

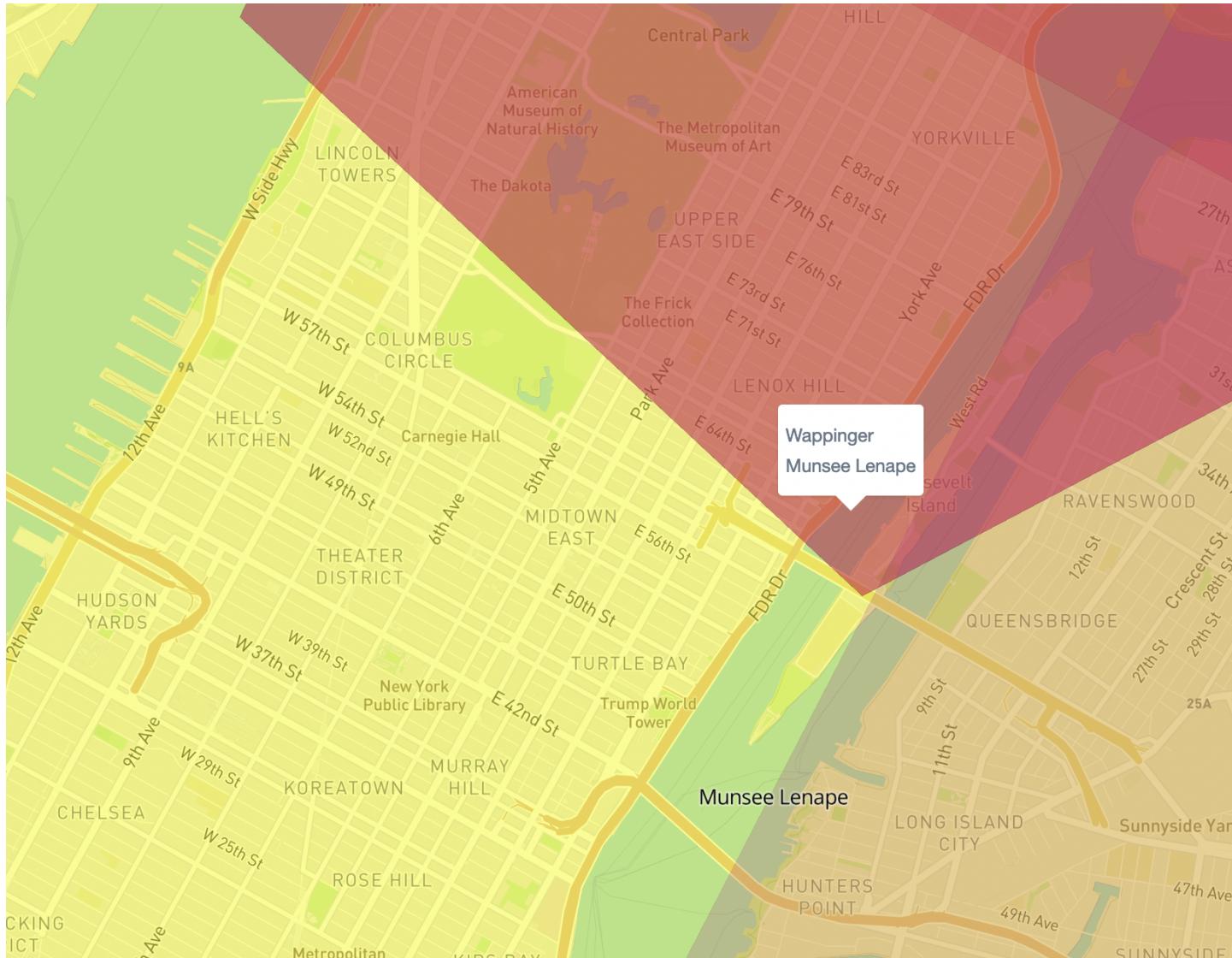
# Moving to Quarto from RMarkdown and Python Jupyter Notebooks

NYR Conference 2023

Daniel Chen

# Hello

# Munsee Lenape



# Daniel Chen



@chendaniely



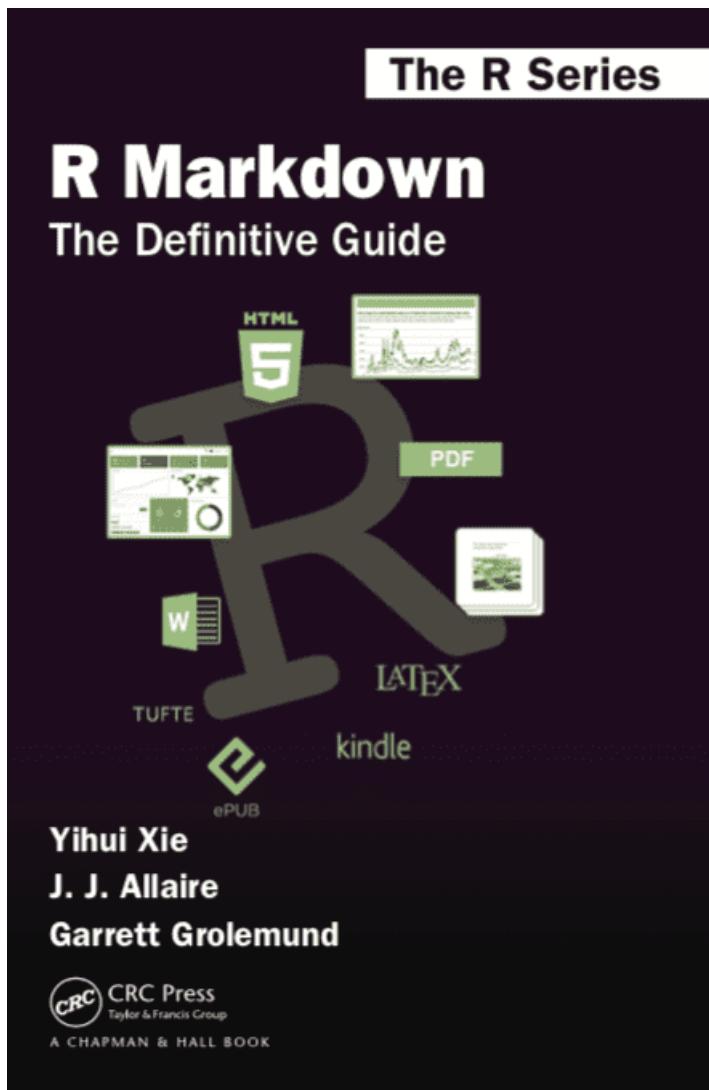
- Postdoctoral Research and Teaching Fellow  
UBC, MDS-Vancouver
- Data Science Educator,  
Posit, PBC
- The Carpentries
- Author, [Pandas for Everyone](#)

# Literate Programming

# Why Literate Programming?

- Data Scientist
  - RMarkdown + Jupyter Notebooks
    - Analysis
    - Reports + Documentation
- Technical Writer
  - Blog
  - Website
  - Presentation
  - Book
- Academic
  - Papers

# RMarkdown



# Code Chunks

```
1  ```{r}
2  cmv <- read_excel("data/cmv.xlsx")
3  head(cmv)
4  ```
```

# RMarkdown Document

```
1 ---  
2 title: "example-analysis"  
3 author: "Daniel Chen"  
4 output: html_document  
5 ---  
6 ```{r setup, include=FALSE}  
7 library(tidyverse)  
8 library(readxl)  
9 library(writexl)  
10 ```  
11 ## Load Data  
12 Retrospective Cohort Study of the Effects of  
13 Donor KIR genotype on the reactivation of cytomegalovirus (CMV)  
14 after myeloablative allogeneic hematopoietic stem cell transplant.  
15 ```{r}  
16 cmv <- read_excel("data/cmv.xlsx")  
17 head(cmv)  
18 ```
```

# Render .Rmd with {rmarkdown}

Demo file: `example-analysis.Rmd`

Render Command:

```
1 Rscript -e "rmarkdown::render('example-analysis.Rmd')"
```

Specify output file (and location):

```
1 Rscript -e "rmarkdown::render(  
2     input = 'example-analysis.Rmd',  
3     output_file = 'output/010-example-analysis-rmd.html')"
```

# Render .Rmd with quarto

Demo file: `example-analysis.Rmd`

Render Command:

```
1 quarto render example-analysis.Rmd
```

Specify output file:

```
1 # output folders only work with quarto projects
2 touch _quarto.yml
3
4 quarto render example-analysis.Rmd \
5   --toc \
6   --output output/020-example-analysis-rmd-qmd.html
```

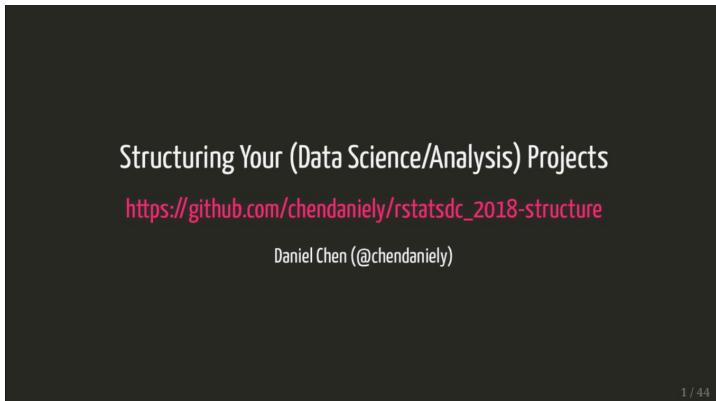
- **quarto is command line tool!**

# Caveat: Single Quarto Document

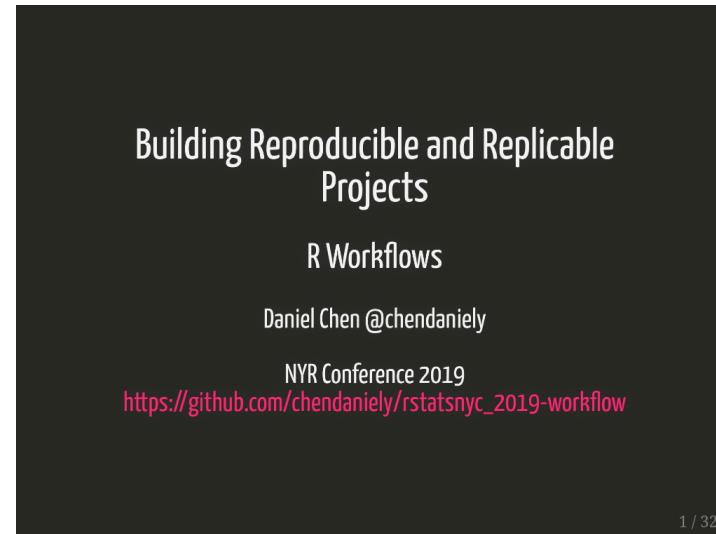
- Output directory Github Discussion
  - <https://github.com/quarto-dev/quarto-cli/discussions/2171>
- Pre and Post Render
  - <https://quarto.org/docs/projects/scripts.html#pre-and-post-render>

# Project templates

DCR 2018: Structuring Your (Data Science/Analysis) Projects

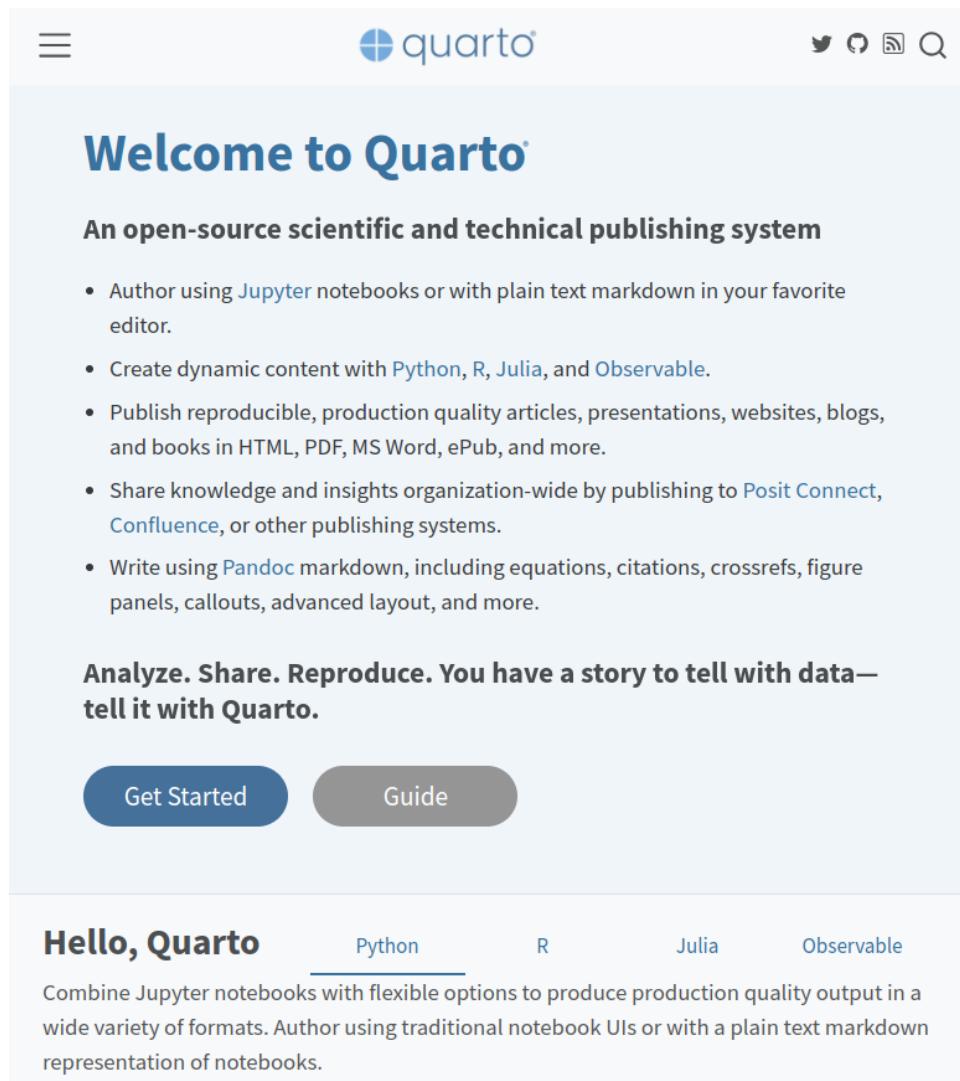


NYR 2019: Building Reproducible and Replicable Projects



Tiffany Timbers DSCI 310: Reproducible and trustworthy workflows for data science:  
<https://ubc-dsci.github.io/dsci-310-student/>

# Quarto

A screenshot of the Quarto website homepage. At the top, there is a navigation bar with a menu icon, the Quarto logo, and social media links for Twitter, GitHub, LinkedIn, and a search icon. The main heading is "Welcome to Quarto". Below it, a sub-headline reads "An open-source scientific and technical publishing system". A bulleted list details the features: using Jupyter notebooks or plain text markdown, creating dynamic content with Python, R, Julia, and Observable, publishing reproducible content in various formats, sharing knowledge across systems like Posit Connect and Confluence, and writing using Pandoc markdown with advanced features. A section titled "Analyze. Share. Reproduce. You have a story to tell with data—tell it with Quarto." includes "Get Started" and "Guide" buttons. At the bottom, a "Hello, Quarto" section encourages combining Jupyter notebooks with flexible output options, listing Python, R, Julia, and Observable as supported languages.

<https://quarto.org/>

- Plain text source document
- Literate programming
- Multiple language support
  - Even in the same document!
- Multiple output formats
  - Pandoc + Markdown
- Familiar
  - Quarto Gallery: <https://quarto.org/docs/gallery/>
  - Quarto Guide: <https://quarto.org/docs/guide/>
  - Quarto Reference: <https://quarto.org/docs/reference/>

# Quarto Documents

## RMarkdown YAML

```
1 ---  
2 title: "Example Analysis"  
3 subtitle: "RMarkdown"  
4 author: "Daniel Chen"  
5 output: html_document  
6 ---
```

## Quarto YAML

```
1 ---  
2 title: "Example Analysis"  
3 subtitle: "Quarto"  
4 author: "Daniel Chen"  
5 format: html  
6 ---
```

## RMarkdown and Quarto chunk options:

```
1 ```{r setup}  
2 #| include: false  
3 knitr::opts_chunk$set(echo = TRUE)  
4 library(tidyverse)  
5 library(readxl)  
6 library(writexl)  
7 ...
```

# Render a Quarto Document

Demo file: `example-analysis.qmd`

Render Command:

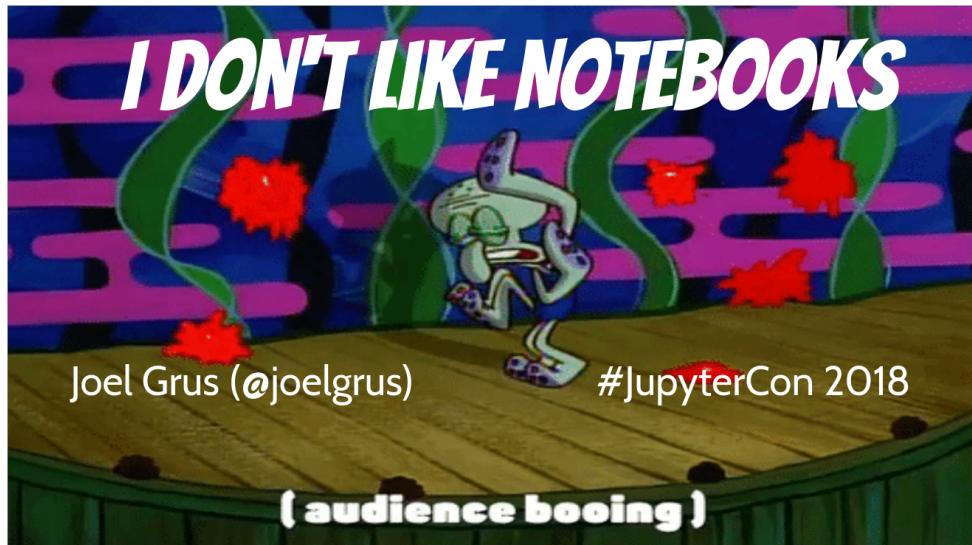
```
1 quarto render example-analysis.qmd
```

Specify output file:

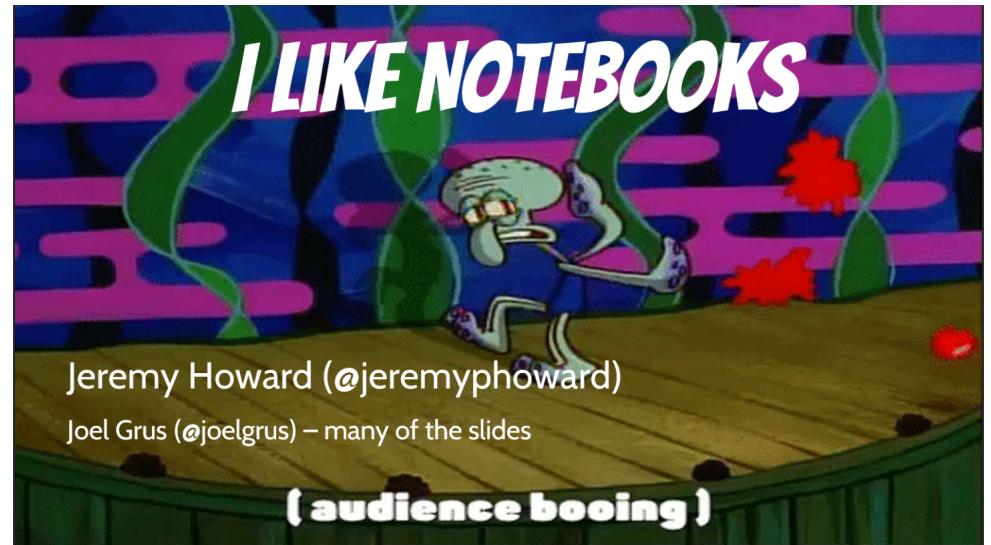
```
1 quarto render example-analysis.Rmd \
2   --toc \
3   --output-dir output \
4   --output 030-example-analysis-rmd-qmd.html
```

# Jupyter

# Notebooks



Youtube



Youtube

# Daniel's List

- Technical Writing
  -  Literate programming
  -  Editing JSON
- Data Science
  - More an output format than a source document
  -  Great for posting code+output (e.g. a workshop)
  -  Not great for source control collaborative document
- Teaching
  -  nbgrader for course assignment creation + grading
  -  Restart Kernel > Run All

# Jupyter Notebooks are JSON

```
1  {
2    "cells": [
3      {
4        "cell_type": "code",
5        "execution_count": 1,
6        "id": "4a9a7246-de20-4aac-945a-b8f0e7db0ac6",
7        "metadata": {},
8        "outputs": [],
9        "source": [
10          "import pandas as pd\n",
11          "import plotnine as p9\n",
12          "from plotnine import ggplot, aes, geom_histogram\n",
13          "import statsmodels.formula.api as smf"
14        ]
15      },
16      {
17        "cell_type": "markdown",
18        "id": "8f8205a7-a172-492a-bb22-e24bc1fc7ce2",
19        "metadata": {}
```

# Need Something to View + Render

# VSCode

The screenshot shows the VSCode interface with a Jupyter Notebook extension. It has a top bar with tabs for 'example-analysis.Rmd U', 'example-analysis.qmd U', 'example-analysis-python.ipynb U X', and 'example-analysis-python.ipynb'. The main area contains several code cells:

- [1]:

```
import pandas as pd
import plotnine as p9
from plotnine import ggplot, aes, geom_histogram
import statsmodels.formula.api as smf
```
- [2]:

Load Data

```
cmv = pd.read_excel("data/cmv.xlsx")
cmv.head()
```

ID	age	prior.radiation	aKIRs	cmv	donor_negative	donor_positive
0	1	61	0	1	1	recipient_positive
1	2	62	1	5	0	recipient_negative
2	3	63	0	3	0	NaN
3	4	33	1	2	0	recipient_positive
4	5	54	0	6	0	NaN
- [3]:

Filter Data

```
cmv_subset = cmv.loc[cmv["age"] > 65]
```
- [4]:

Save Data

```
cmv_subset.to_excel("data/cmv_subset-python-jupyter.xlsx")
```
- Tidy Data

# Jupyter Lab

The screenshot shows the Jupyter Lab interface with a Jupyter Notebook tab bar at the top. It has a sidebar on the left and several code cells:

- [1]:

```
import pandas as pd
import plotnine as p9
from plotnine import ggplot, aes, geom_histogram
import statsmodels.formula.api as smf
```
- [2]:

Load Data

```
cmv = pd.read_excel("data/cmv.xlsx")
cmv.head()
```

ID	age	prior.radiation	aKIRs	cmv	donor_negative	donor_positive
0	1	61	0	1	1	recipient_positive
1	2	62	1	5	0	recipient_negative
2	3	63	0	3	0	NaN
3	4	33	1	2	0	recipient_positive
4	5	54	0	6	0	NaN
- [3]:

Filter Data

```
cmv_subset = cmv.loc[cmv["age"] > 65]
```
- [4]:

Save Data

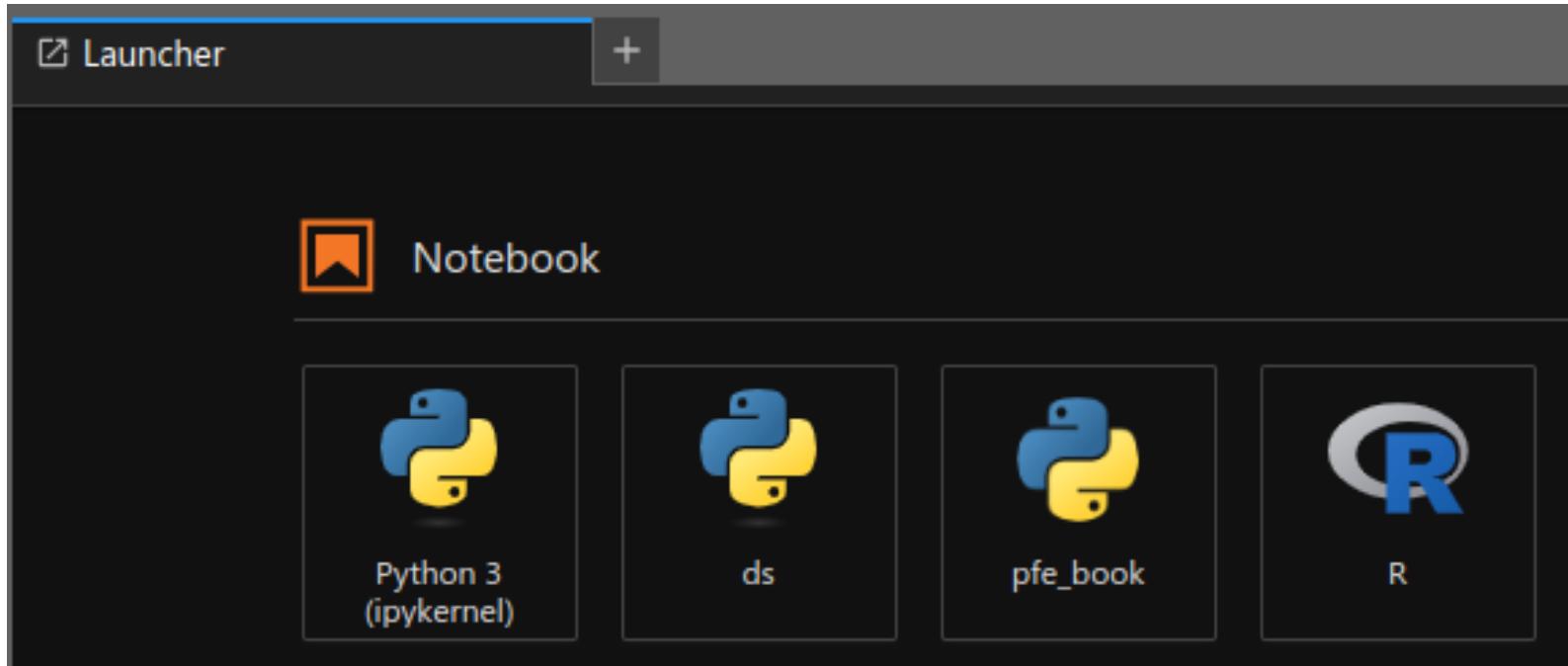
```
cmv_subset.to_excel("data/cmv_subset-python-jupyter.xlsx")
```
- Tidy Data

# Jupyter does R!

- You need the `IRKernel` installed:

`https://github.com/IRkernel/IRkernel`

```
1 install.packages('IRkernel')
2 IRkernel::installspec()
```



# Render .ipynb with nbconvert

- Demo files:
  - `example-analysis-python.ipynb`
  - `example-analysis-r.ipynb`

Python Kernel:

```
1 jupyter nbconvert \
2   --to html \
3   --output output/040-example-analysis-python-jupyter.html \
4   --execute example-analysis-python.ipynb
```

R Kernel:

```
1 jupyter nbconvert \
2   --to html \
3   --output output/050-example-analysis-r-jupyter.html \
4   --execute example-analysis-r.ipynb
```

(Hint: they're the same command)

# Jupyter Notebook as a Source Document

To make your version control diffing easier, you may want to clear the output from the notebook JSON file.

In `nbconvert 6.0+`, you can use `--clear-output --inplace`:

```
1 jupyter nbconvert --clear-output --inplace example-analysis-python.ipynb  
2 jupyter nbconvert --clear-output --inplace example-analysis-r.ipynb
```

Or use the `--to notebook` argument if you want to preserve a rendered notebook

# Render .ipynb with quarto

Takes whatever is in the notebook (no additional execution) and rendered (to html by default)

```
1 quarto render example-analysis-python.ipynb  
2 quarto render example-analysis-r.ipynb
```

## Use `--execute` to execute the cells and render

```
1 quarto render example-analysis-python.ipynb --execute  
2 quarto render example-analysis-r.ipynb --execute
```

# Render .ipynb with quarto

## Python Kernel:

```
1 quarto render example-analysis-python.ipynb \
2   --to html \
3   --execute \
4   --toc \
5   --output-dir output \
6   --output 060-example-analysis-python-ipynb.html
```

## R Kernel:

```
1 quarto render example-analysis-r.ipynb \
2   --to html \
3   --execute \
4   --toc \
5   --output-dir output \
6   --output 060-example-analysis-r-ipynb.html
```

# Embed Jupyter output in Quarto

From a Jupyter notebook with code output:

- Demo files:
  - `example-analysis-python-qmd_meta.ipynb`
  - `example-analysis-python-qmd_meta.qmd`

Using a notebook with existing output:

```
1 jupyter nbconvert \
2   --to notebook \
3   --execute \
4   --inplace \
5   example-analysis-python-qmd_meta.ipynb
```

You can add quarto #| metadata comments to a cell, and use jupyter output directly in a quarto document

# Embed Jupyter output in Quarto

```
1 #| label: fig-age_hist
2 #| fig-cap: >
3 #|     A histogram of the ages in our Cytomegalovirus dataset
4 ggplot(cmv_tidy, aes(x="age")) + geom_histogram()
```

Use a quarto shortcode:

```
1 {{< embed example-analysis-python-qmd_meta.ipynb#fig-age_hist >}}}
```

Render the example:

```
1 quarto render example-analysis-python-qmd_meta.qmd \
2   --to html \
3   --output-dir output \
4   --output 080-example-analysis-python-qmd_meta.html
```

<https://quarto.org/docs/authoring/notebook-embed.html>

# Converting

# jupytext

<https://jupytext.readthedocs.io/>

Rmd -> qmd

```
1 jupytext \
2     --to qmd \
3     --output output/090-convert-rmd_qmd.qmd \
4     example-analysis.Rmd
```

ipynb -> qmd

```
1 jupytext \
2     --to qmd \
3     --output output/100-convert-ipynb_qmd.qmd \
4     example-analysis-python.ipynb
```

# quarto convert

```
1 quarto convert example-analysis-python.ipynb \
2 --output output/120-convert-ipynb_qmd.qmd
```

# Publication

# Publish your files

```
1 quarto publish          # Publish Project (ask provider)
2 quarto publish talk.qmd # Publish document (ask provider)
3
4 quarto publish quarto-pub # Quarto.pub
5
6 quarto publish gh-pages # GitHub Pages
7 quarto publish netlify  # Netlify
8
9 quarto publish connect  # RStudio Connect
10 quarto publish confluence # Confluence
```

<https://quartopub.com/>

# Thanks!

# Thanks

@chendaniely

- [github.com/chendaniely/rstatsnyc-2023-quarto](https://github.com/chendaniely/rstatsnyc-2023-quarto)
- [chendaniely.quarto.pub/rstatsnyc-rmd-jupyter-quarto/](https://chendaniely.quarto.pub/rstatsnyc-rmd-jupyter-quarto/)

