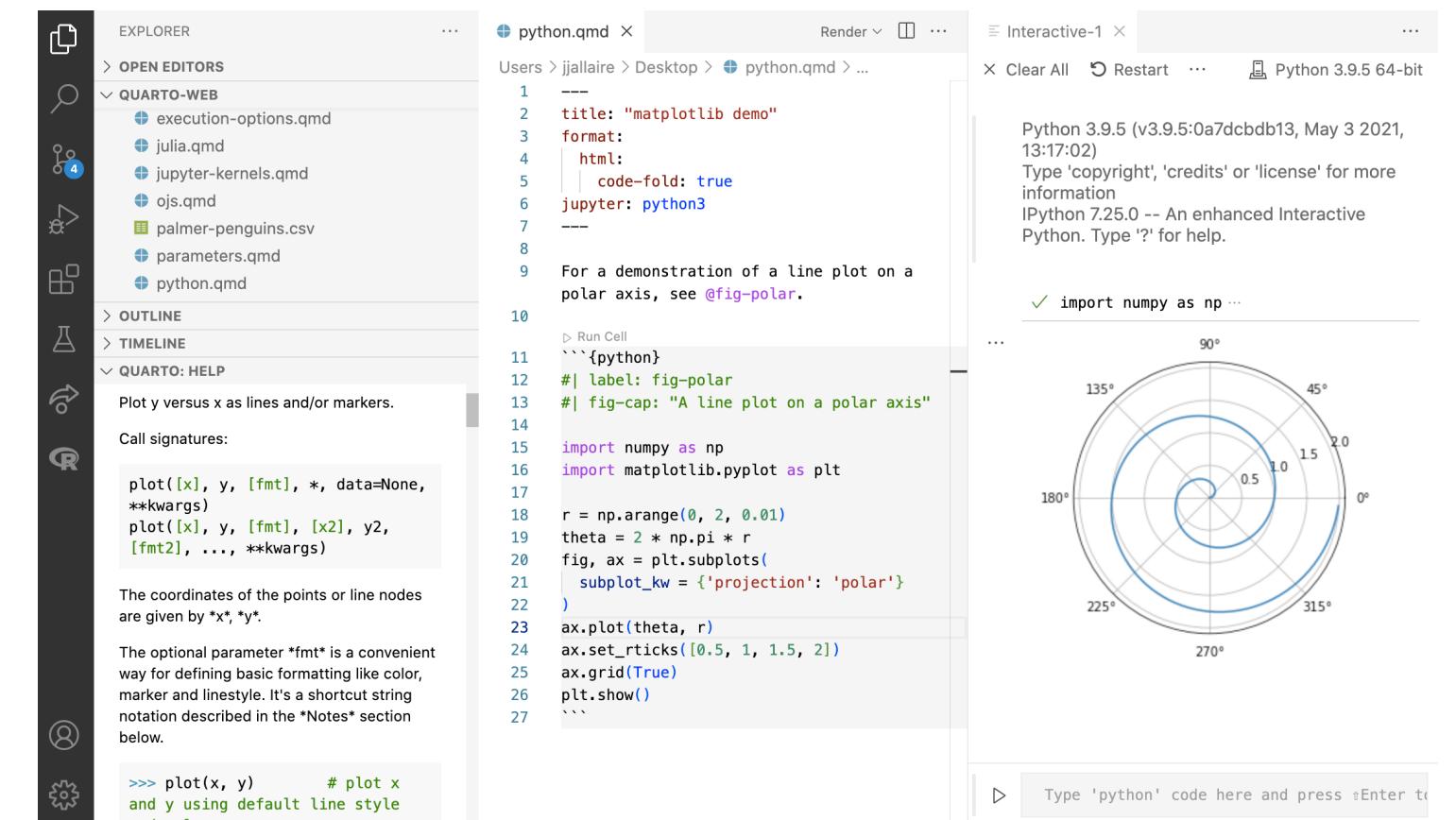
Reference Management Software

- For more information and examples, see [Quarto's Documentation on Citations](#)
- Lot of programs can help you manage references and export complete `.bib` files to use with [R](#)
[Markdown](#)
 - [Mendeley](#) and [Zotero](#) are free and cross-platform
 - I use [Papers](#) (Paid and Mac only)
 - Simplest program (what I use) that makes `.bib` files is [Bibdesk](#)



Plain-Text Editors

- Markdown files are **plain text** files and can be edited in *any* text editor
 - something as basic (and boring!) as **“Notepad,”** for example
 - many good **text editors** out there: **Typora;** **Ulysses**
- Any good editor will have **syntax highlighting** and **coloring** when you use tags (like **bold**, ***italic***, **code**, and **code #comments**).
- **VS Code;** **Notepad++;** **Sublime**



The screenshot shows the Visual Studio Code (VS Code) interface. On the left is the Explorer sidebar, which lists files like 'python.qmd', 'execution-options.qmd', 'julia.qmd', 'jupyter-kernels.qmd', 'ojs.qmd', 'palmer-penguins.csv', 'parameters.qmd', and 'quarto.qmd'. The main area is a code editor with Python code. The code includes imports for numpy and matplotlib.pyplot, generates a sequence of r values from 0 to 2, creates a polar plot with theta = 2 * np.pi * r, and shows a resulting spiral plot. The bottom right corner of the interface has a small decorative icon.

```

1 ---  

2 title: "matplotlib demo"  

3 format:  

4 | html:  

5 | | code-fold: true  

6 jupyter: python3  

7 ---  

8  

9 For a demonstration of a line plot on a  

10 polar axis, see @fig-polar.  

11 >>> plot([x], y, [fmt], *, data=None,  

12 **kwargs)  

13 plot([x], y, [fmt], [x2], y2,  

14 [fmt2], ..., **kwargs)  

15  

16 import numpy as np  

17 import matplotlib.pyplot as plt  

18 r = np.arange(0, 2, 0.01)  

19 theta = 2 * np.pi * r  

20 fig, ax = plt.subplots(  

21 subplot_kw = {'projection': 'polar'})  

22 ax.plot(theta, r)  

23 ax.set_rticks([0.5, 1, 1.5, 2])  

24 ax.grid(True)  

25 plt.show()  

26  

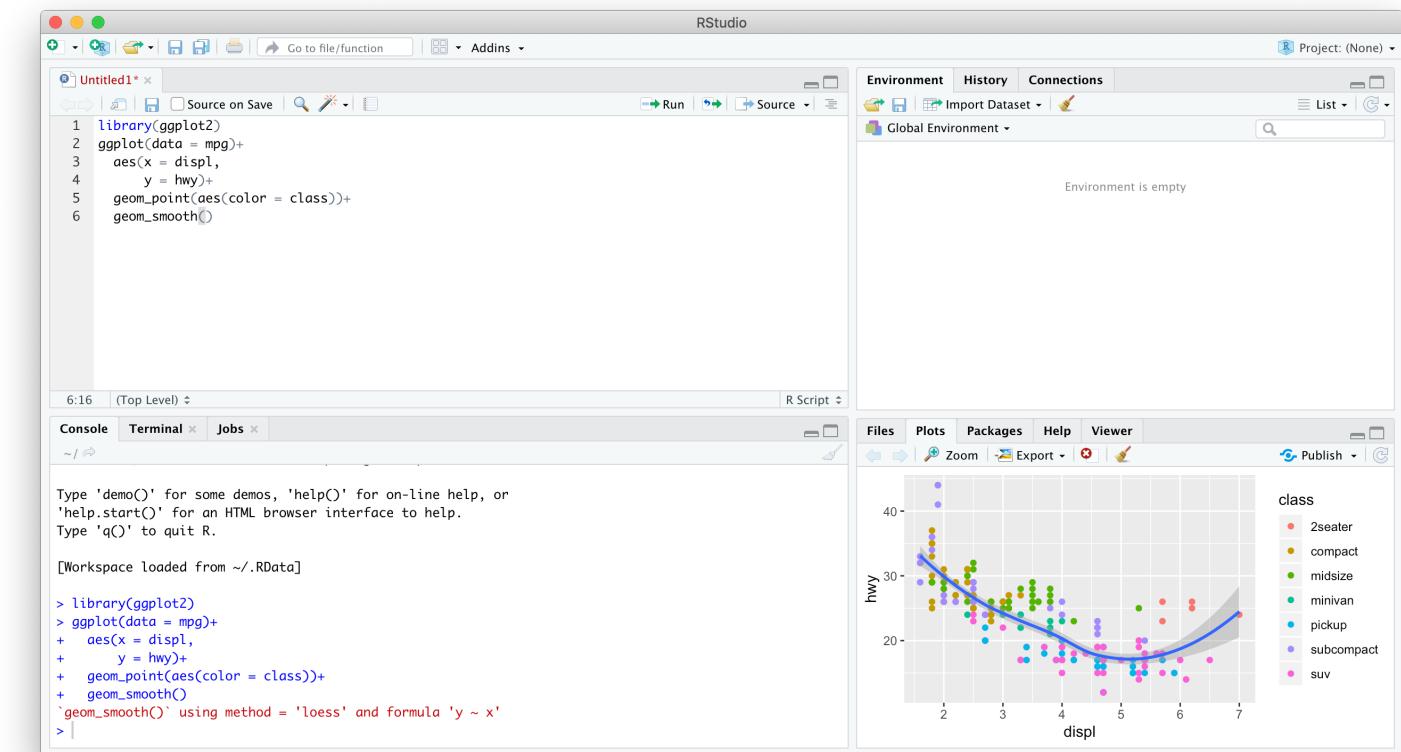
27

```

VS Code

RStudio is My Text Editor of Choice

- Honestly, I write **everything** in R Studio's text editor
 - Syntax highlighting
 - Actually can *run* R code, autocomplete, etc
 - Can render the markdown to an output format: html, pdf, etc.
- You can *write* R code in other text editors, but you can't *execute* them outside of *R Studio* (or the command line, but that's too advanced.) Same with actually rendering your markdown to an output (pdf, html, etc)



Tips with Markdown

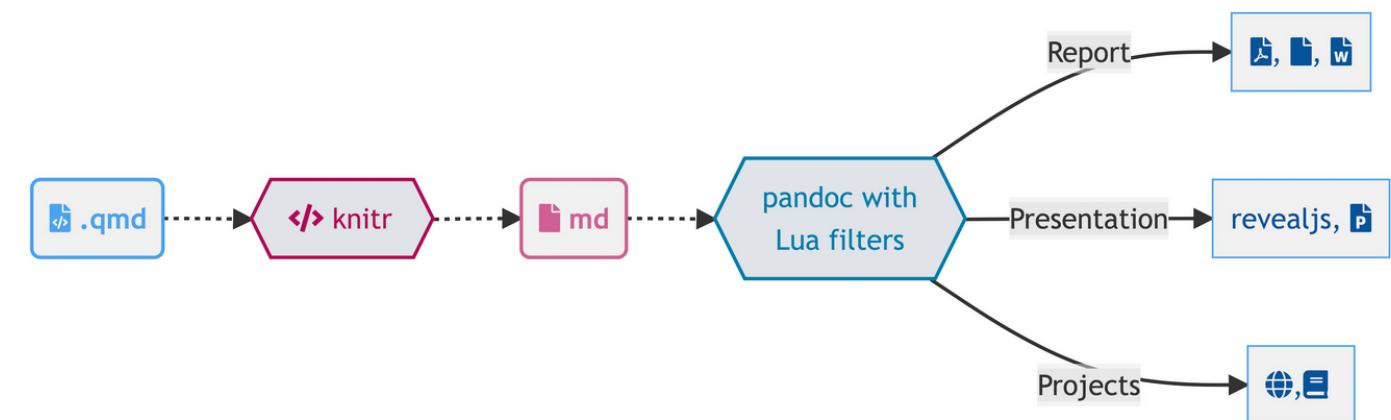
- Empty space is **very important** in markdown
- Lines that begin with a space may not render properly
- Math that contains spaces *between* the dollar-signs may not render properly
- Moving from one type of content to another (e.g. a heading to a list to text to an equation to text) requires *blank lines between them* to work
- Here is a [great general tutorial on markdown syntax](#)



Rendering Your Documents

knitr

- When you are ready, you “redner” your markdown and code into an output format using:
- **knitr**¹, an R package that “**knits**” your R code and markdown **.qmd** into a **.md** file for:
- **pandoc** is a “swiss-army knife” utility that can convert between *dozens* of document types
- All you need to do is click the **Render** button at the top of the text editor!



¹ knitr also relies on the **markdown** package, which will probably be installed when you first knit.



PDF Output

- Producing a PDF uses \LaTeX
- You will need a full distribution of \LaTeX on your computer, OR \LaTeX
- Better to use the package `tinytex` to install a mini-distro of \LaTeX inside of R:

```
1 # install.packages("tinytex") # first install package
2 library(tinytex) # load package
3 install_tinytex() # run this command to install LaTeX in R
```

- Once you've done this (just one time), you can Render to a PDF, make sure your YAML header is set to `pdf` format:

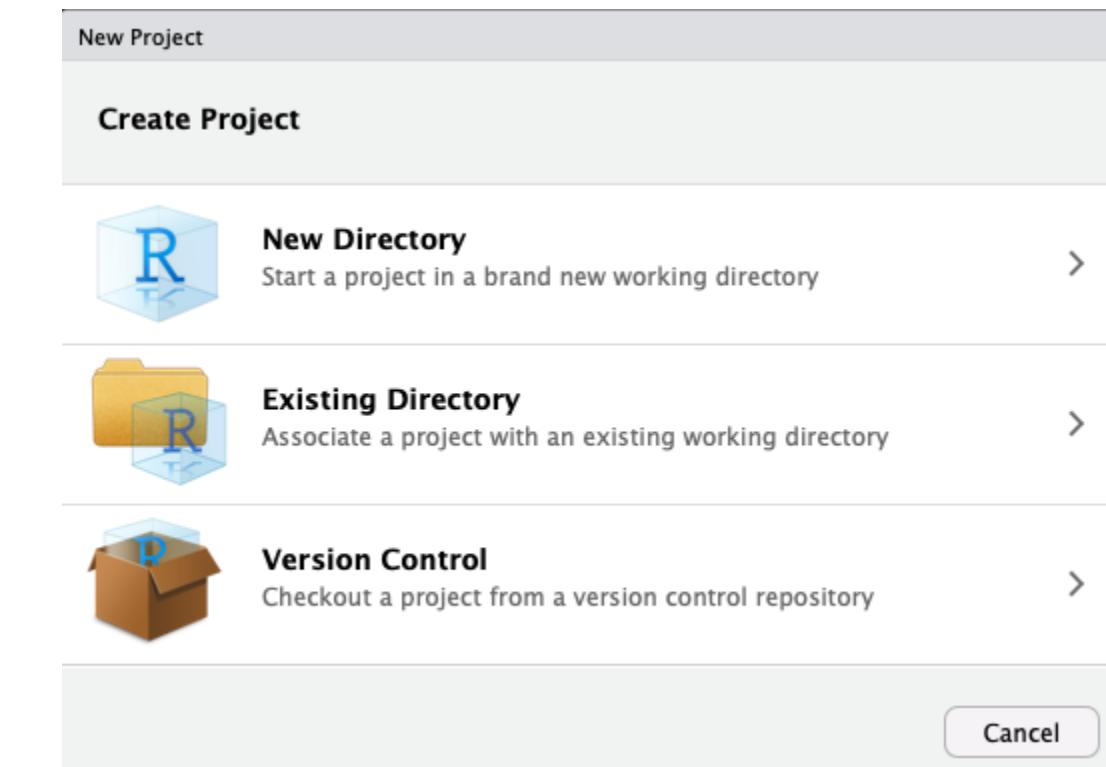
```
1 ---
2 format: pdf
3 ---
```



Project-Oriented Workflow

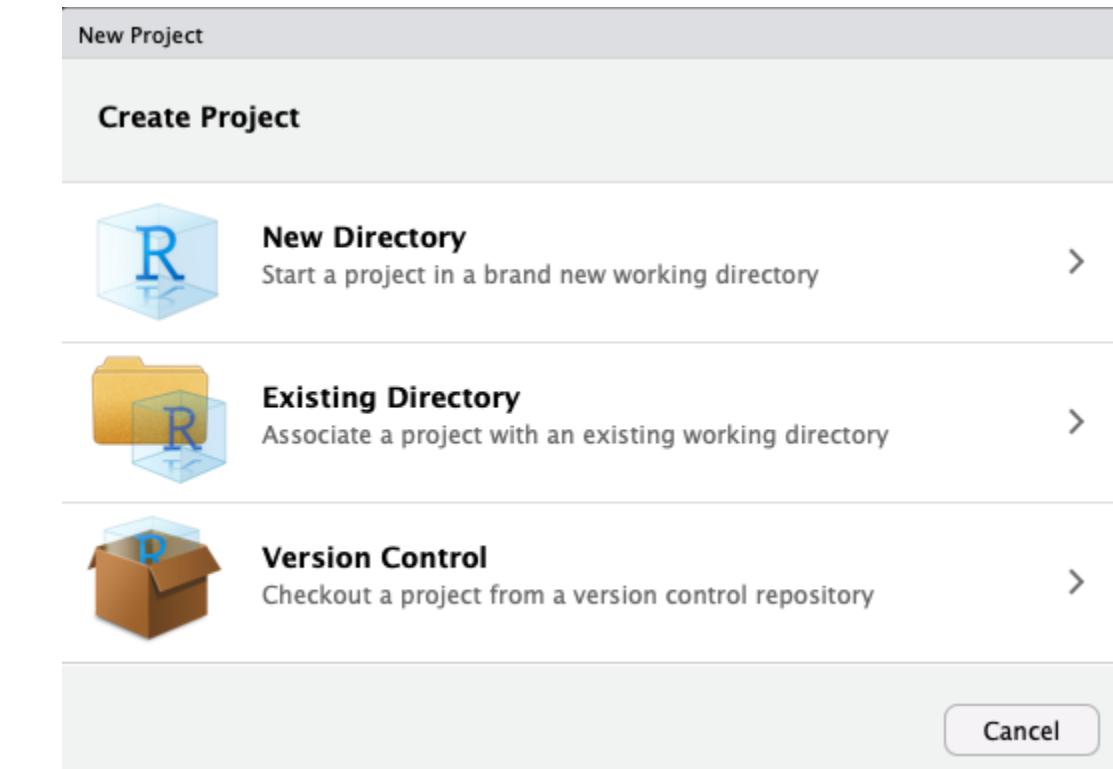
R Projects I

- A **R Project** is a way of systematically organizing your **R** history, working directory, and related files in a single, self-contained directory
- Can easily be sent to others who can reproduce your work easily
- Connects well with version control software like GitHub
- Can open multiple projects in multiple windows



R Projects I

- Projects solve all of the following problems:
 1. Organizing your files (data, plots, text, citations, etc)
 2. Having an accessible working directory (for loading and saving data, plots, etc)
 3. Saving and reloading your commands history and preferences
 4. Sending files to collaborators, so they have the same working directory as you



Creating an R Project I

New Project

Create Project

 **New Directory**
Start a project in a brand new working directory >

 **Existing Directory**
Associate a project with an existing working directory >

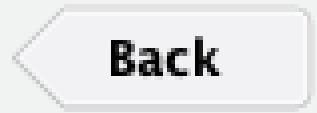
 **Version Control**
Checkout a project from a version control repository >

[Cancel](#)



Creating an R Project II

New Project

 Back Project Type

-  New Project >
-  R Package >
-  Shiny Web Application >
- R Package using Rcpp >
- R Package using RcppArmadillo >
- R Package using RcppEigen >
-  Website using blogdown >

 Cancel



Creating an R Project III

New Project

[Back](#)

Create New Project



Directory name:

Create project as subdirectory of:

 [Browse...](#)

Create a git repository

Use packrat with this project

Open in new session

[Create Project](#) [Cancel](#)



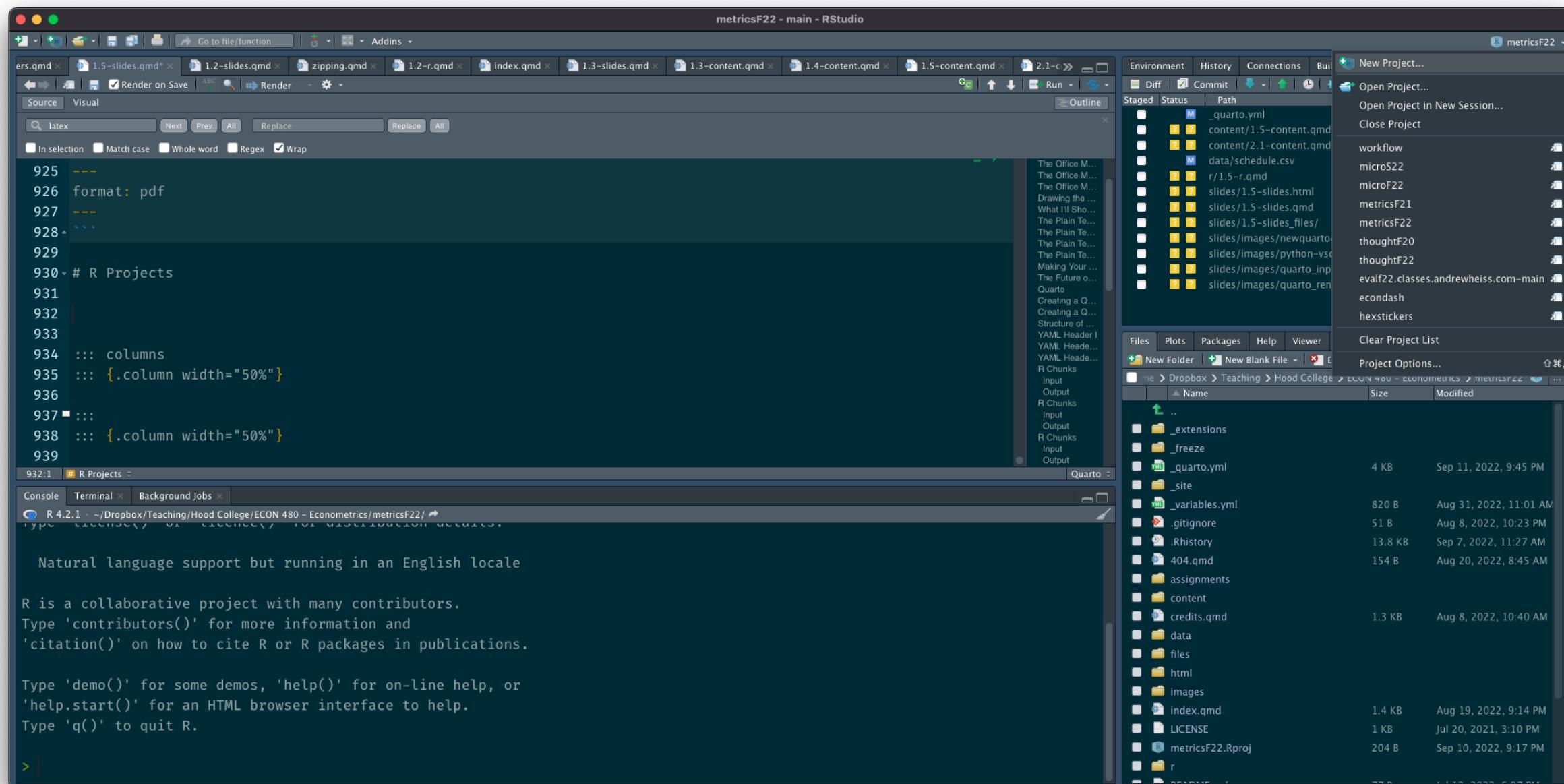
Projects

- Switch between each project (Window) on your computer (this is on a Mac)

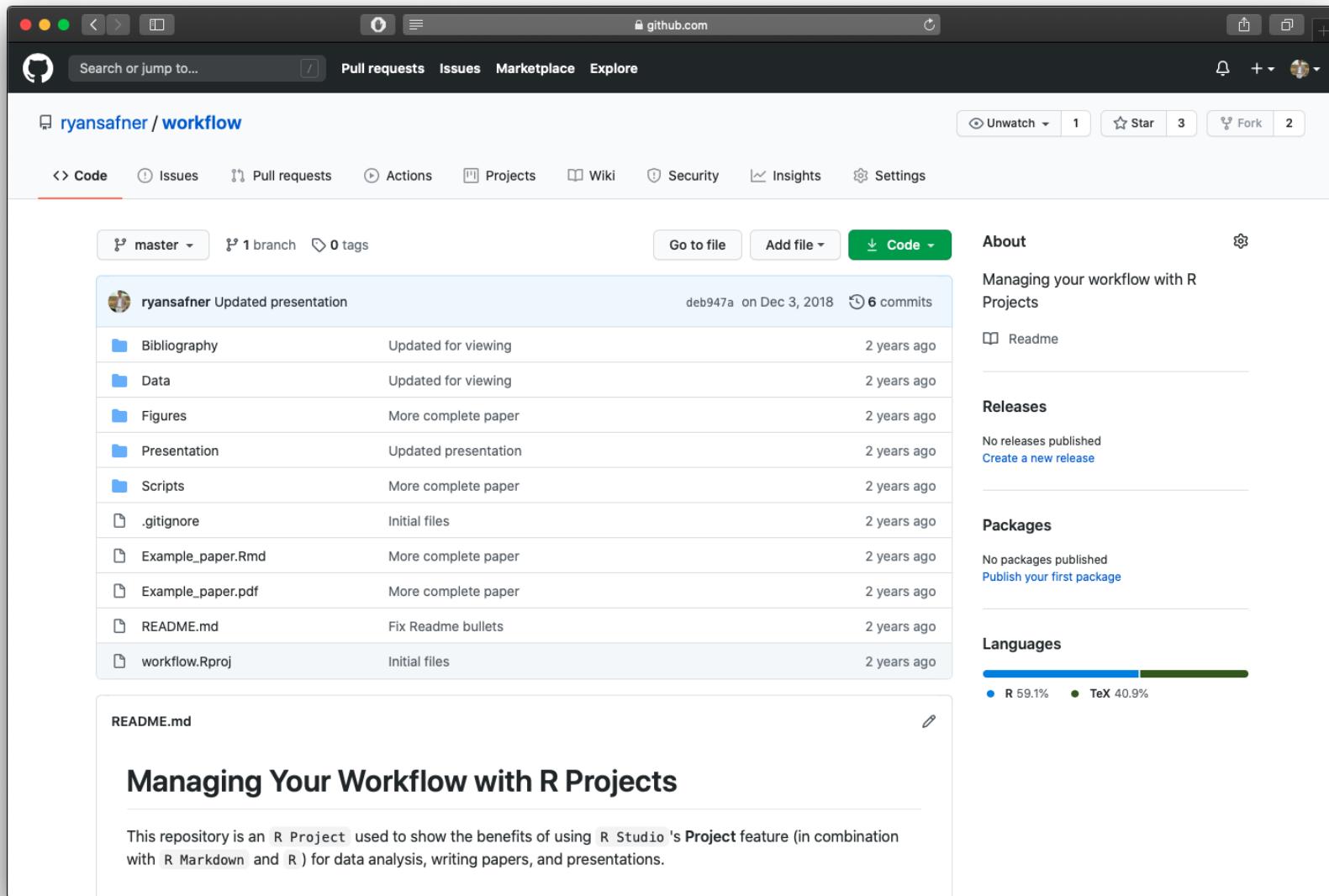


Projects

- At top right corner of RStudio
 - Click the button to the right of the name to open in a new window!



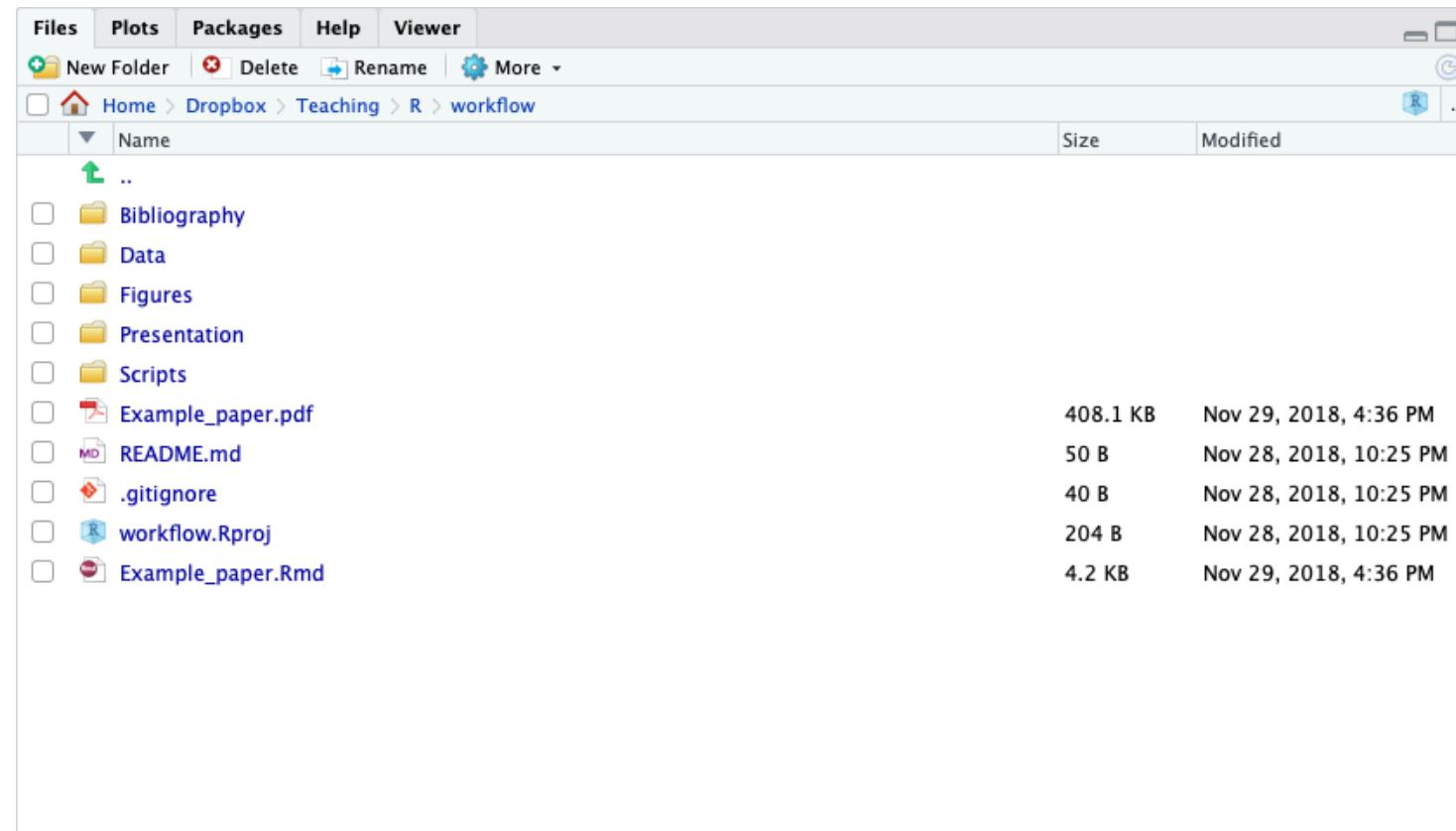
Loading Others' Projects



- This project is on **GitHub**, click the green button, download to your computer, open `.Rproj` file in R Studio
- See my **guide** about unzipping files (especially for Windows)!



A Good File Structure



- Look through this on your own
- Read the **README** of this repository on GitHub for instructions (automatically shows on the main page)
- Look at the **example_paper.qmd**
 - Uses data from **data** folder
 - Uses **.R** scripts from **scripts** folder
 - Uses figures from **figures** folder
 - Uses **bibexample.bib** from **bibliography** folder



Version Control

Have You Done This?



FINAL.doc!



FINAL_rev.2.doc



Source: PhD Comics



Have You Done This?



↑
FINAL_rev.6.COMMENTS.doc



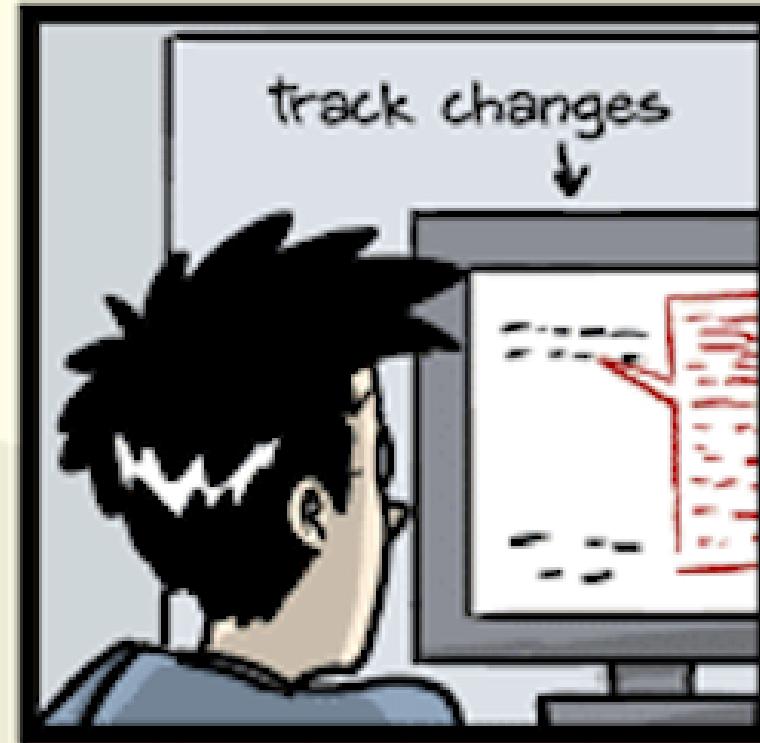
↑
FINAL_rev.8.comments5.
CORRECTIONS.doc

Source: PhD Comics



Have You Done This?

JORGE CHAM © 2012



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10.#@\$%WHYDID
ICOMETOGRADSCHOOL?????.doc



Source: PhD Comics



Do You Want to Be Able to

- Keep your files backed up
- Track changes
- Collaborate on the same files with others
- Edit files on one computer and then open and continue working on another?



The Training-Wheels Version

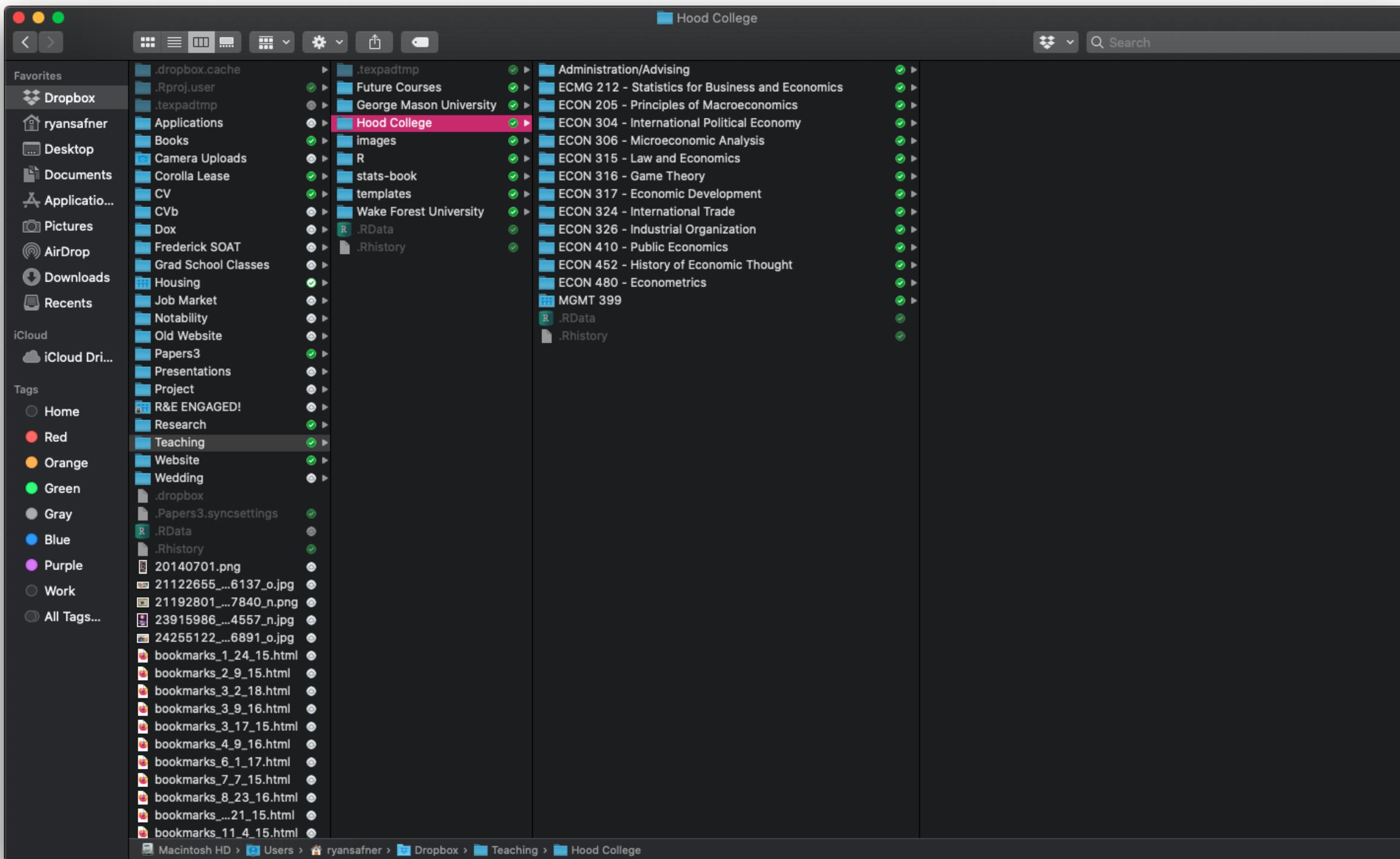


[Dropbox.com](https://www.dropbox.com)

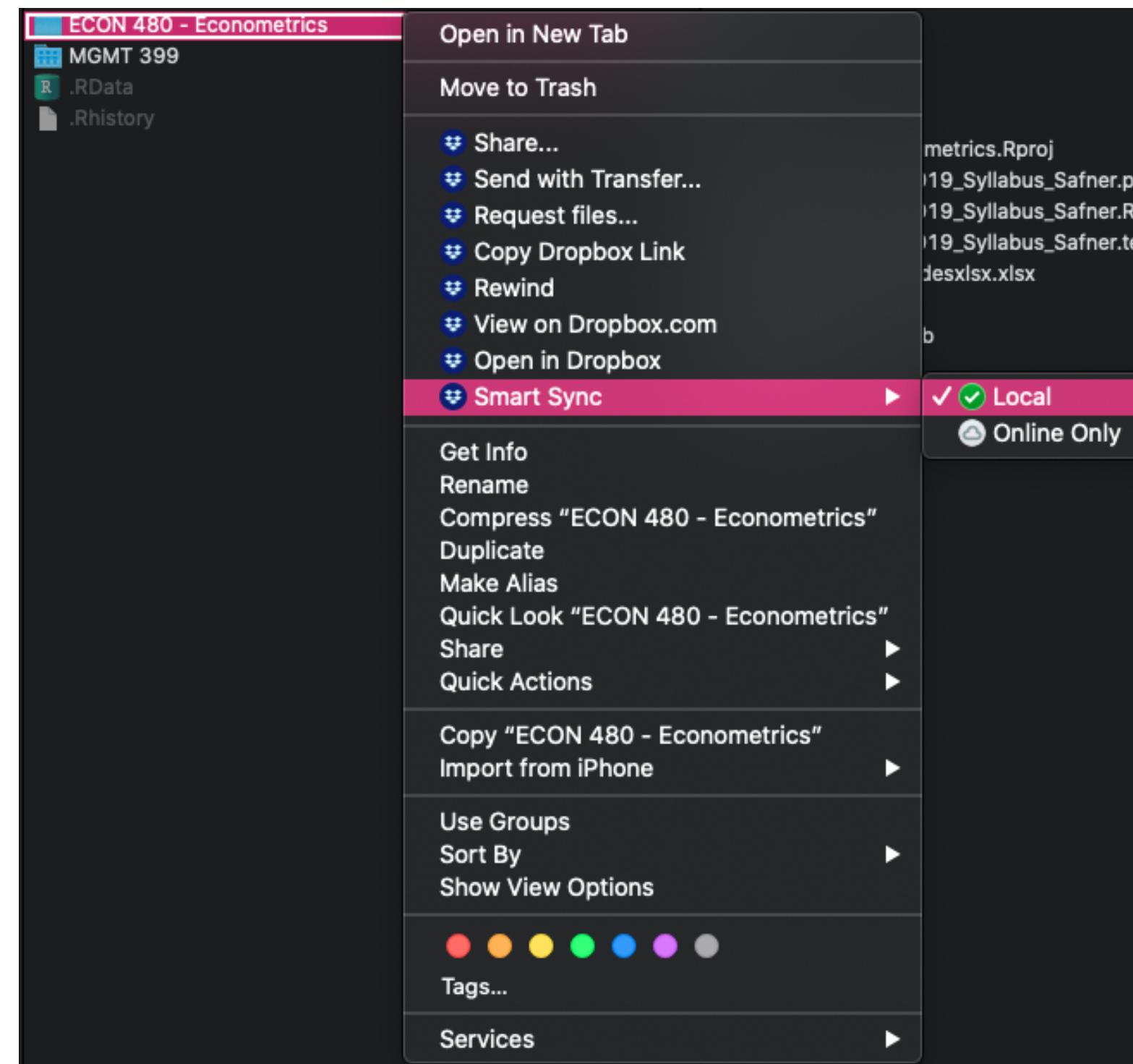
- Register an account for free
- Set up a location on your computer for the **Dropbox/** folder
- Anything you put in this folder will sync to the cloud
 - As soon as you change files, they *automatically* update and sync!
 - Can download any of these files from the *website* on any device
 - Set this up on multiple computers so when you change a file on one, it updates on all the others!



My Life Goes In Here



Smart Sync



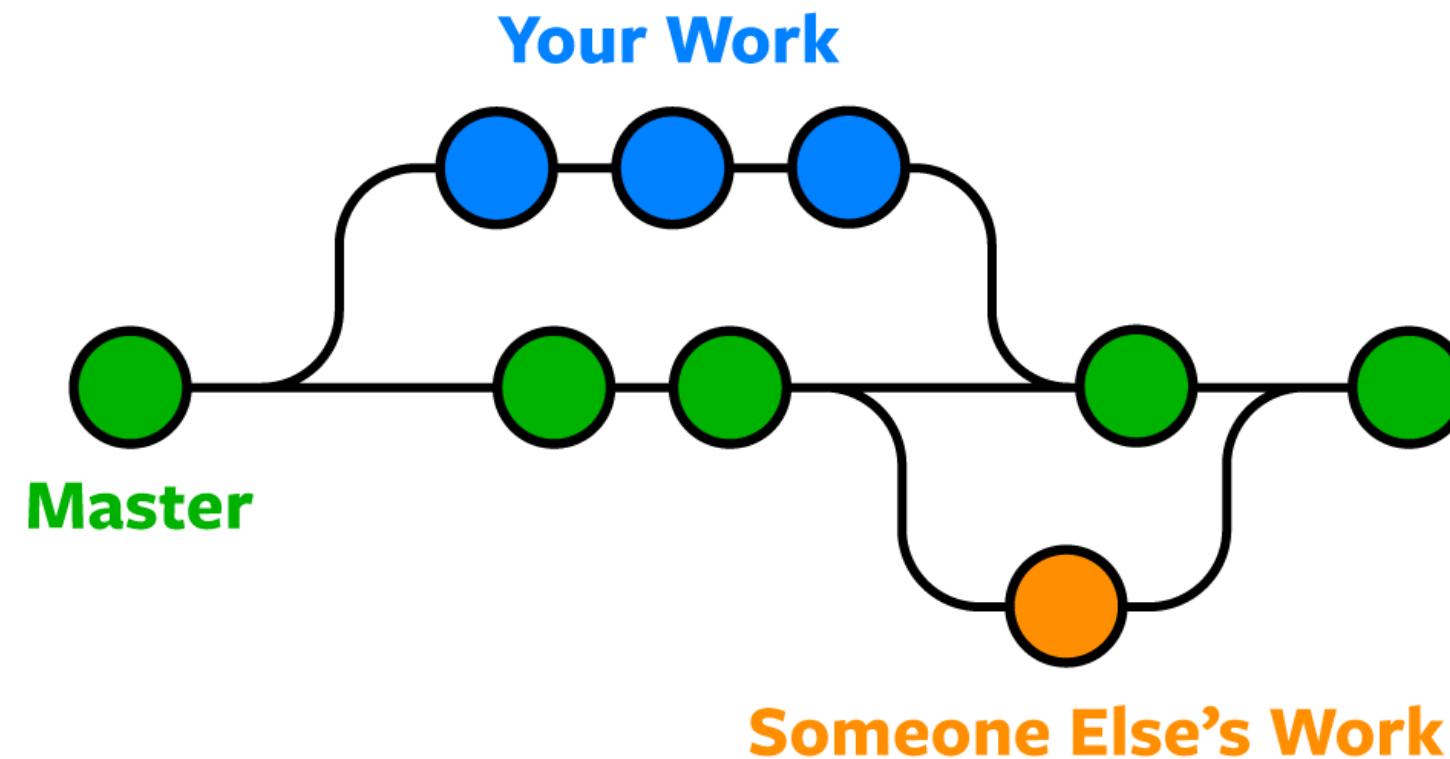
Smart Sync - keep some files online only for space



The Expert Version



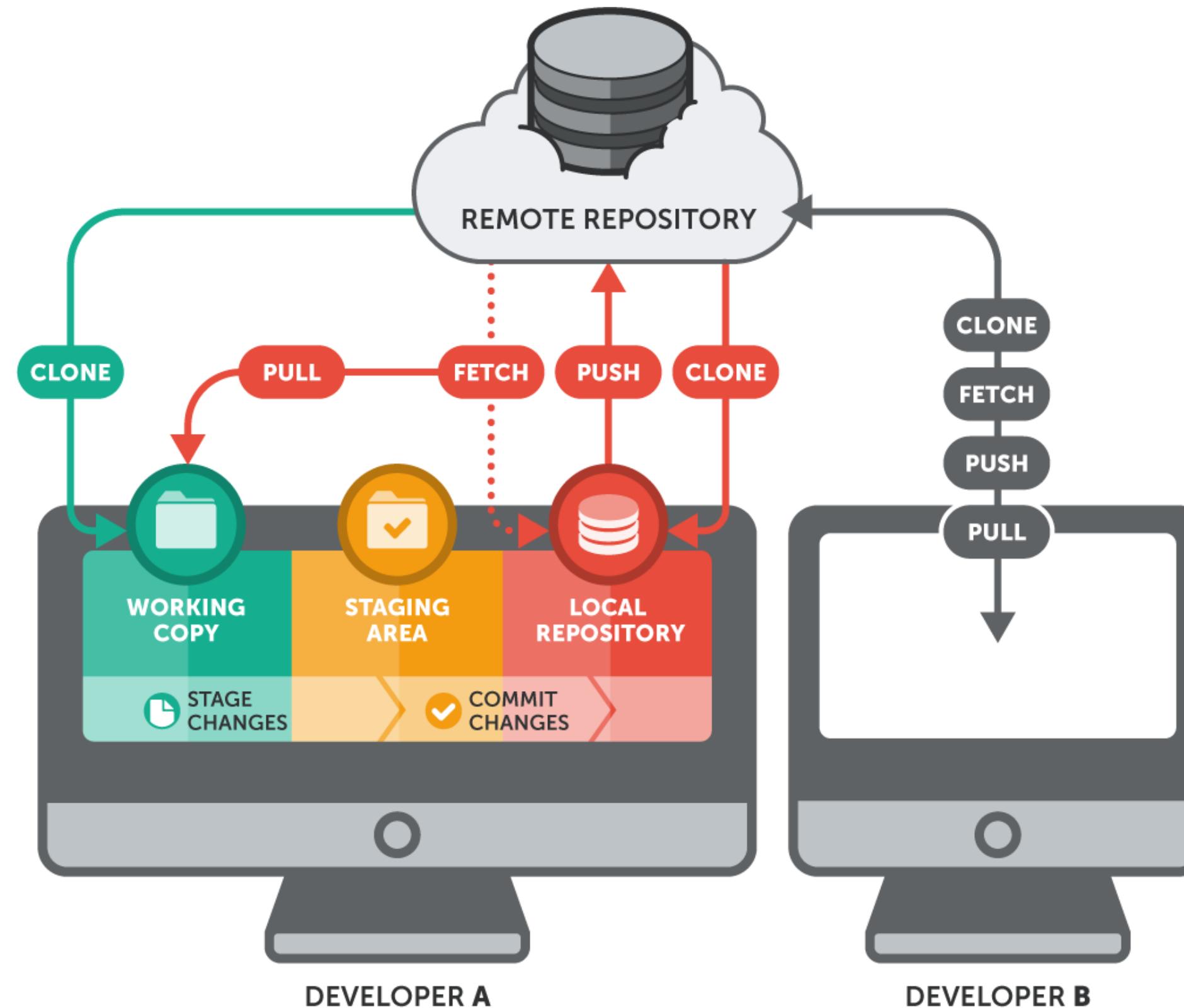
The Expert Version



- Shows history (**versions**) of files with comments
 - Can **fork** or **branch** repository into multiple versions at once
 - Good for “testing” things out without destroying old versions!
 - **revert** back to original versions as needed



The Expert Version



The Expert Version

- Requires *some* advanced set up, see [this excellent guide](#)
- R Studio integrates git and github commands nicely



This Class on Github

The screenshot shows the GitHub repository page for 'ryansafner / metricsF22'. The repository is public and has 1 branch and 0 tags. The main branch has 27 commits from 'ryansafner' dated 4 days ago. The repository description is 'Course website for ECON 480 - Econometrics, Fall 2022 semester'. It includes sections for About, Releases, Packages, and Languages. The Languages section shows a bar chart with HTML at 47.2%, JavaScript at 42.5%, CSS at 10.1%, SCSS at 0.2%, and TeX at 0.0%.

| File | Description | Last Commit |
|---------------------------------|-----------------------------|--------------|
| _extensions/quarto-ext/fonta... | We're committing to quarto! | last month |
| _freeze | Update for 1.1 | 21 days ago |
| _site | Upload 1.4 slide update | 4 days ago |
| assignments | Slides | 23 days ago |
| content | 1.4 | 5 days ago |
| data | 1.4 | 5 days ago |
| files | 1.4 R | 5 days ago |
| html | Update for 1.1 | 21 days ago |
| images | Updates before class | 20 days ago |
| r | 1.4 R | 5 days ago |
| resources | 1.2 | 19 days ago |
| slides | Upload 1.4 slide update | 4 days ago |
| .gitignore | We're committing to quarto! | last month |
| 404.qmd | Update for 1.1 | 21 days ago |
| LICENSE | Transfer from F21 | 2 months ago |
| README.md | Transfer from F21 | 2 months ago |
| _quarto.yml | 1.4 | 5 days ago |
| variables.vml | Render slides may be wrong | 23 days ago |

The screenshot shows the GitHub repository page for 'ryansafner / metricsF22'. The repository is public and has 1 branch and 0 tags. The main branch has 27 commits from 'ryansafner' dated 4 days ago. The repository description is 'Course website for ECON 480 - Econometrics, Fall 2022 semester'. It includes sections for About, Releases, Packages, and Languages. The Languages section shows a bar chart with HTML at 47.2%, JavaScript at 42.5%, CSS at 10.1%, SCSS at 0.2%, and TeX at 0.0%.

| File | Description | Last Commit |
|---------------------------------|-----------------------------|--------------|
| _extensions/quarto-ext/fonta... | We're committing to quarto! | last month |
| _freeze | Update for 1.1 | 21 days ago |
| _site | Upload 1.4 slide update | 4 days ago |
| assignments | Slides | 23 days ago |
| content | 1.4 | 5 days ago |
| data | 1.4 | 5 days ago |
| files | 1.4 R | 5 days ago |
| html | Update for 1.1 | 21 days ago |
| images | Updates before class | 20 days ago |
| r | 1.4 R | 5 days ago |
| resources | 1.2 | 19 days ago |
| slides | Upload 1.4 slide update | 4 days ago |
| .gitignore | We're committing to quarto! | last month |
| 404.qmd | Update for 1.1 | 21 days ago |
| LICENSE | Transfer from F21 | 2 months ago |
| README.md | Transfer from F21 | 2 months ago |
| _quarto.yml | 1.4 | 5 days ago |
| variables.vml | Render slides may be wrong | 23 days ago |

github.com/ryansafner/metricsF22



Most Packages Start on Github

Easily install and load packages from the tidyverse <https://tidyverse.tidyverse.org>

[r](#) [data-science](#) [tidyverse](#)

| Branch: master | New pull request | Create new file | Upload files | Find File | Clone or download |
|------------------------|---|-----------------|--------------|-----------|-------------------|
| hadley Update paper.md | | | | | |
| .github | Add move bot config. | | | | |
| R | Only look at pacakges when considering conflicts | | | | |
| man | Add repos option to tidyverse_{deps,update} (#82) | | | | |
| pkgdown/favicon | Use retina logo | | | | |
| revdep | Re-run revdep checks | | | | |
| tests | Add first test | | | | |
| vignettes | Update paper.md | | | | |
| .Rbuildignore | Merge branch 'master' into joss-paper | | | | |
| .covignore | use_tidy_ci() | | | | |
| .gitignore | Remove built website | | | | |
| .travis.yml | use_tidy_ci() | | | | |
| DESCRIPTION | Update for new R version | | | | |
| LICENSE | Include GPL license text | | | | |
| NAMESPACE | | | | | |

github.com/tidyverse/tidyverse

Excerpt from the Gapminder data, as an R data package and in plain text delimited form

[r](#) [data-science](#) [gapminder](#)

| Branch: master | New pull request | Create new file | Upload files | Find File | Clone or download |
|--|---|-----------------|--------------|-----------|-------------------|
| jennybc Add a test so country_codes stay fixed | Latest commit ddfe773 on Mar 6 | | | | |
| .aspell | Use .aspell | 2 years ago | | | |
| R | Document country_codes | 2 years ago | | | |
| dataRaw | Be careful about preserving the int type of iso_num | 6 months ago | | | |
| data | Be careful about preserving the int type of iso_num | 6 months ago | | | |
| inst/extdata | Fix country codes for North Korea | last year | | | |
| man | Be careful about preserving the int type of iso_num | 6 months ago | | | |
| tests | Add a test so country_codes stay fixed | 6 months ago | | | |
| .Rbuildignore | correct area in bubble plot; fixes #7 | 4 years ago | | | |
| .gitignore | "compile notebook" for cleaning scripts; fixes #1 | 4 years ago | | | |
| DESCRIPTION | attempt to get hyperlinks recognized (#28) | 6 months ago | | | |
| NAMESPACE | import tibble for printing purposes | 3 years ago | | | |
| NEWS.md | Fix country codes for North Korea | last year | | | |
| README.Rmd | attempt to get hyperlinks recognized (#28) | 6 months ago | | | |
| README.md | Be careful about preserving the int type of iso_num | 6 months ago | | | |
| cran-comments.md | Update cran-comments | 2 years ago | | | |

github.com/jennybc/gapminder



My Workflow (That I Suggest to You)

1. Create a new repository on Github.¹
2. Start a New R Project in R Studio (link it to the github repository - [see guide](#))
3. Create a logical file system ([see example](#)), such as:

```
project # folder on my computer (the new working directory)
|
|- data/ # folder for data files
|- scripts/ # folder .R code
|- bibliography/ # folder for .bib files
|- figures/ # folder to plots and figures to
|- paper.qmd # write document here
```

4. Write document in [paper.qmd](#), loading/saving files from/to various folders in project
 - e.g. load data like `df <- read_csv("data/my_data")`; save plots like `ggsave("figures/p.png")`
5. Render document to [pdf](#) or [html](#).
6. Occasionally, [stage](#) and [commit](#) changes with a description, [push](#) to GitHub.



Resources

- Quarto Documentation: [Tutorial: Hello, Quarto](#)
- Quarto Documentation: [Tutorial: Computation](#)
- Quarto Documentation: [Tutorial: Authoring](#)
- Quarto Documentation: [Guide](#)
- Kieran Healey's [The Plain Person's Guide to Plain Text Social Science](#) on managing workflow with plain text files, R, and Git
- Hadley Wickham's (and Garrett Grolemund) [R for Data Science](#) on how to use R and R Markdown for data science work
- Jenny Bryan's [Happy Git with R](#) on how to use git and GitHub with R as a version control system

