

# 1.5 — Optimize Workflow

## ECON 480 • Econometrics • Fall 2022

Dr. Ryan Safner

Associate Professor of Economics

[safner@hood.edu](mailto:safner@hood.edu)

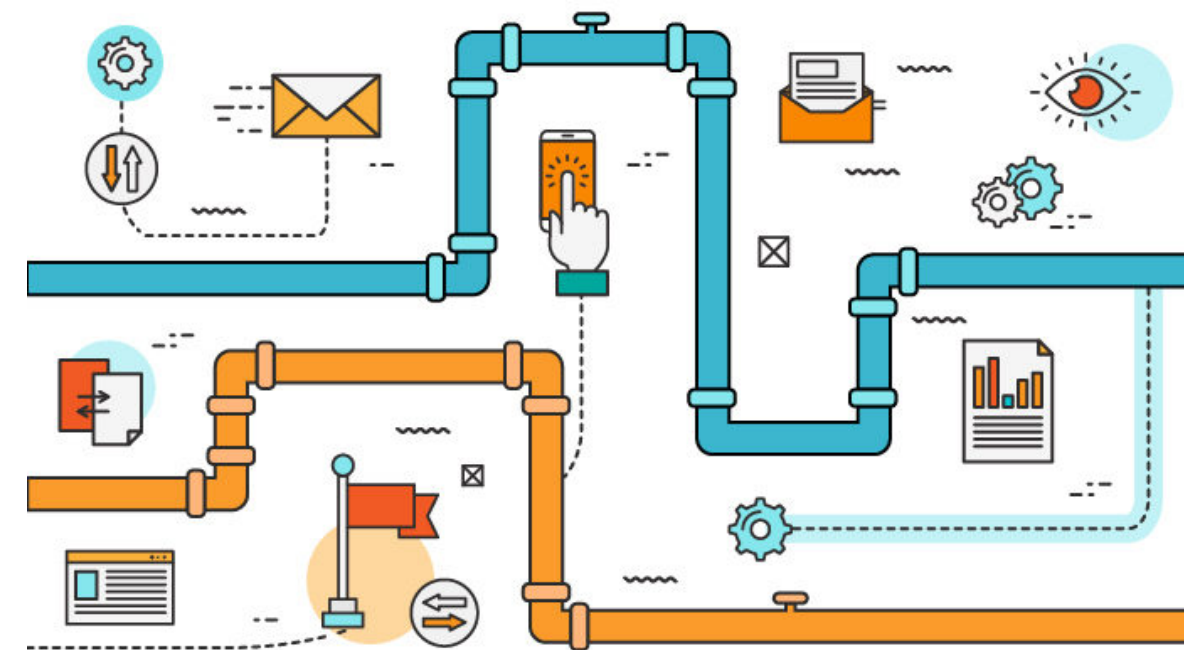
[ryansafner/metricsF22](https://ryansafner/metricsF22)

[metricsF22.classes.ryansafner.com](https://metricsF22.classes.ryansafner.com)



# Your Workflow Has a Lot of Moving Parts

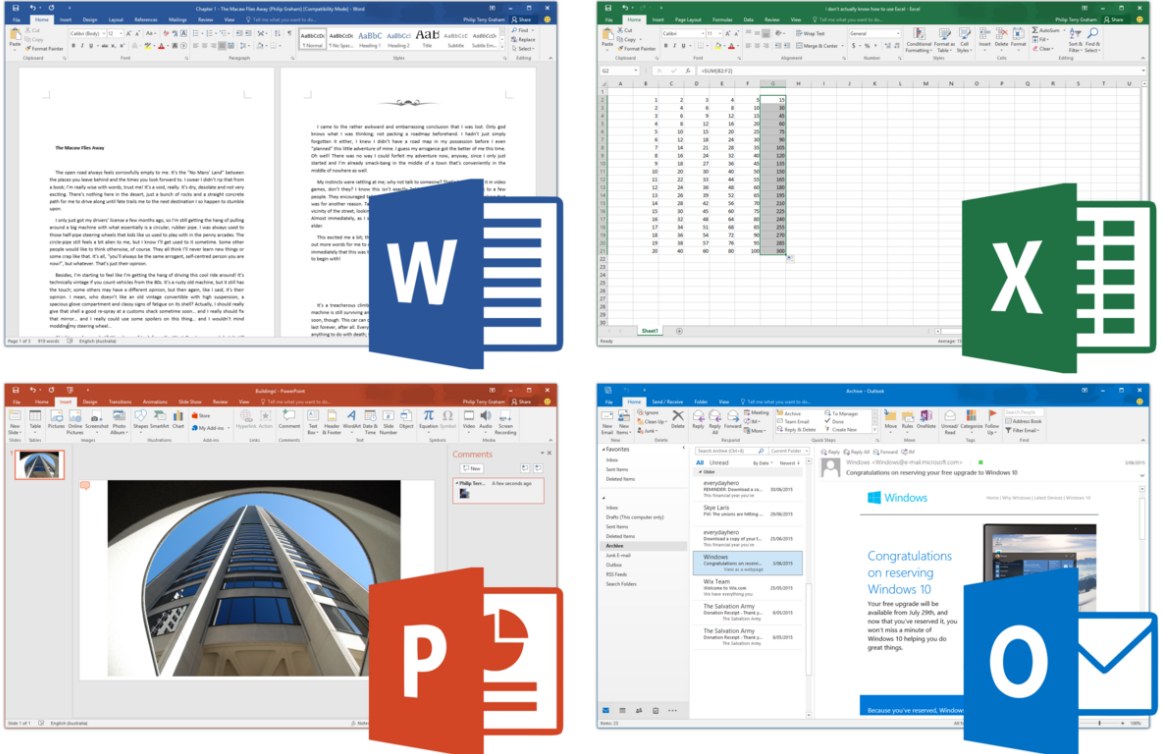
1. Writing text/documents
2. Managing citations and bibliographies
3. Performing data analysis
4. Making figures and tables
5. Saving files for future use
6. Monitoring changes in documents
7. Collaborating and sharing with others
8. Combining into a deliverable (report, paper, presentation, etc.)



# 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:

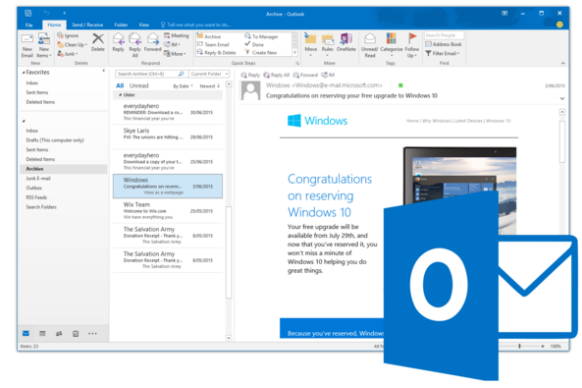
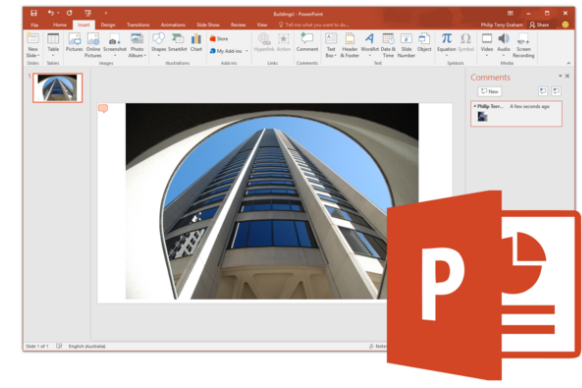
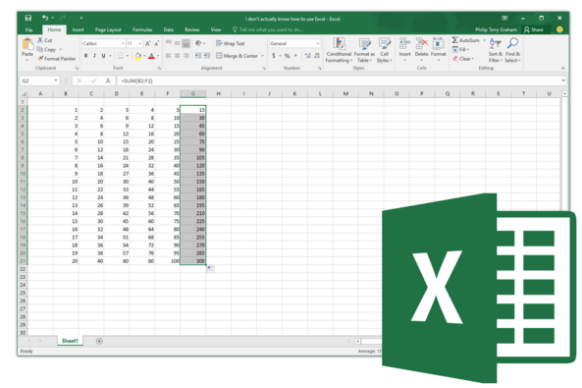
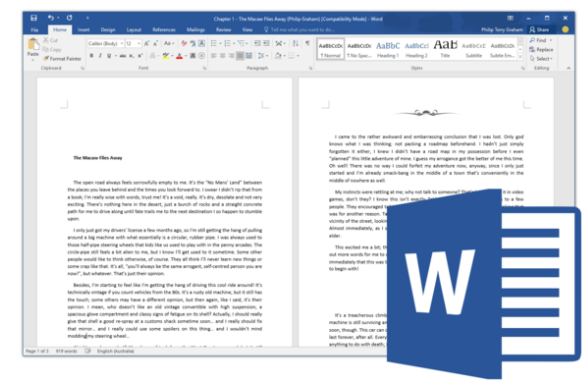
## Moving a picture in Microsoft Word



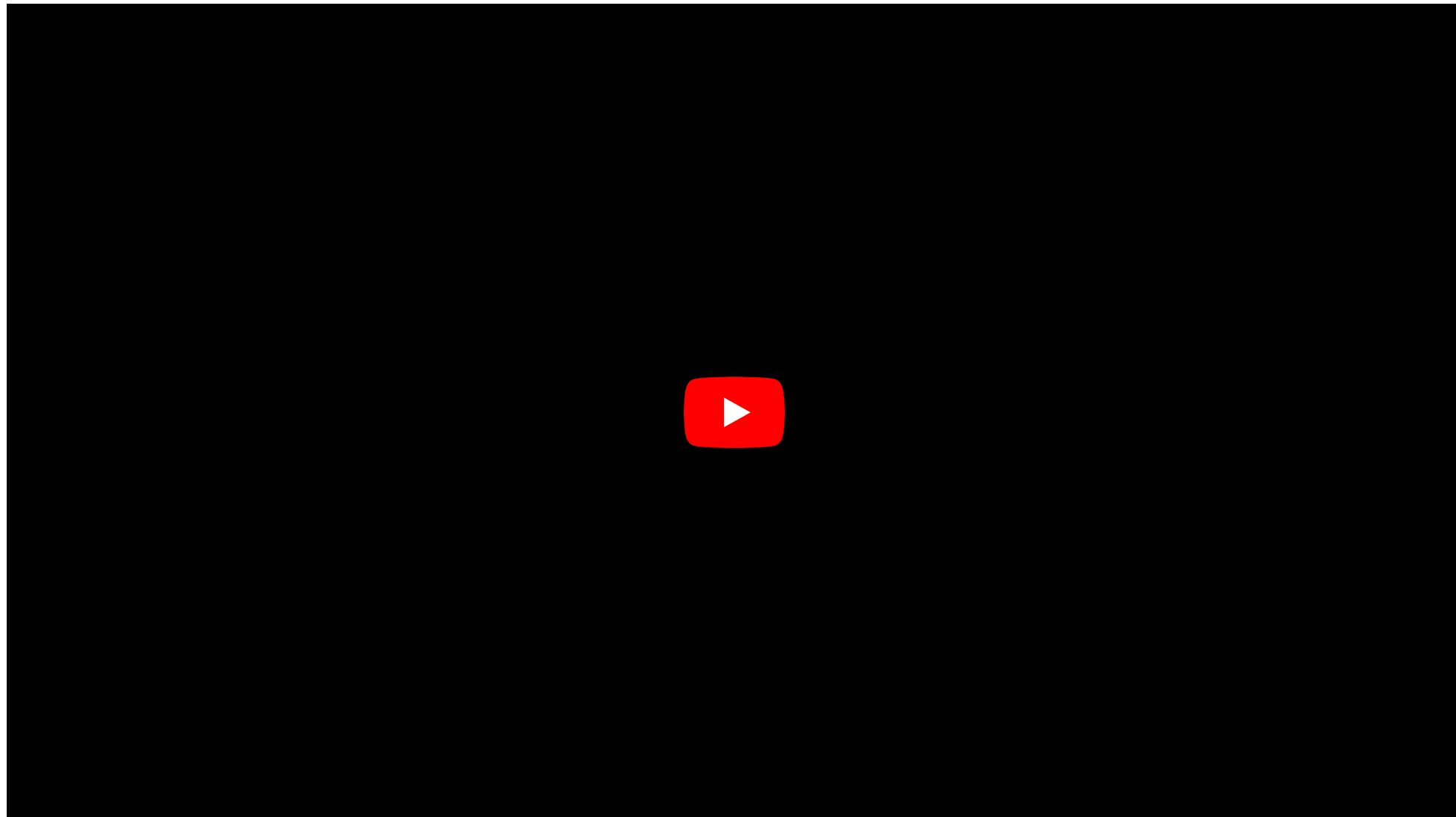
■ You  
the

■ It  
actually  
does  
what  
you  
want

mess up  
whole document



# The Office Model: A Short Horror Film



# The Office Model: Mistakes

sciencemag.org

AAAS Become a Member Log In ScienceMag.org Search

Science Contents News Careers Journals

164 49

expression site soma germ line soma germ line

C. sp. 9	JU1325	-	-	-	-
C. briggsae	AF18	-	-	-	+++
C. sp. 5	JU727	-	-	-	+++
C. remanei	PB4641	-	-	-	+++
C. sp. 10	JU1333	-	-	-	-
C. brenneri	PB2801	-	-	-	-
C. sp. 11	JU1373	-	-	-	+++
C. sp. 16	JU1873	+	-	-	-
C. elegans	N2	++	+++	+++	+++
C. elegans	CB4856	-	+	-	-
C. elegans	JU1580	-	-	-	+++
C. sp. 15	QG122	+++	-	+++	-
C. sp. 19	EG6142	-	-	-	-
C. sp. 17	JU1825	-	-	-	-
C. sp. 18	JU1857	-	-	-	-
C. sp. 14	EG6716	+++	-	-	-
C. sp. 7	JU1199	+++	-	-	-
C. sp. 20	NIC113	+	-	-	-
C. sp. 6	EG4788	+++	-	-	-
C. sp. 13	JU1528	+++	-	-	-
C. sp. 2	DF5070	-	-	-	+
C. drosophila	DF5077	-	-	-	+
C. angaria	RGD1	-	+++	+	++
C. sp. 12	JU1428	-	-	-	-
C. sp. 8	QX1162	-	-	-	-
C. plicata	SB355	+/	-	+/	+++
C. sp. 1	SB341	-	+++	-	+++

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

Source: Science Magazine

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

Menu Search Bloomberg Businessweek Sign In Subscribe

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 >  
Listen to Live Radio >


Bloomberg Television

PHOTOGRAPH BY GREGOR SCHUSTER

Source: Bloomberg



# The Office Model: Not Reproducible



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

""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

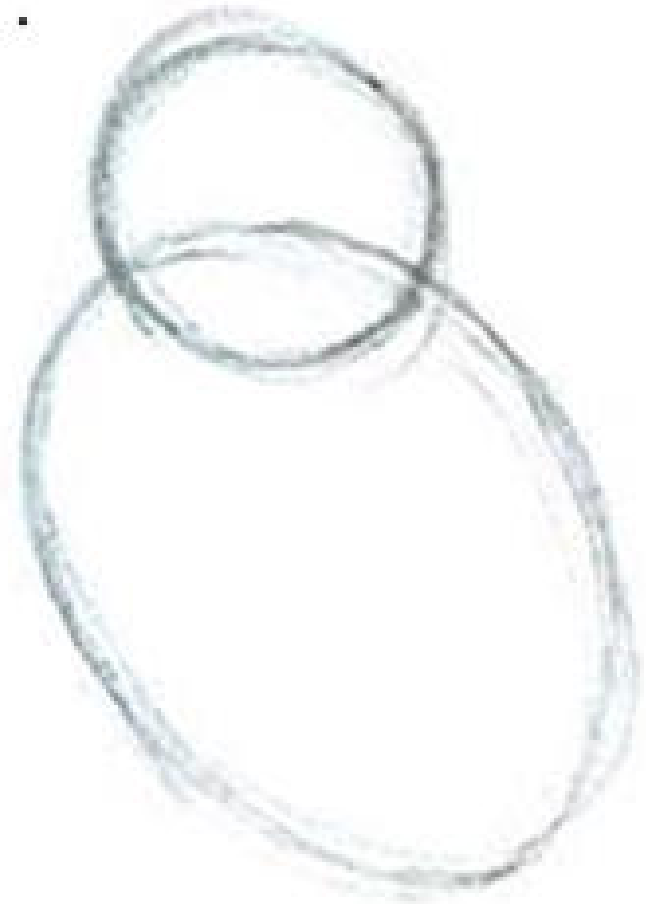
[Read 4 replies](#)



# Drawing the Rest of the Owl

How to draw an owl

1.



1. Draw some circles

2.

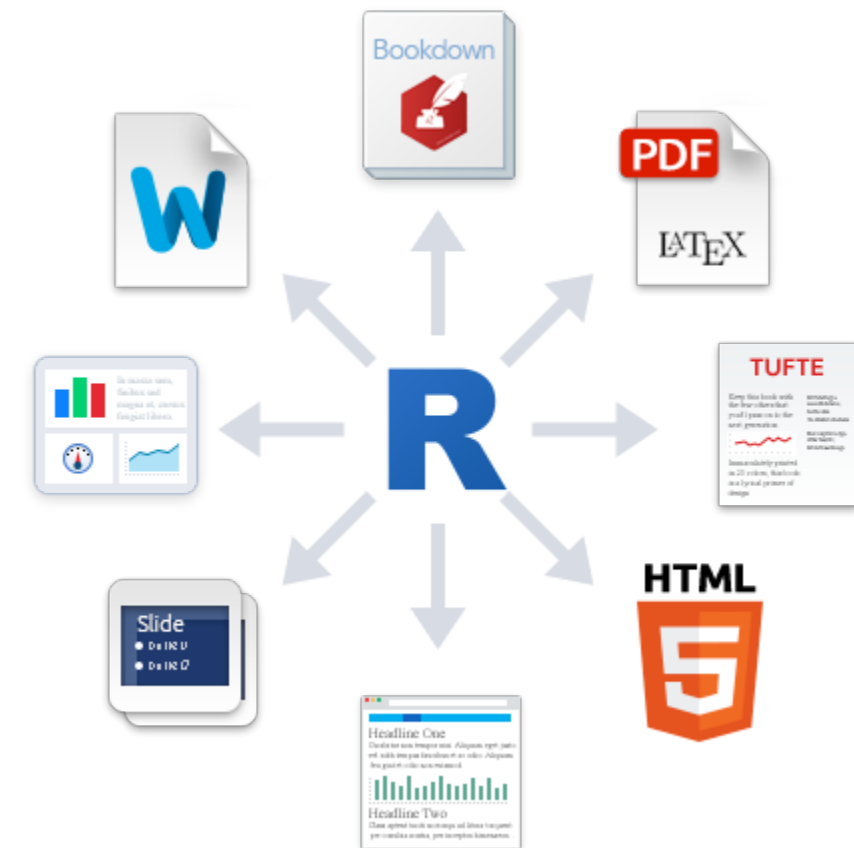


2. Draw the rest of the fucking owl



# 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<sup>®</sup> 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

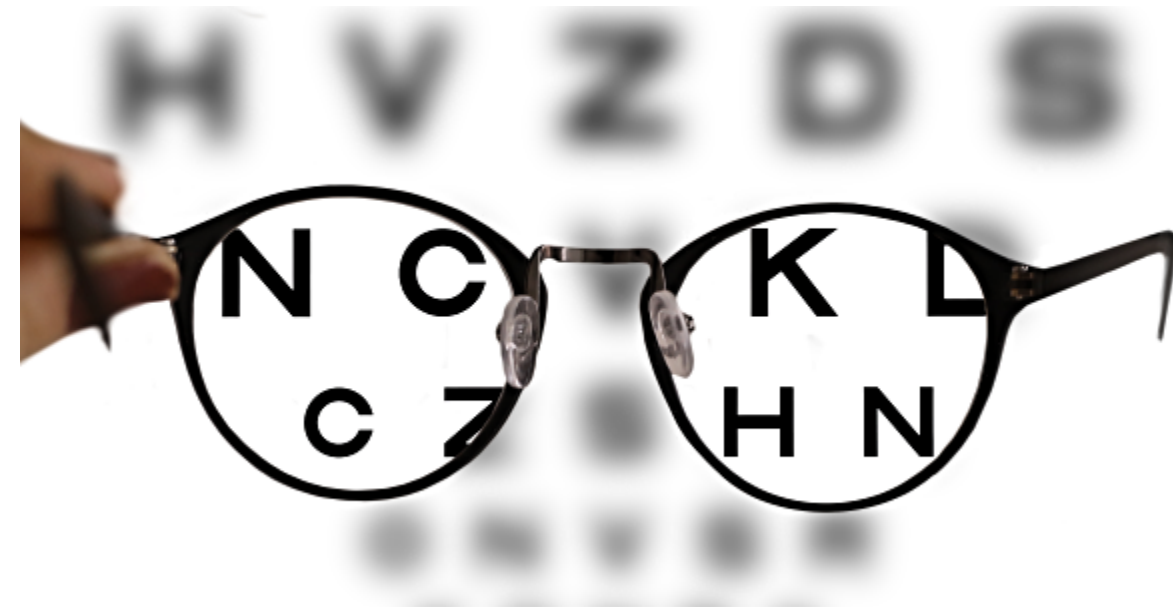
[Quarto.org](#)





# 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

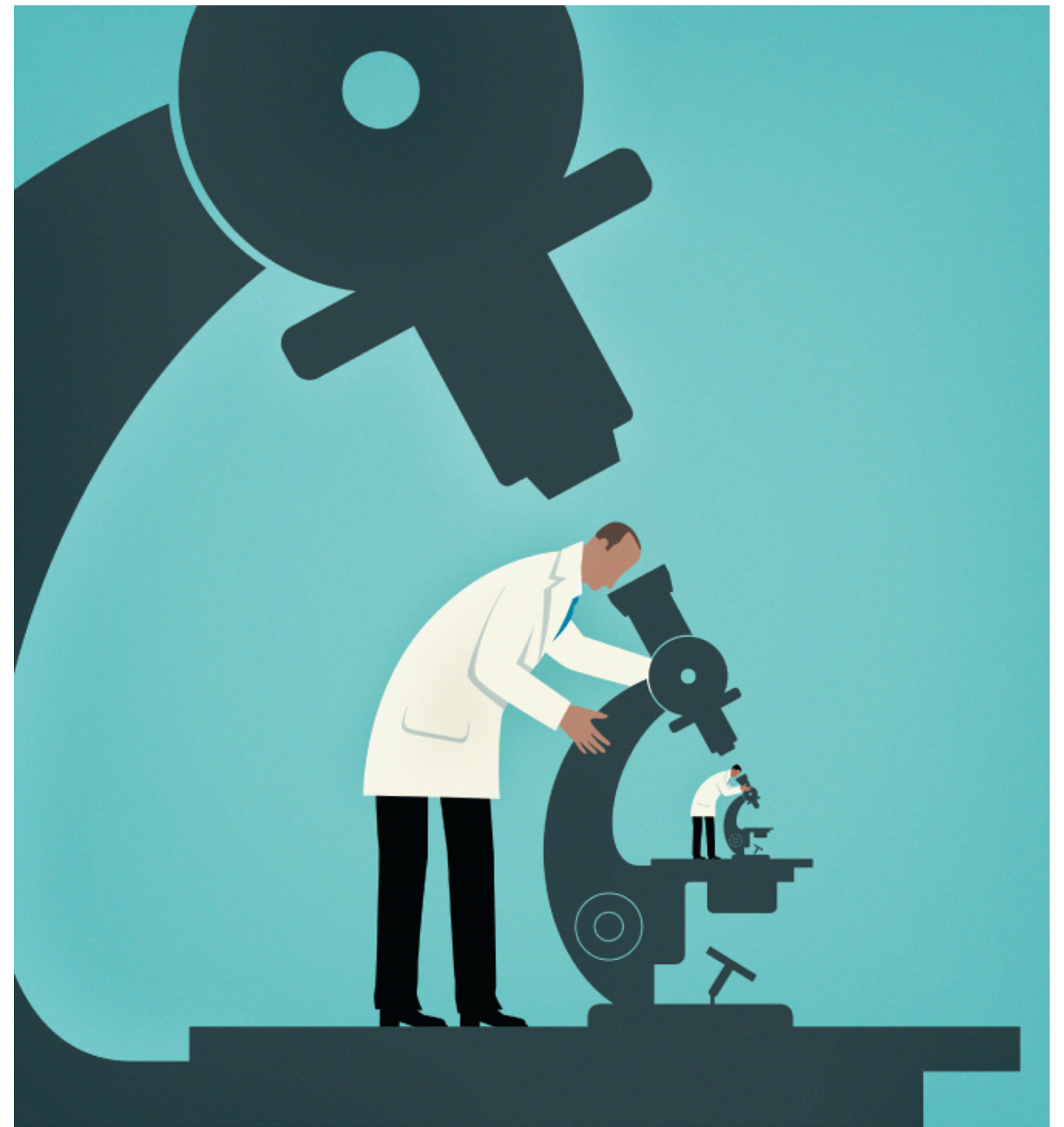
- **Open Source:** free, useable forever, often very small file size
  - Proprietary software is a gamble - can you still open a `.doc` file from Microsoft Word 1997?
- **Automate and Minimize Errors**, especially in repetitive processes
- Can be used with **version control** (see below)



# 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

theatlantic.com

Subscribe Sign In

Popular Latest Sections Magazine

**The Scientific Paper is Obsolete**  
Here's what's next.

JAMES SOMERS | APR 5, 2018 | SCIENCE

Share Tweet

TEXT SIZE - +

Enjoy unlimited access to The Atlantic for less than \$1 per week. [Sign in](#) [Subscribe Now](#)

Source: [The Atlantic](#)

paulromer.net

Paul Romer

home about archive search

**Jupyter, Mathematica, and the Future of the Research Paper**  
*April 13, 2018*

The Atlantic has a great [article](#) on new ways to share research results. Its three parts make three points:

1. A graphical user interface (GUI) can facilitate better technical writing.
2. Wolfram's proprietary notebook showcased innovative technology, but decades after its introduction, still has few users.
3. Jupyter is a new open-source alternative that is well on the way to becoming a standard for exchanging research results.

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

TABLE OF CONTENTS

- 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

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

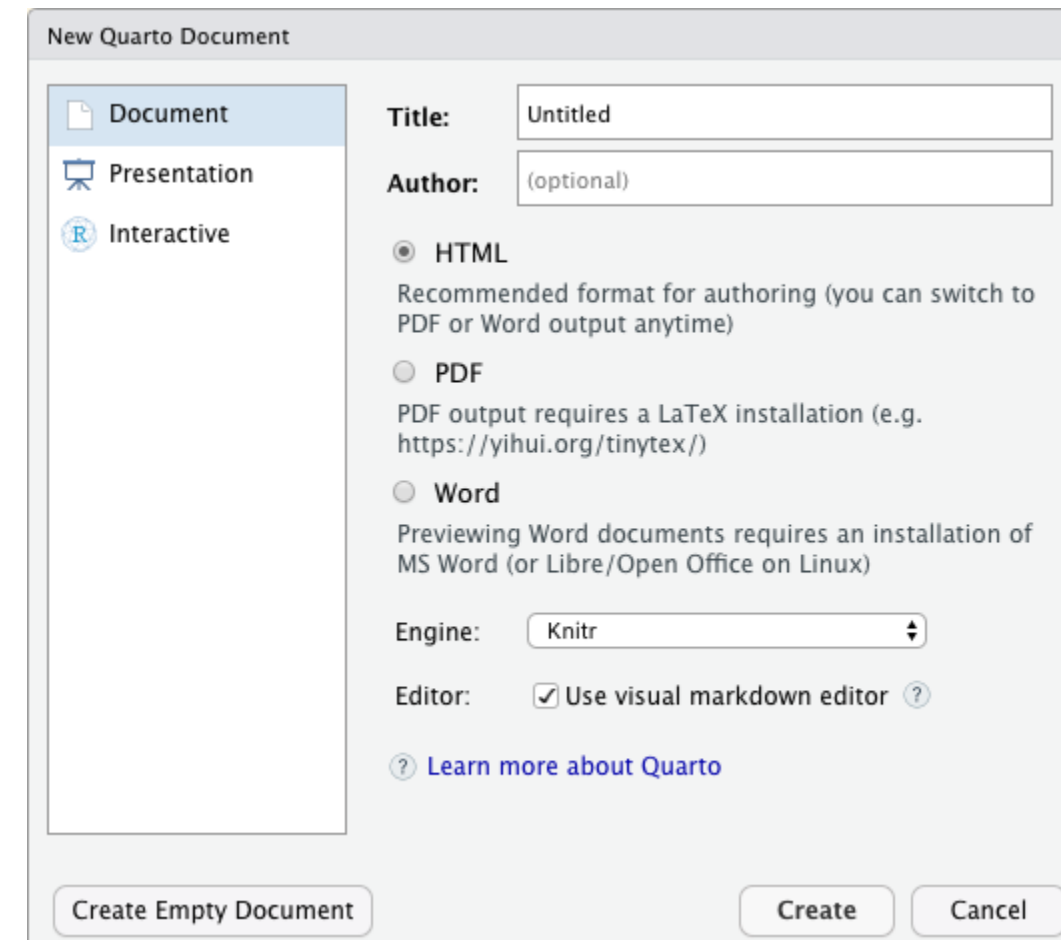


# Quarto

# Creating a Quarto Document I

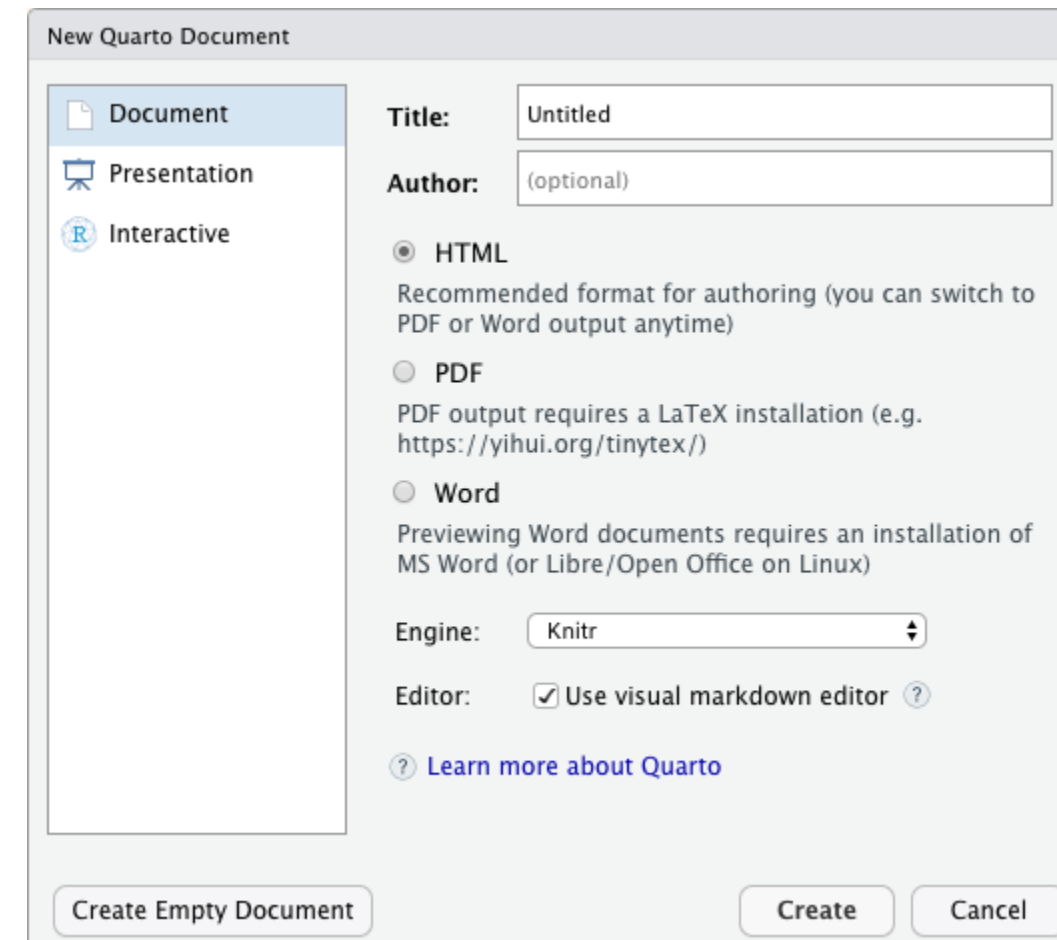
File → New File → Quarto Document...

- 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 a Quarto document editor window titled 'Untitled1'. The editor is in 'Visual' mode. The source code is displayed in a text area, showing a YAML header and two R code chunks. The rendered output is shown below the source code, including a title, a section header, and the output of the R code chunks.

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

### Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

### 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**<sup>1</sup> 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



# 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](https://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 developm
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**<sup>1</sup> above and below your code
- After the first pair of backticks, signify the **language** of the code<sup>2</sup> 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

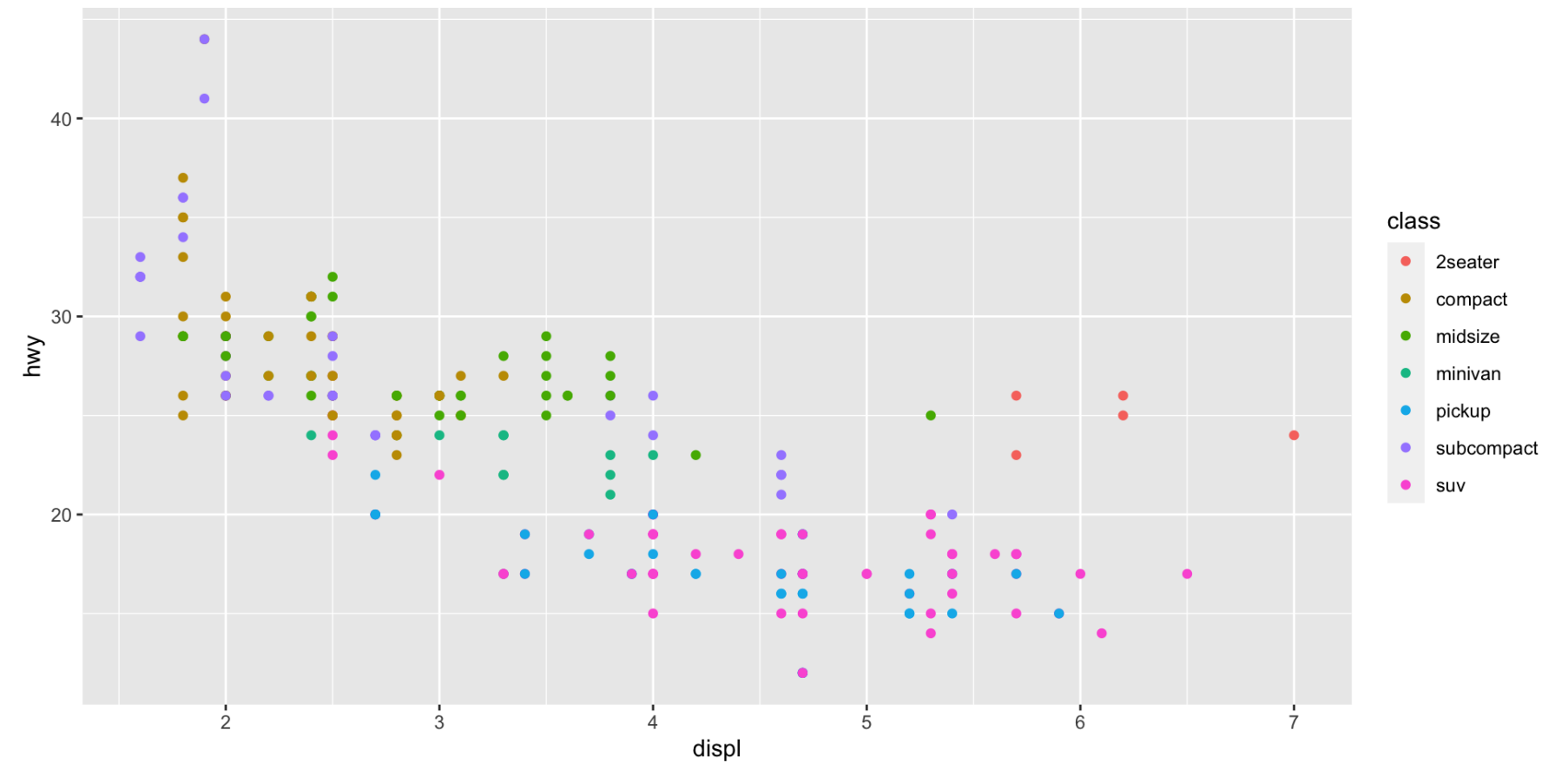
## Input

```

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**`

*italics* and **bold**

`superscript^2^` / `subscript~2~`

superscript<sup>2</sup> / subscript<sub>2</sub>

`~~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
<pre>* unordered list   + sub-item 1   + sub-item 2   - sub-sub-item 1</pre>	<ul style="list-style-type: none"> <li>• unordered list       <ul style="list-style-type: none"> <li>▪ sub-item 1</li> <li>▪ sub-item 2           <ul style="list-style-type: none"> <li>○ sub-sub-item 1</li> </ul> </li> </ul> </li> </ul>
<pre>* item 2  Continued (indent 4 spaces)</pre>	<ul style="list-style-type: none"> <li>• item 2       <ul style="list-style-type: none"> <li>Continued (indent 4 spaces)</li> </ul> </li> </ul>
<pre>1. ordered list 2. item 2    i) sub-item 1      A. sub-sub-item 1</pre>	<ol style="list-style-type: none"> <li>1. ordered list</li> <li>2. item 2       <ol style="list-style-type: none"> <li>1. sub-item 1           <ol style="list-style-type: none"> <li>1. sub-sub-item 1</li> </ol> </li> </ol> </li> </ol>



# Writing Text with Markdown: Headings

## Markdown Syntax

## Output

---

# Header 1

# Header 1

---

## Header 2

## Header 2

---

### Header 3

### Header 3

---

#### Header 4

#### Header 4

---

##### Header 5

##### Header 5

---

##### Header 6

###### 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/tidyverse1.png)
2
3 
```

# 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()
```

	<b>mpg</b>	<b>cyl</b>	<b>disp</b>	<b>hp</b>	<b>drat</b>	<b>wt</b>	<b>qsec</b>	<b>vs</b>	<b>am</b>	<b>gear</b>	<b>carb</b>
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()
```

	<b>mpg</b>	<b>cyl</b>	<b>disp</b>	<b>hp</b>	<b>drat</b>	<b>wt</b>	<b>qsec</b>	<b>vs</b>	<b>am</b>	<b>gear</b>	<b>carb</b>
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()
```

	<b>mpg</b> <dbl>	<b>cyl</b> <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  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , used by mathematicians, economists, and others to write papers and slides with perfect math ~~and~~ formatting
  - I used to use for everything before I found **R** and **markdown**
  - Producing **pdf** output actually converts **markdown** files into first!  $\text{T}_{\text{E}}\text{X}$
  - 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	Output
<code>\alpha</code>	
<code>\pi</code>	$\alpha$
<code>\frac{1}{2}</code>	$\pi$
<code>\hat{x}</code>	
<code>\bar{y}</code>	$\hat{x}$
<code>x_{1,2}</code>	$\bar{y}$
<code>x^{a-1}</code>	$x_{1,2}$
<code>\lim_{x \to \infty}</code>	$x^{a-1}$
<code>A=\begin{bmatrix} a_{1,1} &amp; a_{1,2} \\ a_{2,1} &amp; a_{2,2} \end{bmatrix}</code>	$\lim_{x \rightarrow \infty}$
	$A = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$







# 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  
6           and the Opportunity of the Commons},  
7   volume = {12},  
8   number = {4},  
9   pages = {743-771}  
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



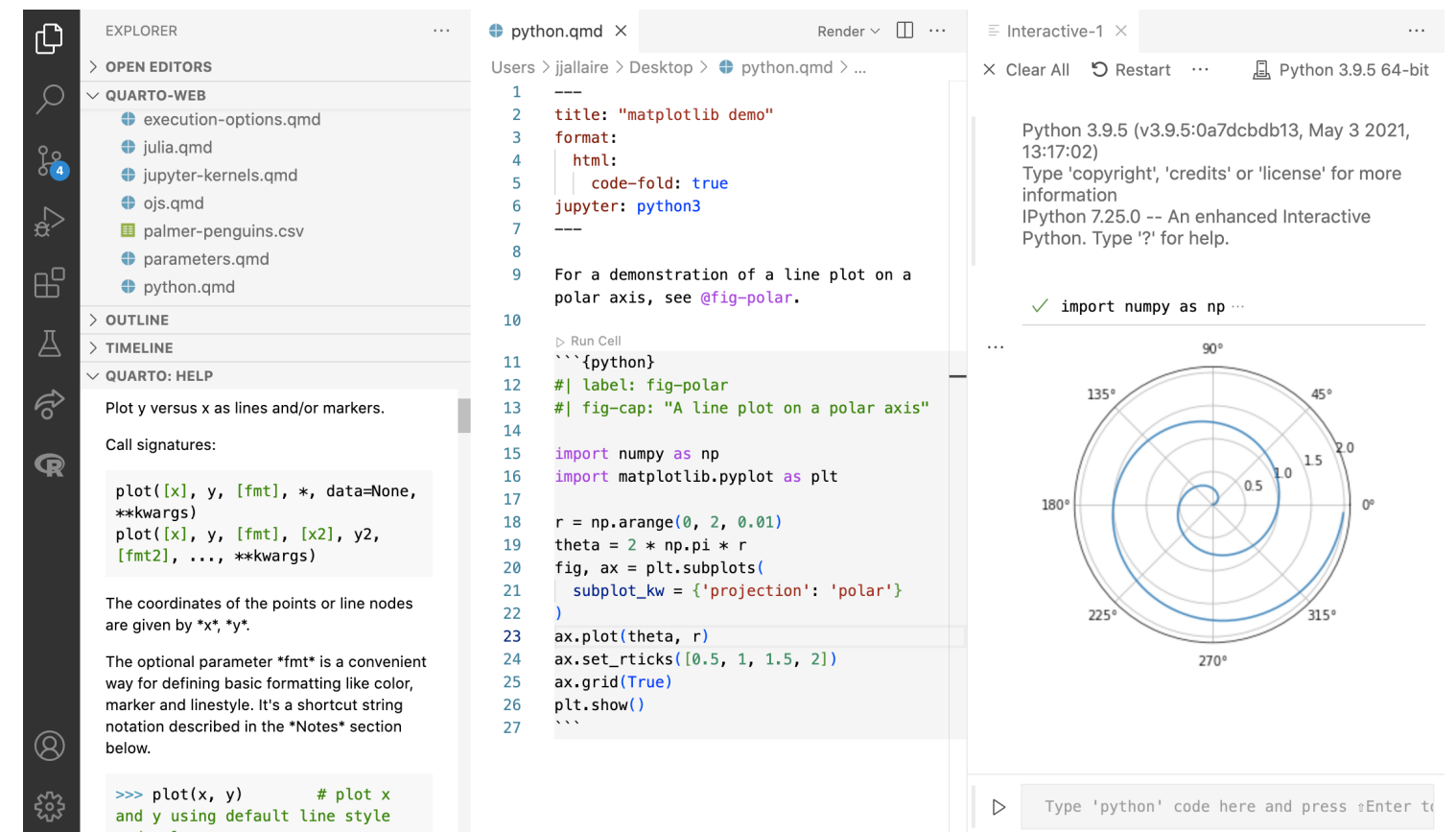
# Reference Management Software

- 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**

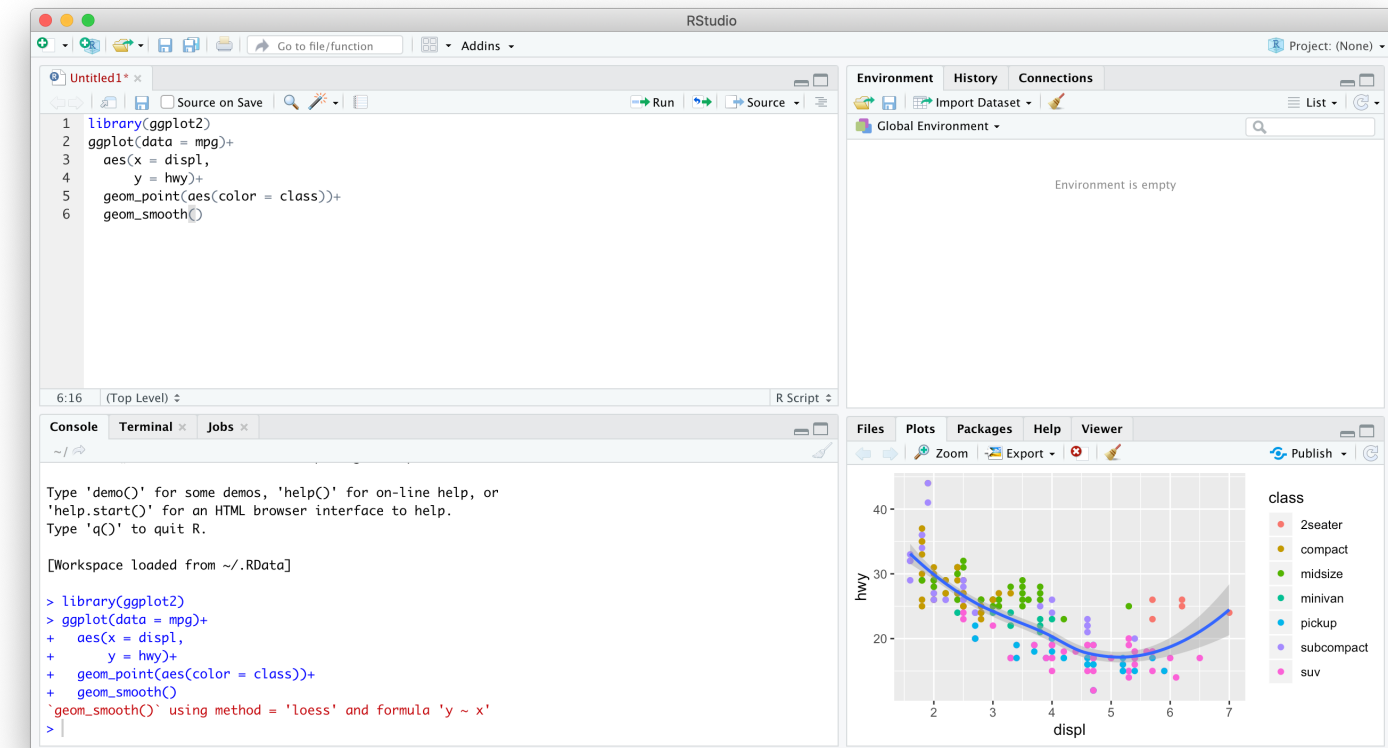


VS Code



# 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](#)

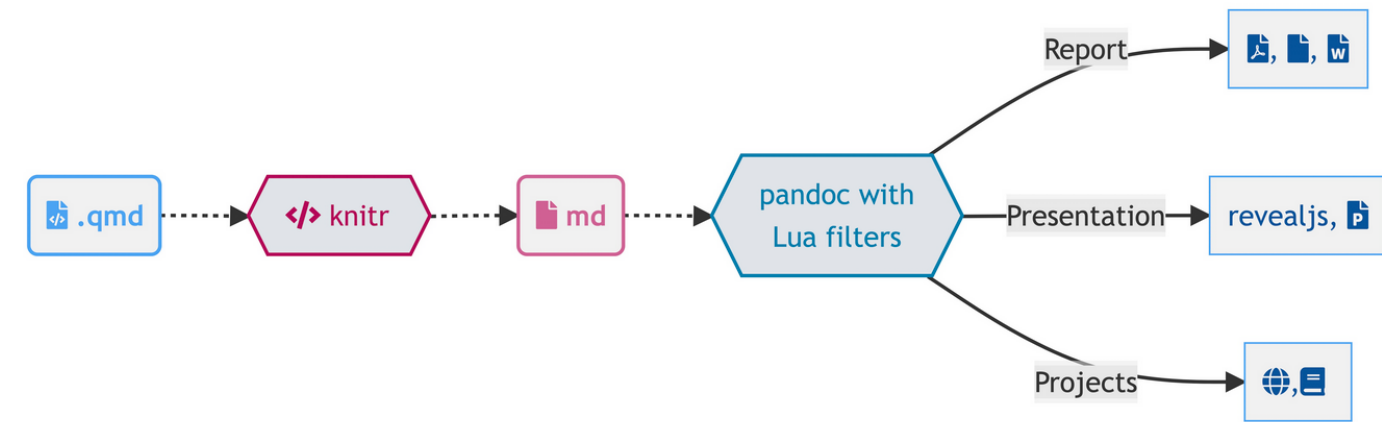




# Rendering Your Documents

# knitr

- When you are ready, you “render” your markdown and code into an output format using:
- `knitr`<sup>1</sup>, 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!



# PDF Output

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

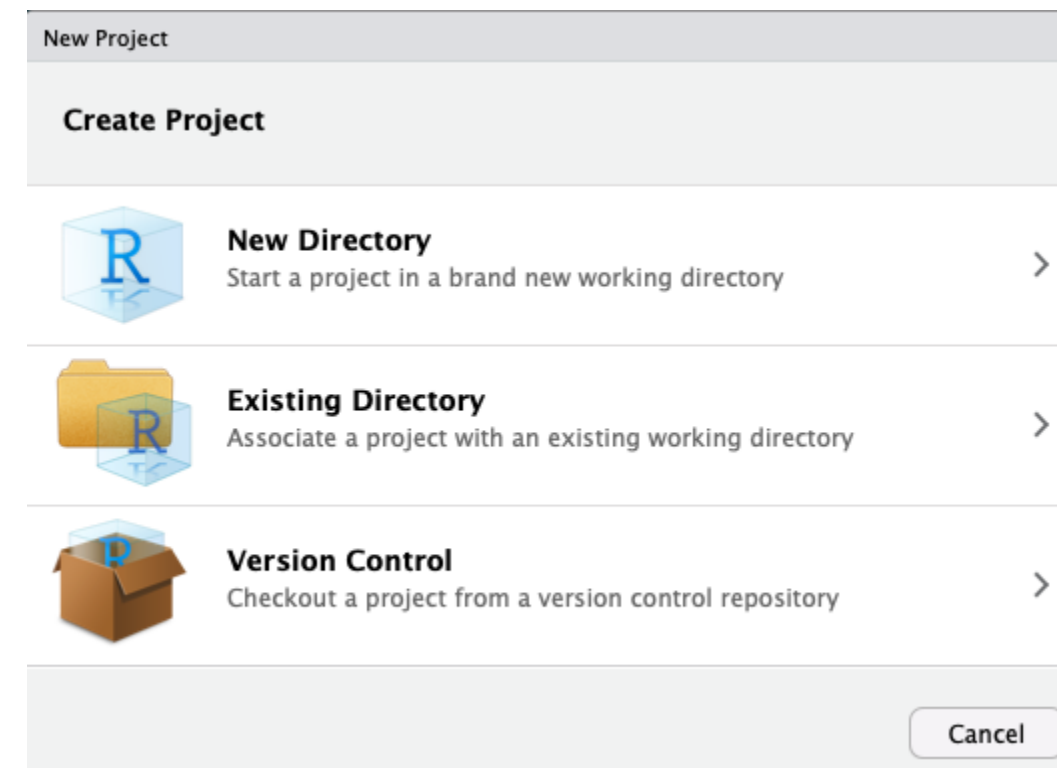
```
1 ---
2 format: pdf
3 ---
```



# Project-Oriented Workflow

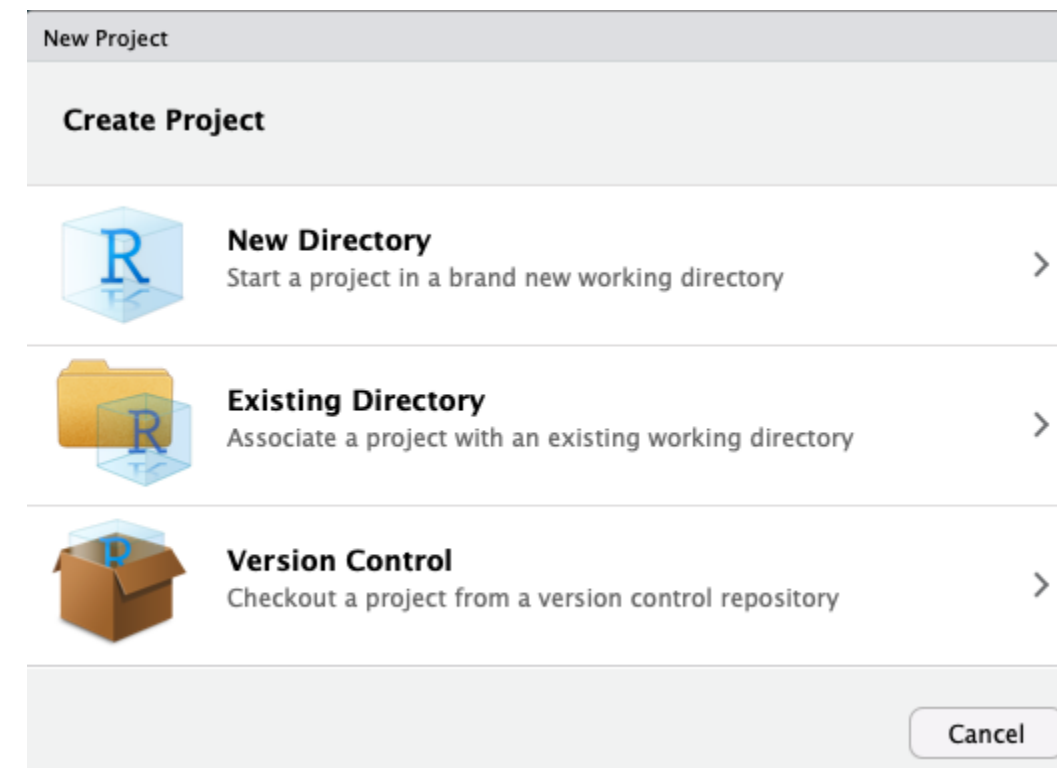
# R Projects I

- 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**



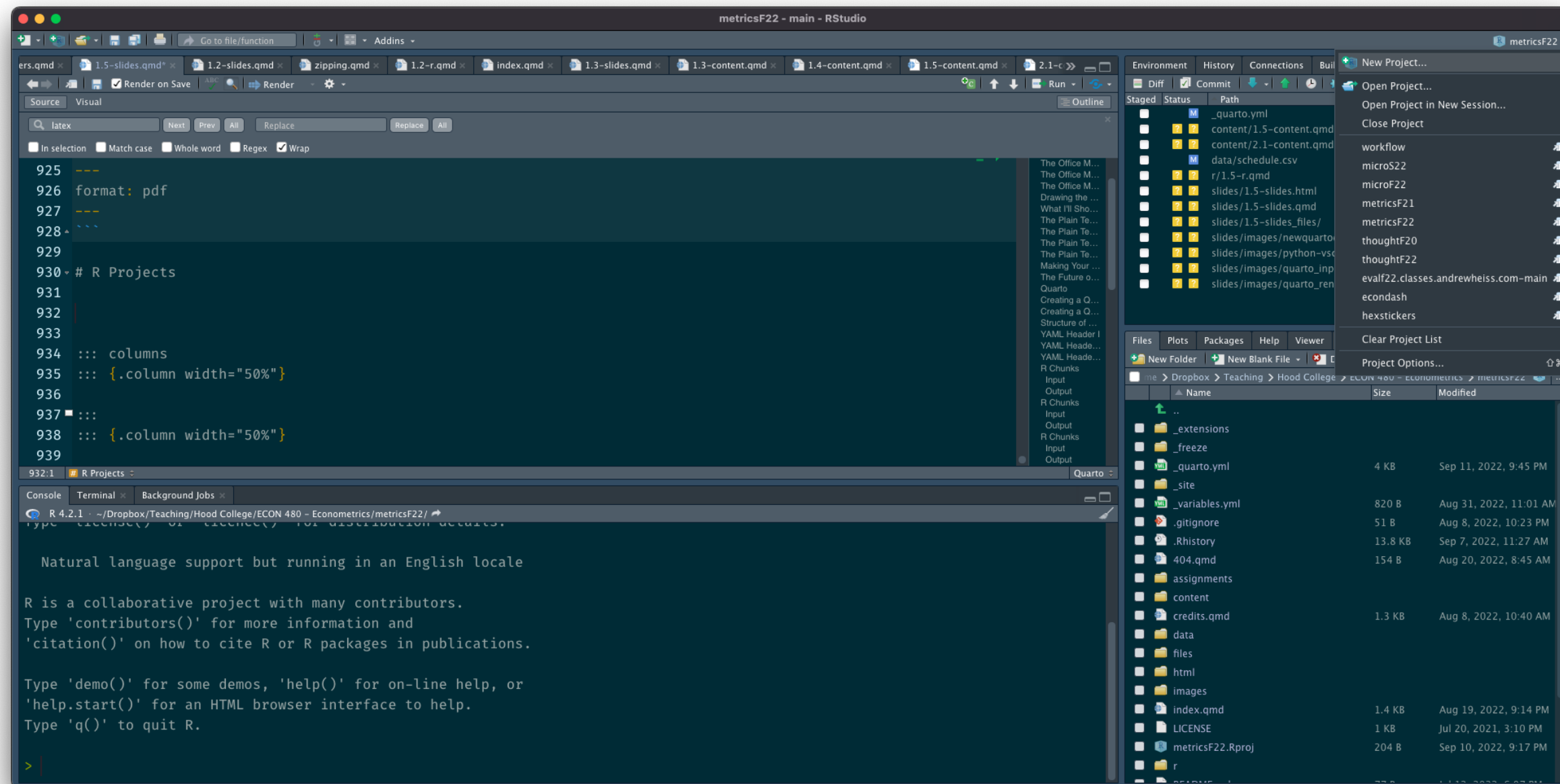
# Projects

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

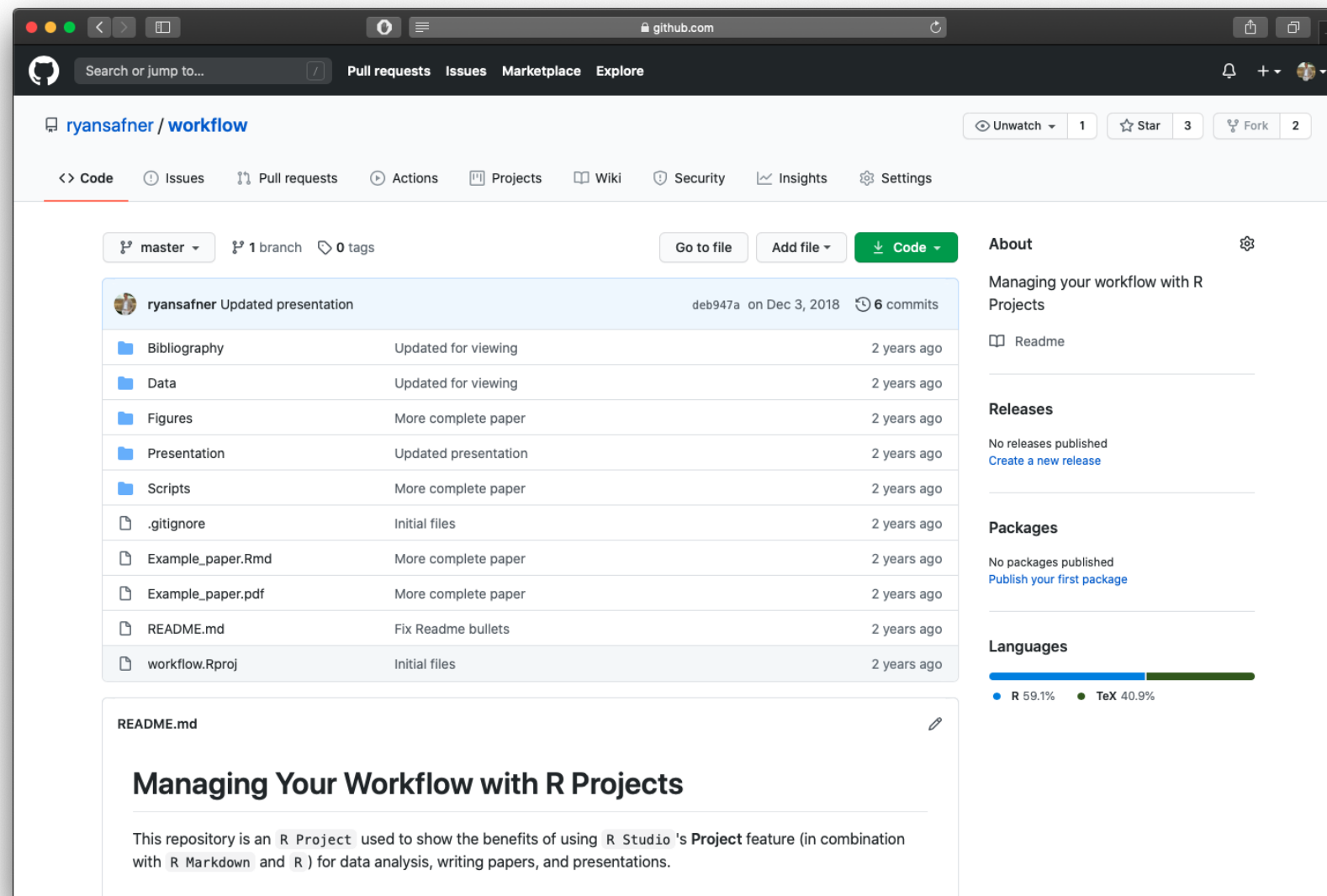


# Projects

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



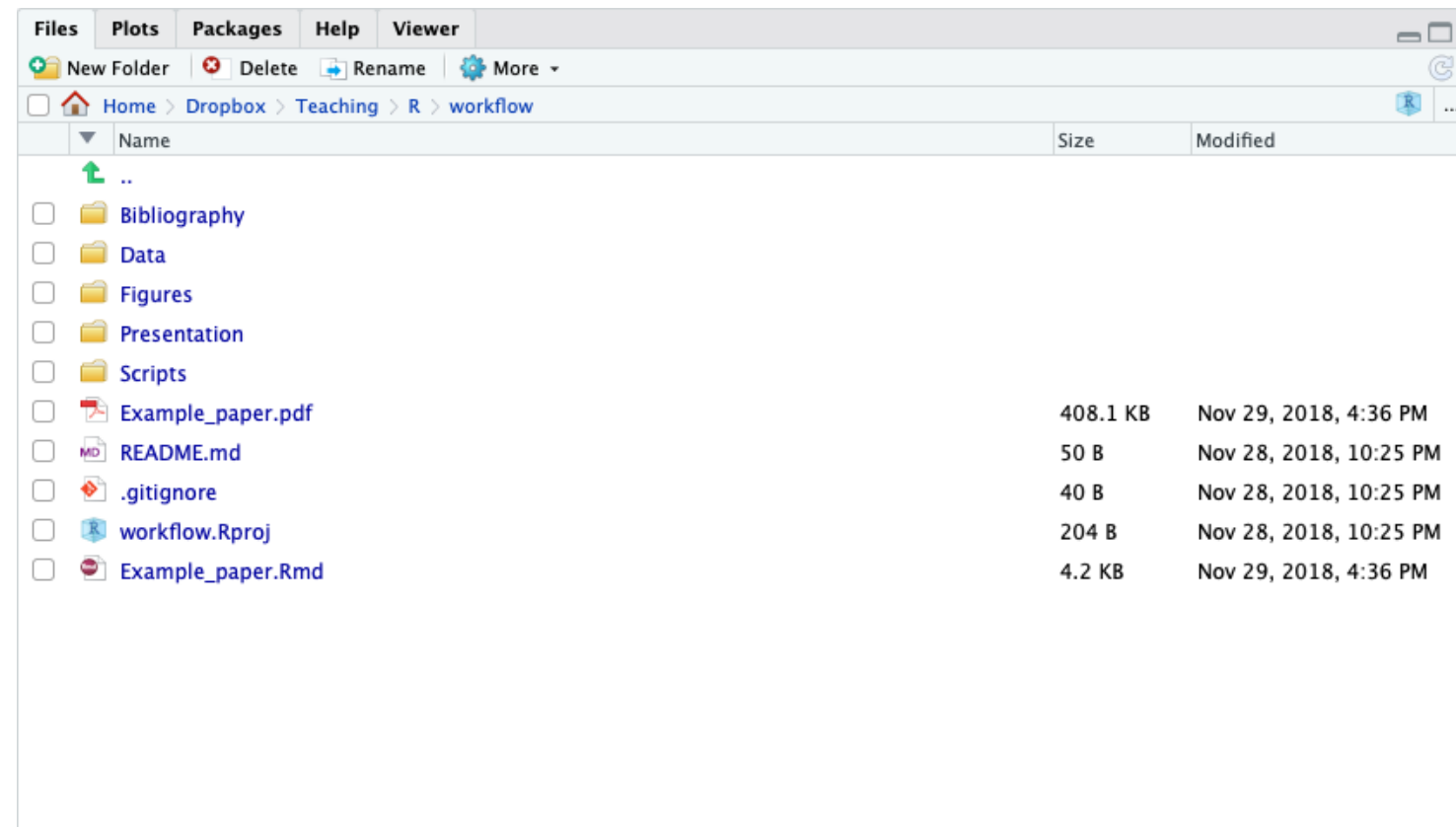
# Loading Others' Projects



- 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!

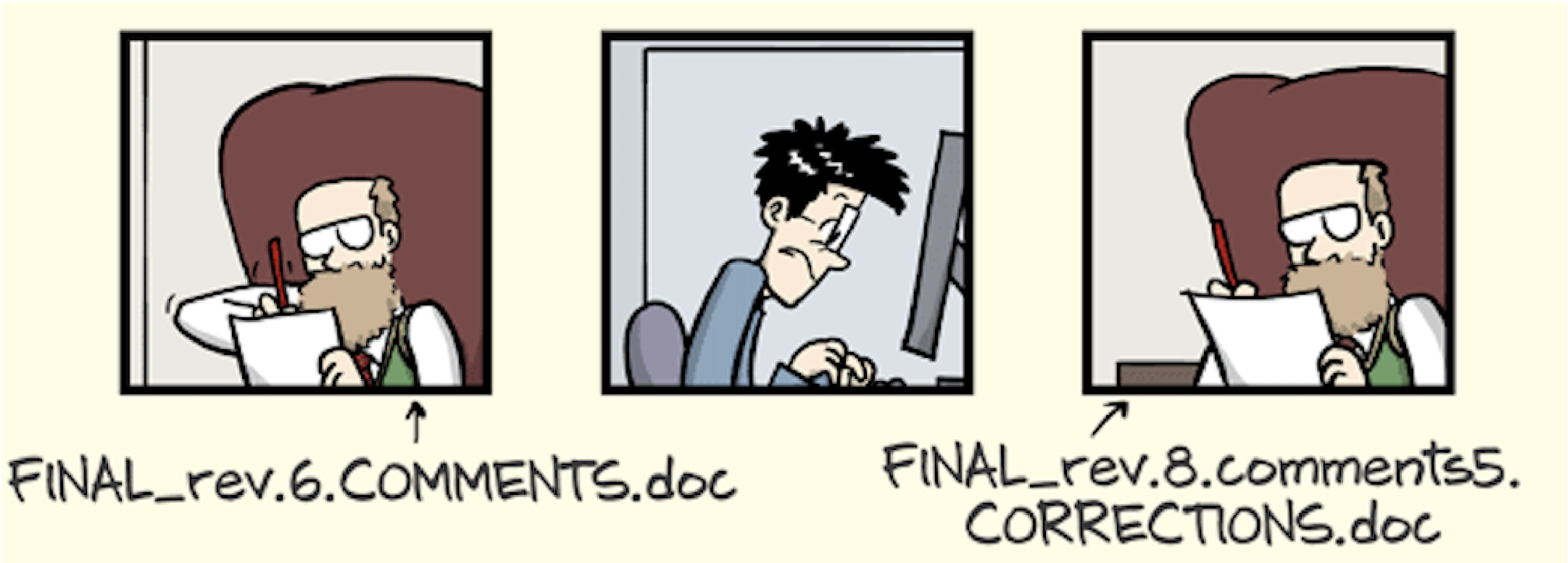


FINAL\_rev.2.doc

Source: [PhD Comics](#)



# Have You Done This?



Source: PhD Comics





# Have You Done This?



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

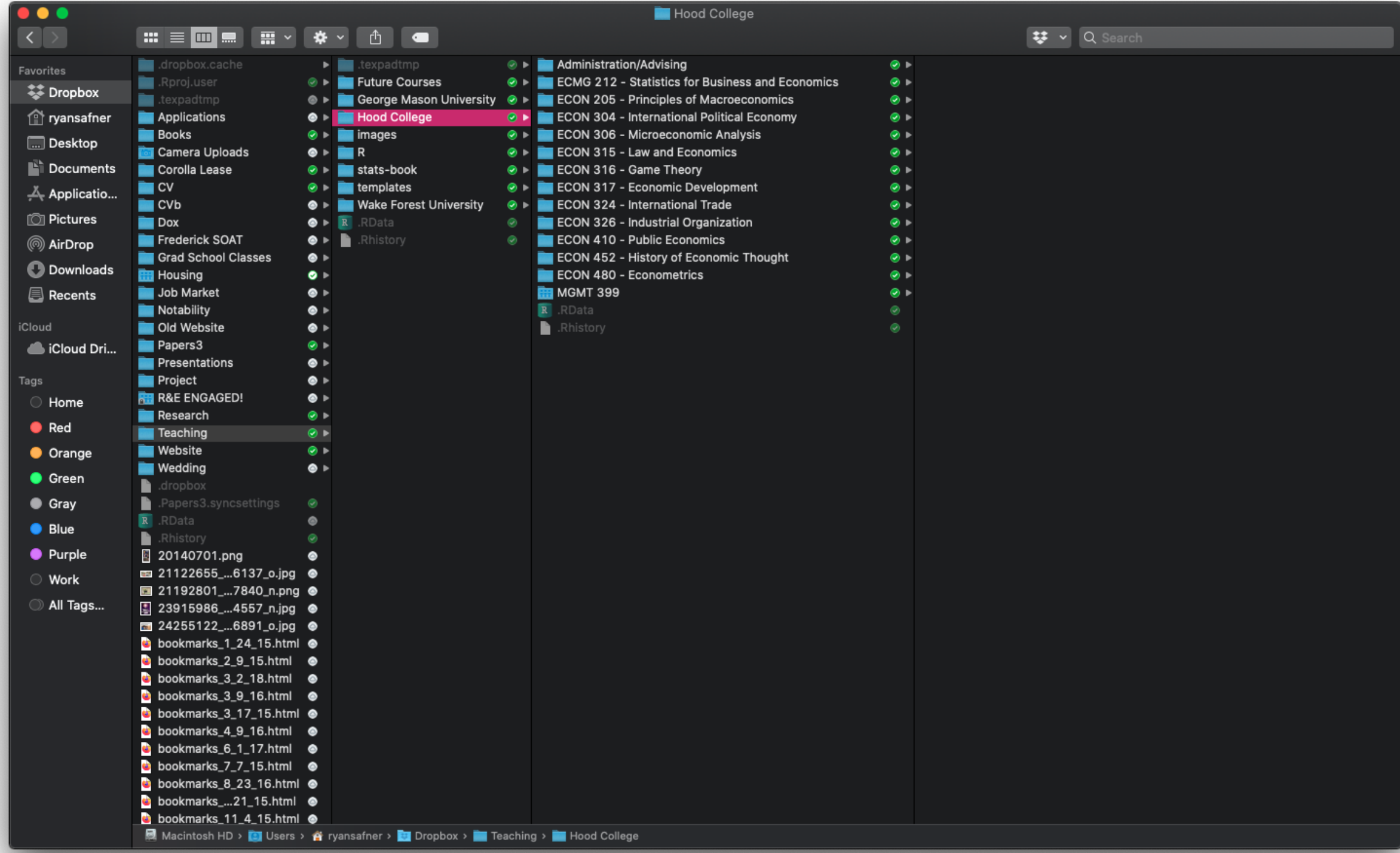


Dropbox.com

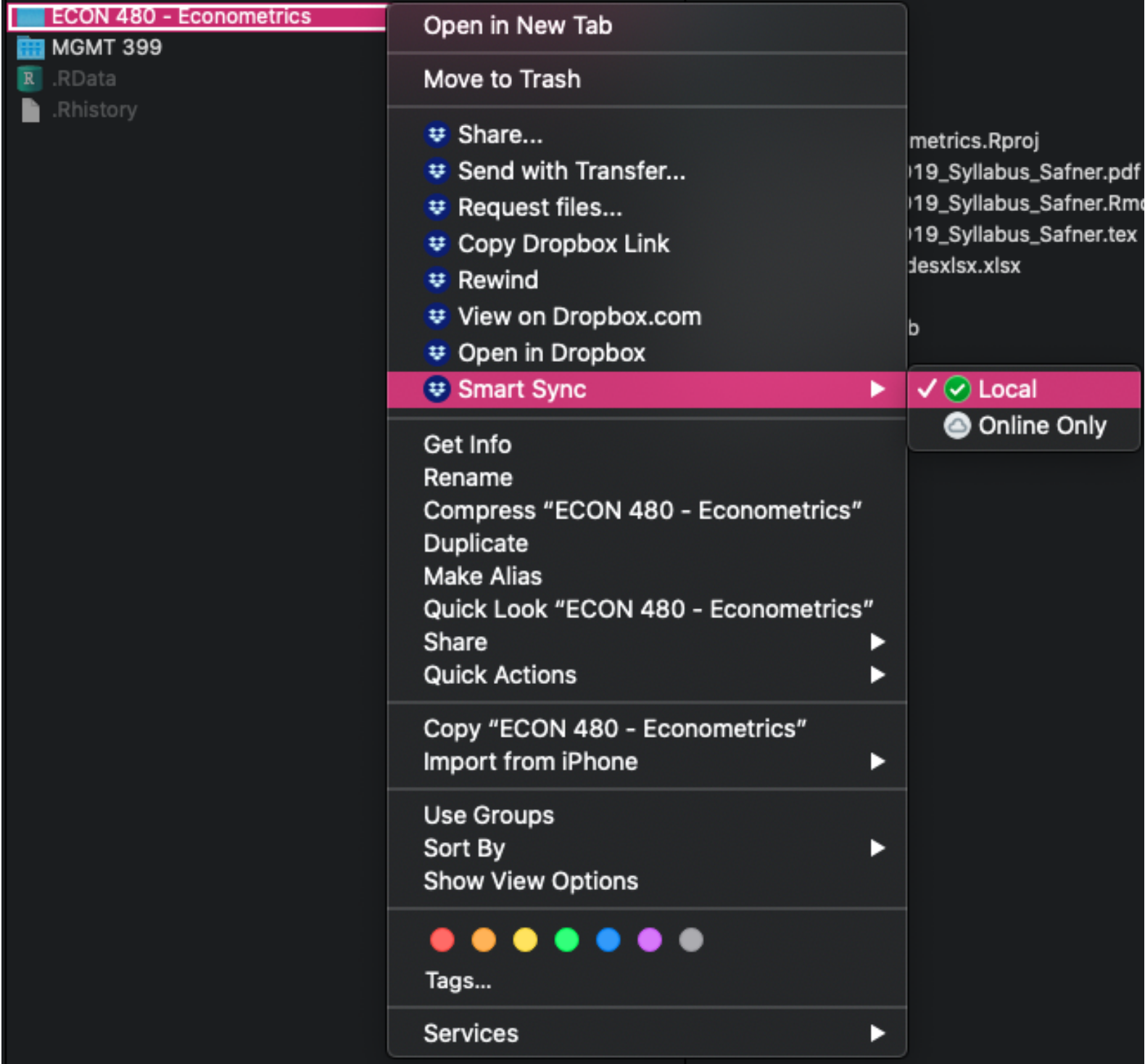
- Register an account for free
- Set up a location on your computer for the [Dropbox/](#) folder
- Anything you put in this folder will sync to the cloud
  - As soon as you change files, they *automatically* update and sync!
  - Can download any of these files from the *website* on any device
  - Set 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

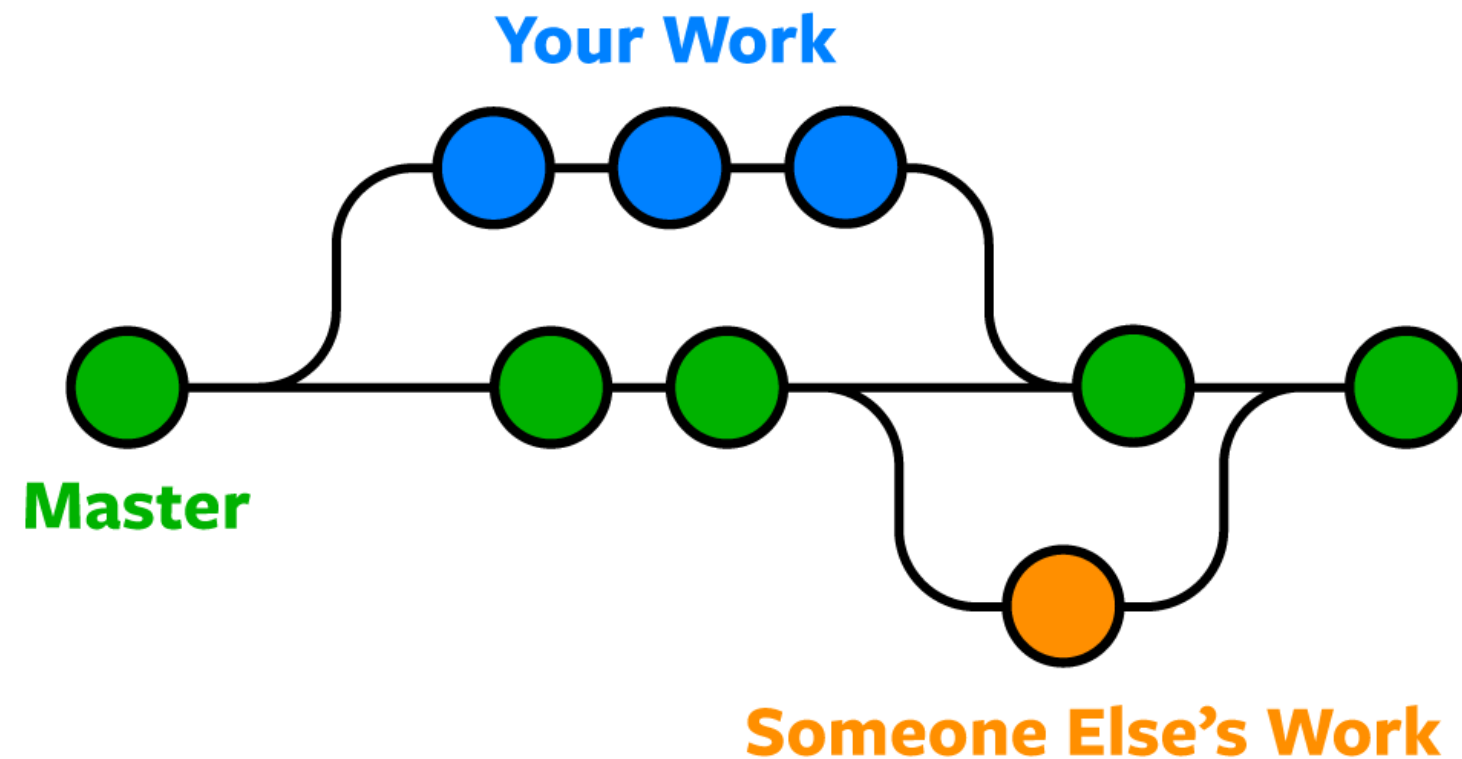


# The Expert Version





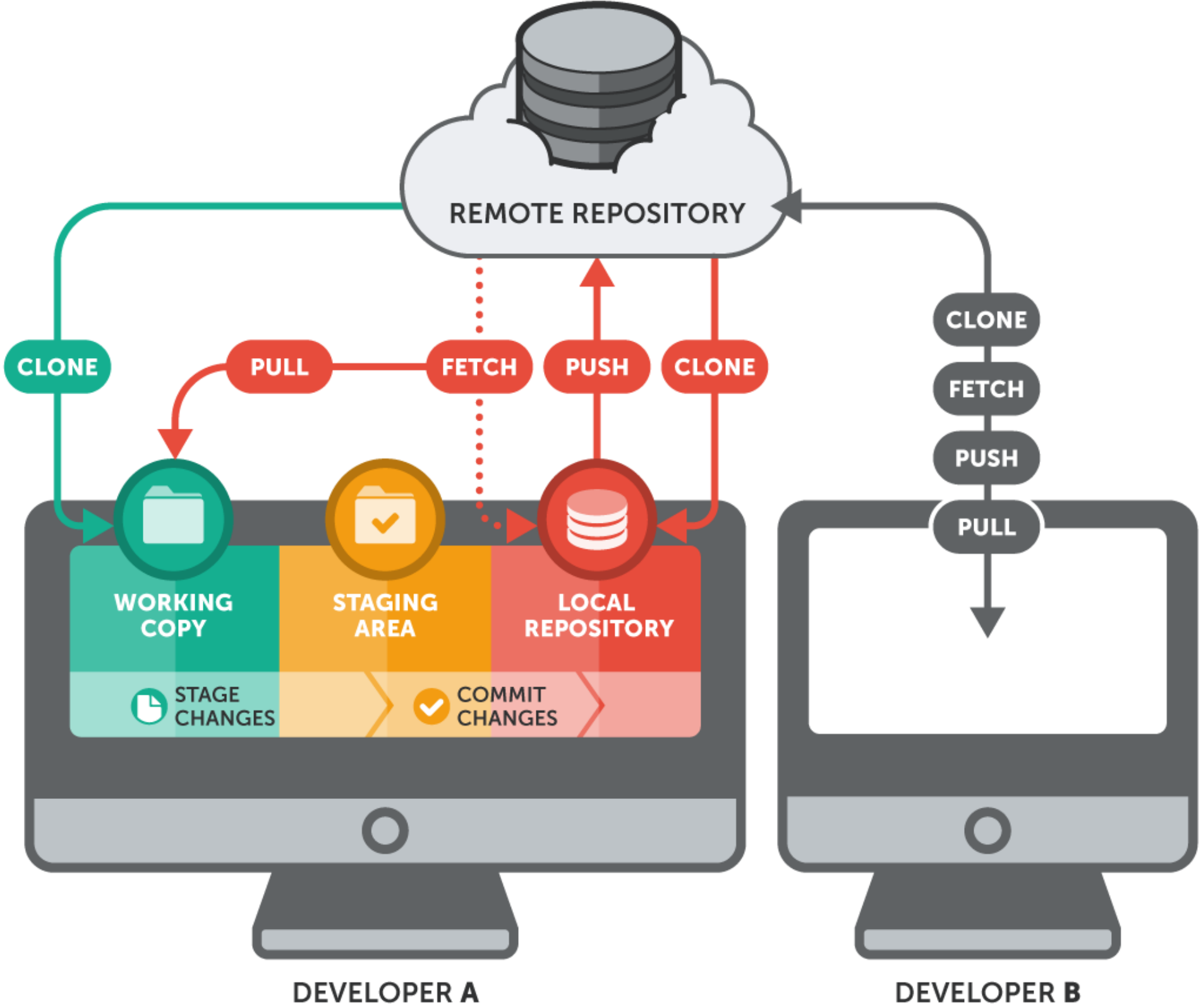
# The Expert Version



- 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

ryansafner / metricsF22 Public

Code Issues Pull requests Actions Projects Security Insights

main 1 branch 0 tags

ryansafner Upload 1.4 slide update 4 days ago 27 commits

File/Folder	Commit Message	Time Ago
_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

About: Course website for ECON 480 - Econometrics, Fall 2022 semester

Releases: No releases published

Packages: No packages published

Languages: HTML 47.2%, JavaScript 42.5%, CSS 10.1%, SCSS 0.2%, TeX 0.0%, Lua 0.0%

ryansafner / metricsF22 Public

Code Issues Pull requests Actions Projects Security Insights

main 1 branch 0 tags

ryansafner Upload 1.4 slide update 4 days ago 27 commits

File/Folder	Commit Message	Time Ago
_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

About: Course website for ECON 480 - Econometrics, Fall 2022 semester

Releases: No releases published

Packages: No packages published

Languages: HTML 47.2%, JavaScript 42.5%, CSS 10.1%, SCSS 0.2%, TeX 0.0%, Lua 0.0%

[github.com/ryansafner/metricsF22](https://github.com/ryansafner/metricsF22)



# Most Packages Start on Github

tidyverse / tidyverse

Watch 72 Star 751 Fork 156

Code Issues 5 Pull requests 1 Actions Security Insights

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

r data-science tidyverse

234 commits 2 branches 5 releases 24 contributors View license

Branch: master New pull request Create new file Upload files Find File Clone or download

Author	Commit Message	Time Ago
hadley	Update paper.md	6 days ago
	Add move bot config.	last year
	Only look at packages when considering conflicts	4 months ago
	Add repos option to tidyverse_{deps,update} (#82)	4 months ago
	Use retina logo	8 months ago
	Re-run revdep checks	4 months ago
	Add first test	last year
	Update paper.md	6 days ago
	Merge branch 'master' into joss-paper	3 months ago
	use_tidy_ci()	8 months ago
	Remove built website	8 months ago
	use_tidy_ci()	3 months ago
	Update for new R version	4 months ago
	Include GPL license text	3 years ago

[github.com/tidyverse/tidyverse](https://github.com/tidyverse/tidyverse)

jennybc / gapminder

Watch 14 Unstar 147 Fork 183

Code Issues 5 Pull requests 1 Projects 0 Wiki Security Insights

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

126 commits 1 branch 4 releases 4 contributors

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 ddf773 on Mar 6

File	Description	Time Ago
.aspell	Use .aspell	2 years ago
R	Document country_codes	2 years ago
data-raw	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](https://github.com/jennybc/gapminder)



# My Workflow (That I Suggest to You)

1. Create a new repository on Github.<sup>1</sup>
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

