

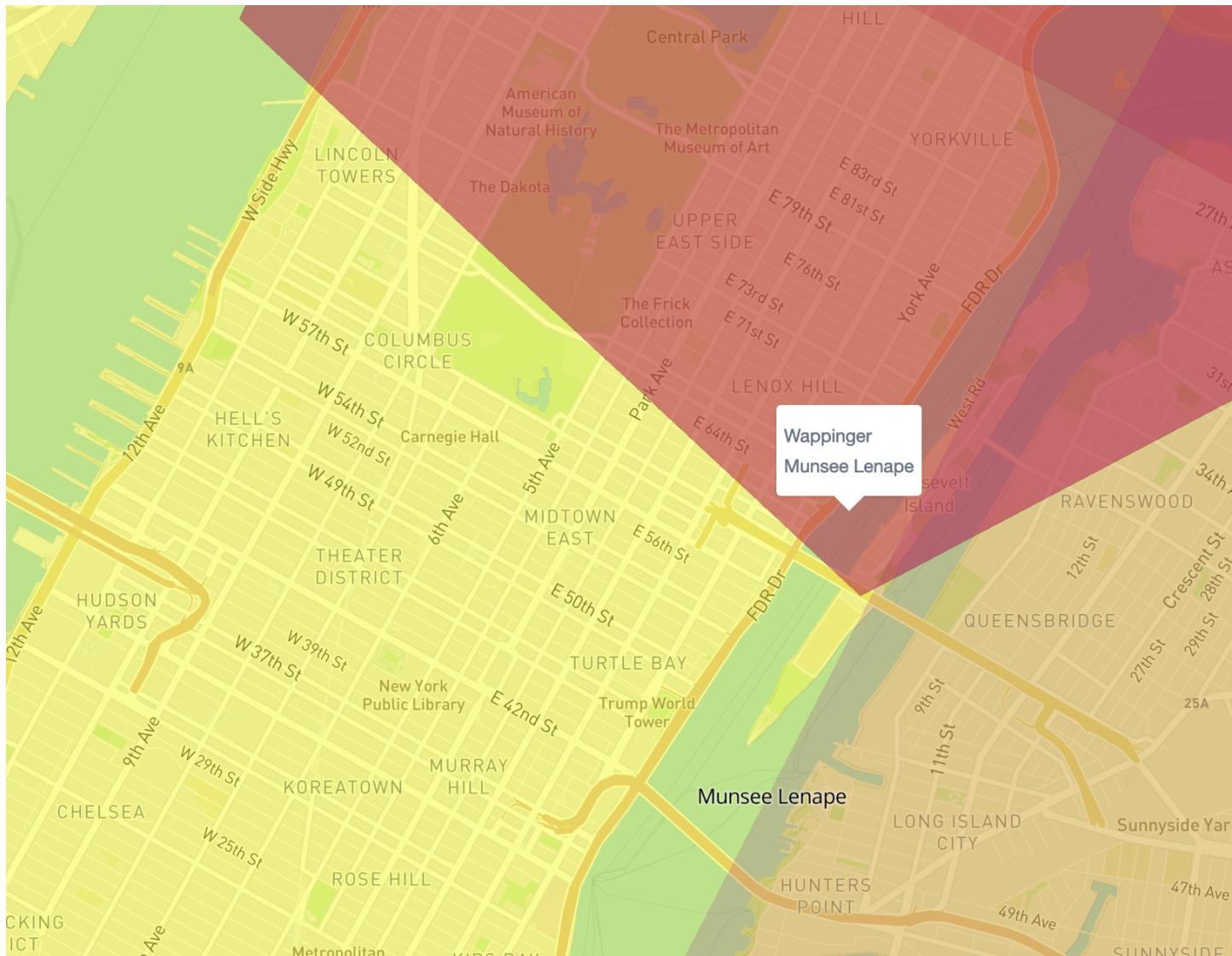
Install Python. Quarto Render All the Things

PyData NYC 2022

Daniel Chen

Hello 

Munsee Lenape



Daniel Chen

@chendaniely



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

Python + R



Quarto

What is Quarto?

The screenshot shows the Quarto website's main landing page. At the top left is the Quarto logo (a blue circle with a white cross). At the top right is a menu icon (three horizontal lines). The main title "Welcome to Quarto" is centered above a brief description: "Quarto® is an open-source scientific and technical publishing system built on Pandoc". Below this is a bulleted list of features:

- 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.

At the bottom of the main content area are two buttons: "Get Started" (blue) and "Guide" (grey).

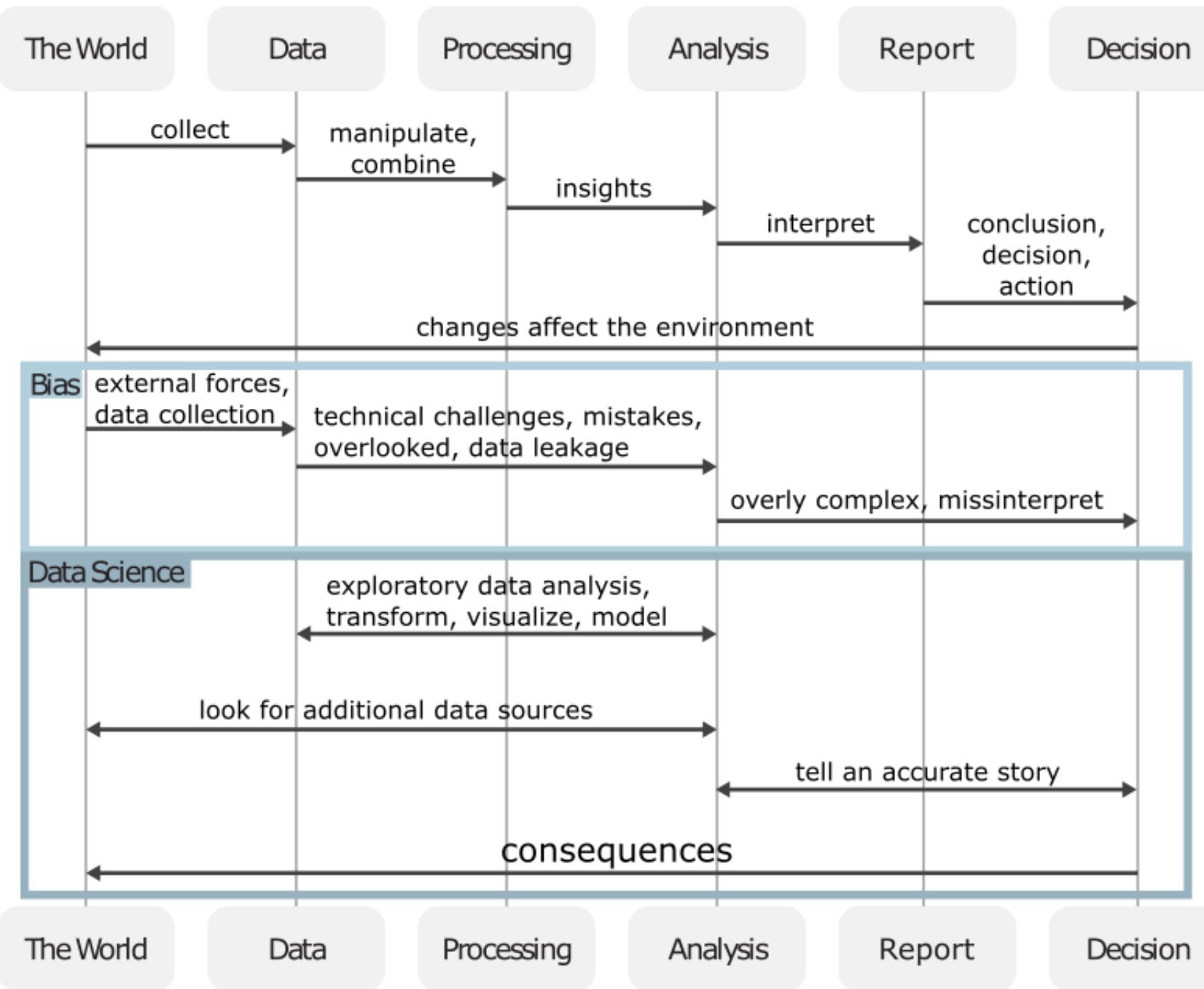
Below the main content, there is a navigation bar with tabs: "Hello, Quarto" (selected), "Python" (underlined), "R", "Julia", and "Observable". A descriptive paragraph follows:

Weave together narrative text and code to produce elegantly formatted output. Quarto documents are fully reproducible. Use markdown with code cells executed via Jupyter (shown below) or render existing Jupyter notebooks.

Who Would Use Quarto?

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

- Quarto Gallery: <https://quarto.org/docs/gallery/>
- Quarto Guide: <https://quarto.org/docs/guide/>
- Quarto Reference: <https://quarto.org/docs/reference/>





 Alison Presmanes Hill
@apreshill@fosstodon.org



Website apreshill.com/

I have more websites than blog posts. I love thinking about strategies for making software tools more usable, and data science more inclusive. Recovering academic. Former open-source product manager. PhD in psychology. Co- author of blogdown book and some R packages: palmerpenguins, bakeoff, blogdown, distill

Joined Nov 2022

370 Following 617 Followers

Doesn't Jupyter do that?

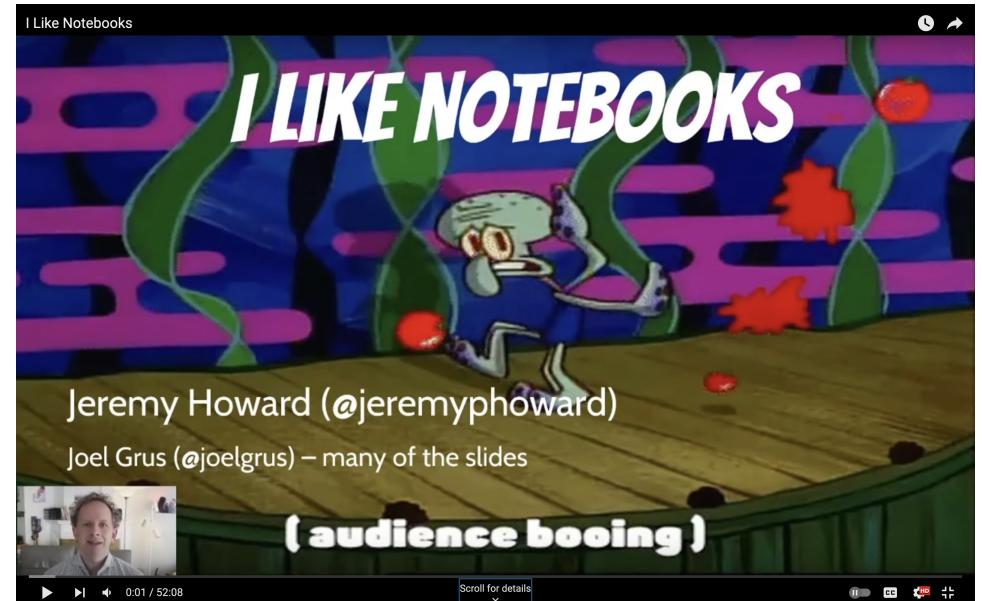
Julia + Python + R

Let's talk about Jupyter Notebooks...

Joel Grus JupyterCon 2018
“I don’t like notebooks”



Jeremy Howard “I like notebooks”



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

Quarto vs Jupyter

01-quarto.qmd — examples

01-quarto.qmd > Model

```

1 format: html
2
3
4
5 # Load
6
7 > Run Cell | Run Next Cell
8 ````{python}
9 from palmerpenguins import load_penguins
10 penguins = load_penguins()
11
12
13 # EDA
14
15 > Run Cell | Run Next Cell | Run Above
16 ````{python}
17 import pandas as pd
18
19 penguins.head()
20 penguins["sex_01"] = penguins.sex.replace({"male": 1, "female":0})
21 penguins.describe()
22
23
24 # Plot
25
26 > Run Cell | Run Next Cell | Run Above
27 ````{python}
28 from plotnine import ggplot, aes, geom_boxplot
29
30 g = (
31     ggplot(data=penguins, mapping=aes(x="sex", y="body_mass_g", color="species"))
32     + geom_boxplot()
33 )
34 g
35
36
37 # Model
38
39
40 > Run Cell | Run Above
41 ````{python}
42 import statsmodels.formula.api as smf
43
44 pen_no_na = penguins.dropna()
45 log_reg = smf.logit("sex_01 ~ body_mass_g", data=pen_no_na).fit()
46 log_reg.summary()
47

```

4 main* 0 0 0 0 9 Live Share Git Graph Quarto: 1.1.251 Spaces: 4 UTF-8 LF Quarto ▲ 9 Spell 🔍

01-jupyter.ipynb — JupyterLab

File Edit View Run Kernel Tabs Settings Help

Launcher 01-jupyter.ipynb + Python 3 (ipykernel)

Load

```
[4]: from palmerpenguins import load_penguins
penguins = load_penguins()
```

EDA

```
[5]: import pandas as pd
penguins.head()
penguins["sex_01"] = penguins.sex.replace({"male": 1, "female":0})
penguins.describe()
```

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	year	sex_01
count	342.000000	342.000000	342.000000	342.000000	344.000000	333.000000
mean	43.921930	17.151170	200.915205	4201.754386	2008.029070	0.504505
std	5.459584	1.974793	14.061714	801.954536	0.818356	0.500732
min	32.100000	13.100000	172.000000	2700.000000	2007.000000	0.000000
25%	39.225000	15.600000	190.000000	3550.000000	2007.000000	0.000000
50%	44.450000	17.300000	197.000000	4050.000000	2008.000000	1.000000
75%	48.500000	18.700000	213.000000	4750.000000	2009.000000	1.000000
max	59.600000	21.500000	231.000000	6300.000000	2009.000000	1.000000

Plot

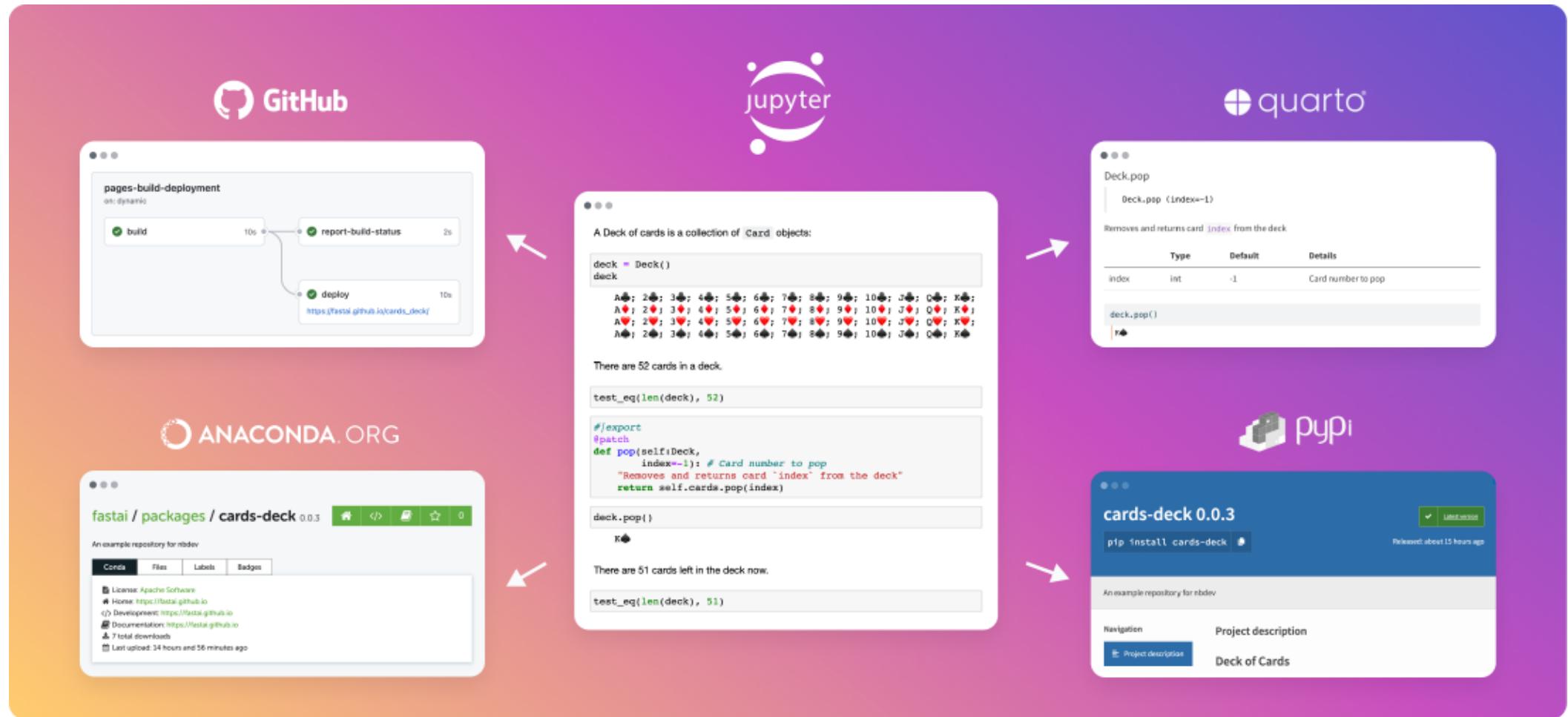
```
[6]: from plotnine import ggplot, aes, geom_boxplot
g = (
    ggplot(data=penguins, mapping=aes(x="sex", y="body_mass_g", color="species"))
    + geom_boxplot()
)
g
/Users/danielchen/.pyenv/versions/3.9.11/envs/ds/lib/python3.9/site-packages/plotnine/layer.py:334: PlotnineWarning: stat_boxplot : Removed 2 rows containing non-finite values.
```

Simple 0 Saving completed Mode: Command Ln 1, Col 1 01-jupyter.ipynb

Quarto ➔ Jupyter



fast.ai



- <https://www.fast.ai/posts/2022-07-28-nbdev2.html>

Let's do an analysis

Load - EDA - Plot - Model

```
1 from palmerpenguins import load_penguins  
2  
3 penguins = load_penguins()  
4 penguins.head()
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	bo
0	Adelie	Torgersen	39.1	18.7	181.0	37
1	Adelie	Torgersen	39.5	17.4	186.0	38
2	Adelie	Torgersen	40.3	18.0	195.0	32
3	Adelie	Torgersen	NaN	NaN	NaN	Na
4	Adelie	Torgersen	36.7	19.3	193.0	34

Load - EDA - Plot - Model

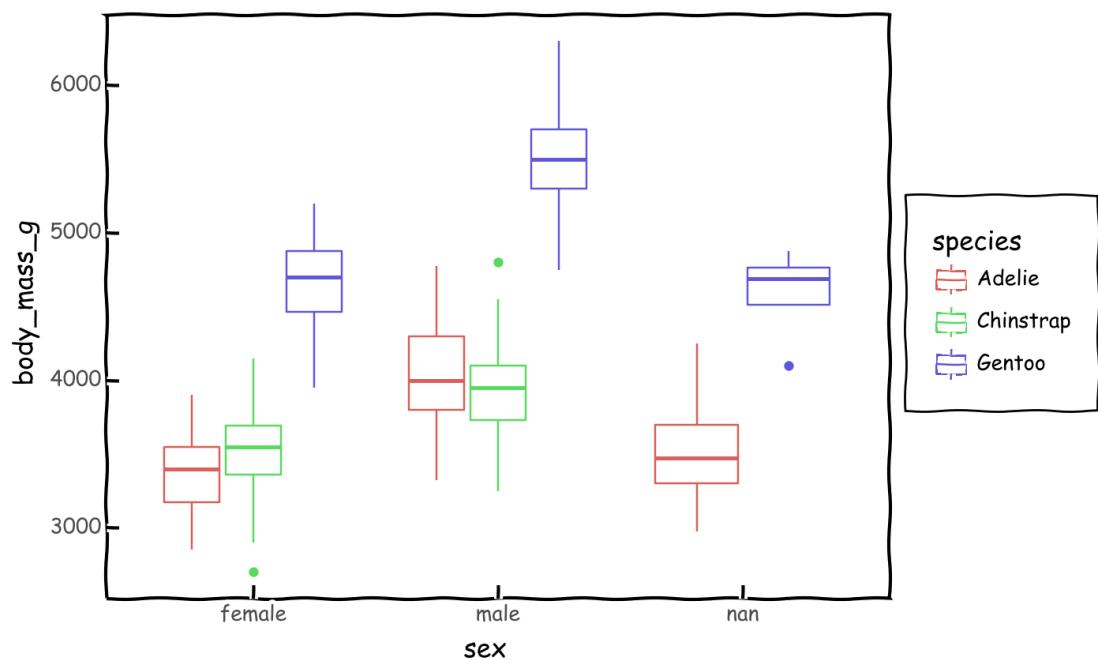
```
1 import pandas as pd  
2  
3 penguins.describe()
```

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	
count	342.000000	342.000000	342.000000	342.000000	344.
mean	43.921930	17.151170	200.915205	4201.754386	2008
std	5.459584	1.974793	14.061714	801.954536	0.81
min	32.100000	13.100000	172.000000	2700.000000	200
25%	39.225000	15.600000	190.000000	3550.000000	200
50%	44.450000	17.300000	197.000000	4050.000000	2008
75%	48.500000	18.700000	213.000000	4750.000000	2009
max	59.600000	21.500000	231.000000	6300.000000	2009

Load - EDA - *Plot*- Model

```
1 from plotnine import ggplot, aes, geom_boxplot, theme_xkcd
2
3 (
4     ggplot(
5         data=penguins,
6         mapping=aes(x="sex", y="body_mass_g", color="species")
7     )
8     + geom_boxplot()
9     + theme_xkcd()
10 )
```

<ggplot: (327126866)>



Load - EDA - Plot - Model

```
1 import statsmodels.formula.api as smf
2
3 penguins["sex_01"] = penguins.sex.replace({"male": 1, "female":0})
4 pen_no_na = penguins.dropna()
5
6 log_reg = smf.logit("sex_01 ~ body_mass_g", data=pen_no_na).fit()
7 log_reg.params
```

Optimization terminated successfully.

Current function value: 0.595563
Iterations 5

Intercept -5.162542
body_mass_g 0.001240
dtype: float64

Model Ops



Understand
and clean data



tidyverse



siuba



data.t

Isabel Zimmerman
Holistic MLOps for
Better Science



Train and
evaluate model

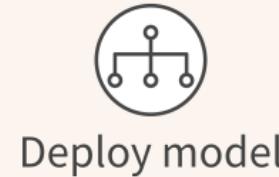


caret

K Keras PyT



Version model



Let's go make things

Create .qmd or .ipynb

01-quarto.qmd — examples

01-quarto.qmd > Model

```

1 format: html
2
3
4
5 # Load
6
7 > Run Cell | Run Next Cell
8 ````{python}
9 from palmerpenguins import load_penguins
10 penguins = load_penguins()
11
12
13 # EDA
14
15 > Run Cell | Run Next Cell | Run Above
16 ````{python}
17 import pandas as pd
18
19 penguins.head()
20 penguins["sex_01"] = penguins.sex.replace({"male": 1, "female":0})
21 penguins.describe()
22
23
24 # Plot
25
26 > Run Cell | Run Next Cell | Run Above
27 ````{python}
28 from plotnine import ggplot, aes, geom_boxplot
29
30 g = (
31     ggplot(data=penguins, mapping=aes(x="sex", y="body_mass_g", color="species"))
32     + geom_boxplot()
33 )
34 g
35
36
37 # Model
38
39
40 > Run Cell | Run Above
41 ````{python}
42 import statsmodels.formula.api as smf
43
44 pen_no_na = penguins.dropna()
45 log_reg = smf.logit("sex_01 ~ body_mass_g", data=pen_no_na).fit()
46 log_reg.summary()
47

```

main* ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ Live Share Git Graph Quarto: 1.1.251 Spaces: 4 UTF-8 LF Quarto ▲ ⑩ Spell 🔍

01-jupyter.ipynb — JupyterLab

File Edit View Run Kernel Tabs Settings Help

Launcher 01-jupyter.ipynb + Python 3 (ipykernel)

Load

```
[4]: from palmerpenguins import load_penguins
penguins = load_penguins()
```

EDA

```
[5]: import pandas as pd
penguins.head()
penguins["sex_01"] = penguins.sex.replace({"male": 1, "female":0})
penguins.describe()
```

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	year	sex_01
count	342.000000	342.000000	342.000000	342.000000	344.000000	333.000000
mean	43.921930	17.151170	200.915205	4201.754386	2008.029070	0.504505
std	5.459584	1.974793	14.061714	801.954536	0.818356	0.500732
min	32.100000	13.100000	172.000000	2700.000000	2007.000000	0.000000
25%	39.225000	15.600000	190.000000	3550.000000	2007.000000	0.000000
50%	44.450000	17.300000	197.000000	4050.000000	2008.000000	1.000000
75%	48.500000	18.700000	213.000000	4750.000000	2009.000000	1.000000
max	59.600000	21.500000	231.000000	6300.000000	2009.000000	1.000000

Plot

```
[6]: from plotnine import ggplot, aes, geom_boxplot
g = (
    ggplot(data=penguins, mapping=aes(x="sex", y="body_mass_g", color="species"))
    + geom_boxplot()
)
g
/Users/danielchen/.pyenv/versions/3.9.11/envs/ds/lib/python3.9/site-packages/plotnine/layer.py:334: PlotnineWarning: stat_boxplot : Removed 2 rows containing non-finite values.
```

Simple ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ Python 3 (ipykernel) | Idle Saving completed Mode: Command Ln 1, Col 1 01-jupyter.ipynb

The YAML header

```
1 ---  
2 format: html  
3 ---
```

```
1 ---  
2 format: html  
3 title: Quarto Document  
4 subtitle: Data Science!  
5 author: Daniel Chen  
6 toc: true  
7 toc-deph: 3  
8 code-overflow: scroll  
9 code-line-numbers: true  
10 execute:  
11   echo: true  
12   keep-md: true  
13   keep-ipynb: true  
14   jupyter: python3  
15 ---
```

- HTML Reference: <https://quarto.org/docs/reference/formats/html.html>

The `.qmd` code chunk

```
```{python}
#| echo: true
#| eval: true

from palmerpenguins import load_penguins

penguins = load_penguins()
penguins.head()
```
```

- Markdown
- Engine: `{python}`, `{r}`, etc
 - Knitr for `{r}`
 - Jupyter for any other block `{python}`, `{julia}`, etc
- Chunk options: `#|`
- Code

Make the document!

```
1 % quarto render examples/01-quarto.qmd
2
3 Executing '01-quarto.ipynb'
4   Cell 1/4...Done
5   Cell 2/4...Done
6   Cell 3/4...Done
7   Cell 4/4...Done
8
9 pandoc
10  to: html
11  output-file: 01-quarto.html
12  standalone: true
13  section-divs: true
14  html-math-method: mathjax
15  wrap: none
16  default-image-extension: png
17  toc: true
18
19
```

Works on your existing Jupyter notebook

```
1 % quarto render examples/02-jupyter.ipynb --execute
2
3 Starting python3 kernel...Done
4
5 Executing '02-jupyter.ipynb'
6   Cell 1/5...Done
7   Cell 2/5...Done
8   Cell 3/5...Done
9   Cell 4/5...Done
10  Cell 5/5...Done
11
12 pandoc
13   to: html
14   output-file: 02-jupyter.html
15   standalone: true
16   section-divs: true
17   html-math-method: mathjax
18   wrap: none
19   
```

Profit!

The screenshot shows a Quarto Document window titled "Quarto Document". The interface includes a top navigation bar with file, search, and zoom controls, and a sidebar with a "Table of contents" section containing "Load", "EDA", "Plot", and "Model" items. The main content area is divided into sections: "Data Science!", "AUTHOR Daniel Chen", and "Load".

Load

```
1 from palmerpenguins import load_penguins
2
3 penguins = load_penguins()
4 penguins.head()
```

| | species | island | bill_length_mm | bill_depth_mm | flipper_length_mm | body |
|---|---------|-----------|----------------|---------------|-------------------|------|
| 0 | Adelie | Torgersen | 39.1 | 18.7 | 181.0 | 3750 |
| 1 | Adelie | Torgersen | 39.5 | 17.4 | 186.0 | 3800 |
| 2 | Adelie | Torgersen | 40.3 | 18.0 | 195.0 | 3250 |
| 3 | Adelie | Torgersen | NaN | NaN | NaN | NaN |
| 4 | Adelie | Torgersen | 36.7 | 19.3 | 193.0 | 3450 |

EDA

```
1 import pandas as pd
2
3 penguins.describe()
```

| | bill_length_mm | bill_depth_mm | flipper_length_mm | body_mass_g | year |
|-------|----------------|---------------|-------------------|-------------|------------|
| count | 342.000000 | 342.000000 | 342.000000 | 342.000000 | 344.000000 |
| mean | 39.21930 | 17.151170 | 200.915205 | 4201.754386 | 2008 |
| std | 5.459584 | 1.974793 | 14.061714 | 801.954536 | 0.818 |
| min | 32.100000 | 13.100000 | 172.000000 | 2700.000000 | 2007 |
| 25% | 39.225000 | 15.600000 | 190.000000 | 3550.000000 | 2007 |
| 50% | 44.450000 | 17.300000 | 197.000000 | 4050.000000 | 2008 |
| 75% | 48.500000 | 18.700000 | 213.000000 | 4750.000000 | 2009 |
| max | 59.600000 | 21.500000 | 231.000000 | 6300.000000 | 2009 |

Profit More!

This is a Shinylive application embedded in a Quarto doc.

```
1  #| standalone: true
2
3  from shiny import *
4
5  app_ui = ui.page_fluid(
6      ui.input_slider("n", "N", 0, 100, 40),
7      ui.output_text_verbatim("txt"),
8  )
9
10 def server(input, output, session):
11     @output
12     @render.text
13     def txt():
14         return f"The value of n*2 is {input.n() * 2}"
15
16 app = App(app_ui, server)
```

- Shinylive Quarto extension repo: <https://github.com/quarto-ext/shinylive>
- Shinylive Quarto extension post: <https://quarto.org/docs/blog/posts/2022-10-25-shinylive-extension/>

Plots make people go WOO

```
1 #| standalone: true
2 #| viewerHeight: 420
3
4 from shiny import App, render, ui
5 import numpy as np
6 import matplotlib.pyplot as plt
7 app_ui = ui.page_fluid(
8     ui.layout_sidebar(
9         ui.panel_sidebar(
10            ui.input_slider("period", "Period", 0.5, 2, 1, step=0.5),
11            ui.input_slider("amplitude", "Amplitude", 0, 2, 1, step=0.25),
12            ui.input_slider("shift", "Phase shift", 0, 2, 0, step=0.1),
13        ),
14        ui.panel_main(
15            ui.output_plot("plot"),
16        ),
17    ),
18 )
```

Maps?

```
1 #| standalone: true
2 #| viewerHeight: 420
3
4 from htmltools import css
5 from shiny import App, reactive, render, ui
6 from shinywidgets import output_widget, reactive_read, register_widget
7
8 import ipyleaflet as L
9
10 app_ui = ui.page_fluid(
11     ui.div(
12         ui.input_slider("zoom", "Map zoom level", value=12, min=1, max=18),
13         ui.output_ui("map_bounds"),
14         style=css(
15             display="flex", justify_content="center", align_items="center",
16         ),
17     ),
18     output_widget("map"),
19 )
```

Shiny for Python!

“Interactive apps and dashboards made easy-ish”



Joe Cheng



Winston Chang

The screenshot shows the Shiny for Python homepage. At the top, there's a navigation bar with a back button, forward button, refresh button, and a URL field showing <https://shiny.rstudio.com/py/>. Below the navigation is a blue header with the text "Shiny for Python". The main content area has a title "Shiny for Python" and a subtitle "Shiny makes it easy to build interactive web applications with the power of Python's data and scientific stack." It lists three bullet points: "Approachable: Write your application in Python; no web development skills required.", "Flexible: Built from the ground up to support custom layouts, styles, and the modularity required for full-fledged applications, all from Python.", and "Performant: Uses reactivity to efficiently handle data processing and minimize expensive re-computations.". A red-bordered box contains a note: "Note! Shiny for Python is currently in Alpha. It may be unstable, and the API may change. We're excited to hear your feedback, but please don't use it for production applications just yet!". Below the note are two buttons: "Get Started" and "Examples". The "Examples" section displays three cards with titles: "Simulation" (with a 3D plot of a simulation trajectory), "How Does Regularization Strength Affect Coefficient Estimates?" (with a scatter plot of coefficient estimates), and "CPU Usage %" (with a heatmap of CPU usage). Each card has a "View app" and "View source" link below it.

Python...

Common error

```
1 $ quarto preview talk.qmd
2
3 Starting python3 kernel...Traceback (most recent call last):
4   File "/opt/quarto/share/jupyter/jupyter.py", line 21, in <module>
5     from notebook import notebook_execute, RestartKernel
6   File "/opt/quarto/share/jupyter/notebook.py", line 16, in <module>
7     import nbformat
8 ModuleNotFoundError: No module named 'nbformat'
9 Python 3 installation:
10    Version: 3.10.8
11    Path: /usr/bin/python3
12    Jupyter: (None)
13
14 Jupyter is not available in this Python installation.
15 Install with python3 -m pip install jupyter
```

Allison Horst

@allison_horst · Follow



(shivers) Happy Halloween, everybody 🖤🎃🕷🕸

12:44 PM · Oct 31, 2022

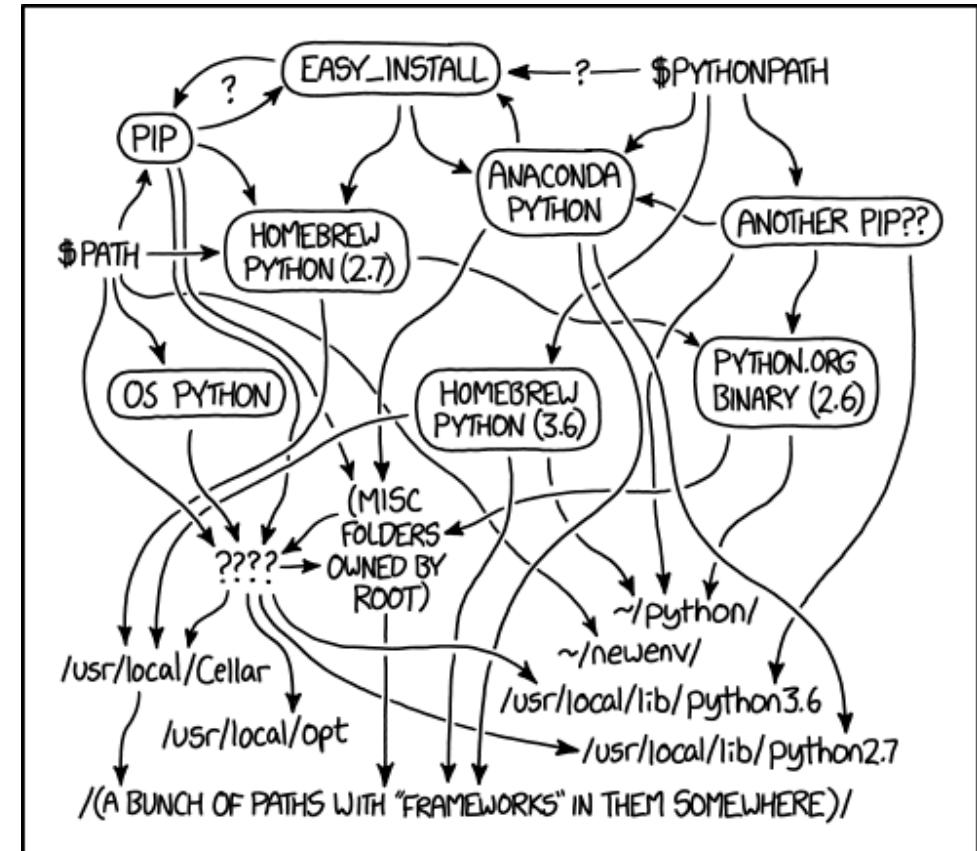
ⓘ

127

Reply

Copy link

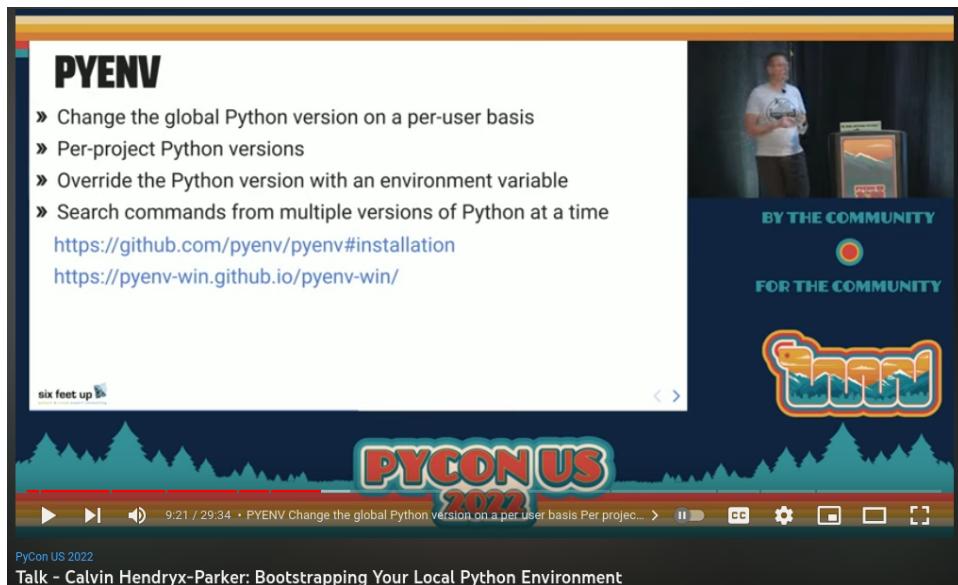
Read 6 replies



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Python Setup

Pyenv



Calvin Hendryx-Parker -
Bootstrapping Your Local
Python Environment

Anaconda



Virtual Environments

- Built-in Python 3.5+ `venv`
- `pyenv-virtualenv` plugin
- `pipenv`
- `conda`

Posit Academy

- Pyenv + pipenv

• Posit/Rstudio Minimum viable python: <https://solutions.rstudio.com/python/minimum-viable-python/>

UBC-MDS

- Anaconda based installation guide

Finding the binaries

- Make sure you are in the correct environment
 - `which python`
 - `pyenv versions`
- Check your Jupyter settings in Quarto
 - `quarto check`
- In VSCode
 - Python: Select Interpreter

The Jupyter kernel

- In your YAML:

```
1 jupyter: python3
```

- You do not need to “register” the kernel in your env

```
python -m ipykernel install --user --name  
myenv --display-name "Python (myenv)"
```

All the Things!

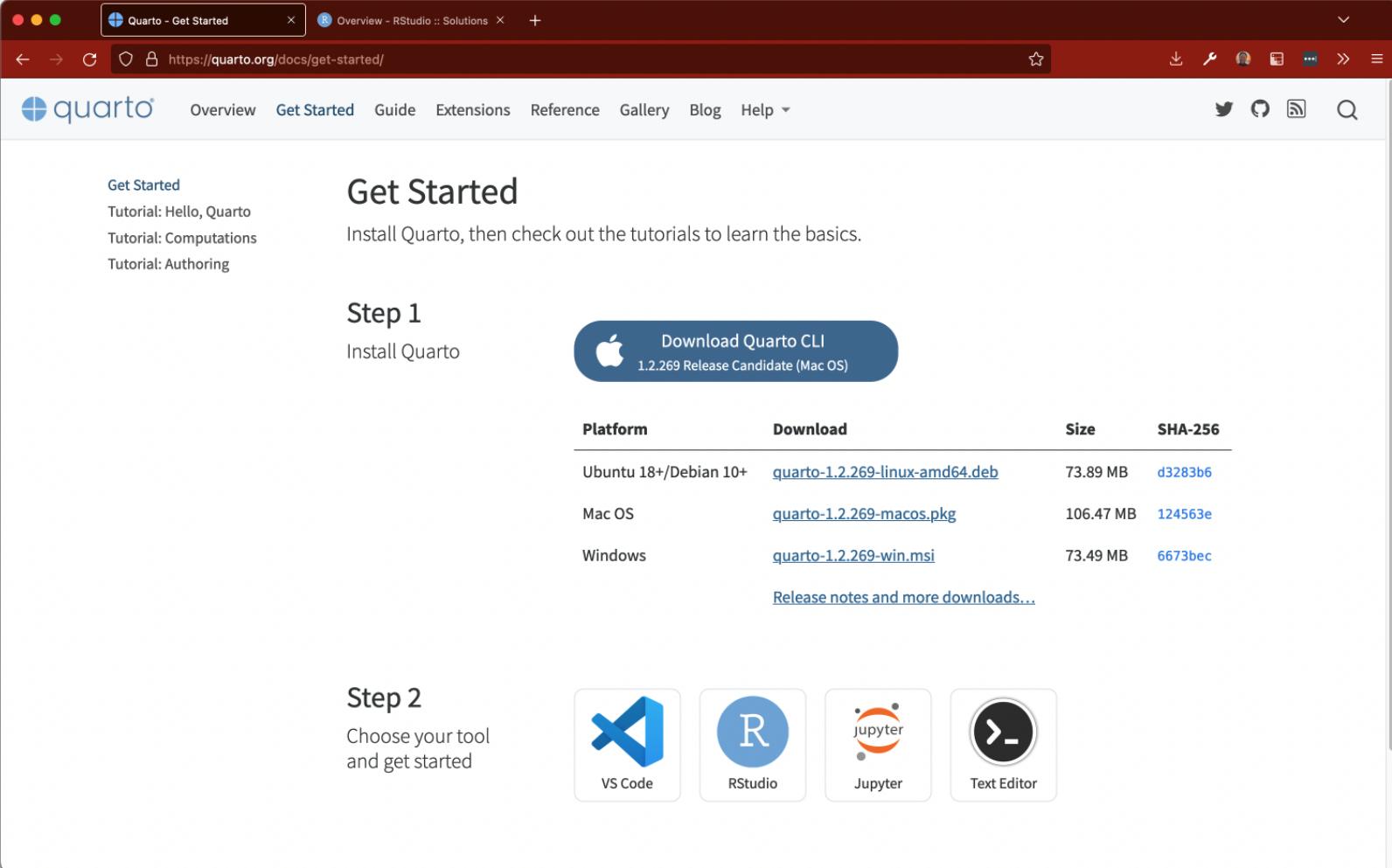
What can you do: Gallery

The screenshot shows a web browser displaying the Quarto Gallery page at <https://quarto.org/docs/gallery/>. The page features a large title "Gallery" and a sub-section "Articles & Reports". It highlights that Quarto can produce a wide variety of output formats, including HTML, PDF, and MS Word. A central feature is the "Visualization Curriculum" by Jeffrey Heer, Dominik Moritz, Jake VanderPlas, and Brock Craft, which is described as a data visualization curriculum of interactive notebooks. The curriculum page includes sections like "Introduction", "Getting Started", and "Local Installation". Below the curriculum, there's a section titled "Create multi-format books" with a dashed line separator. At the bottom, there are three tabs for "HTML for web publishing", "PDF for high quality print" (which is selected), and "MS Word for Office workflows".

<https://quarto.org/docs/gallery/>

Daniel Chen. @chendaniely. Using Quarto. Slides: https://github.com/chendaniely/pydata-nyc-2022-python_quarto

How can you do: Get Started + Guide



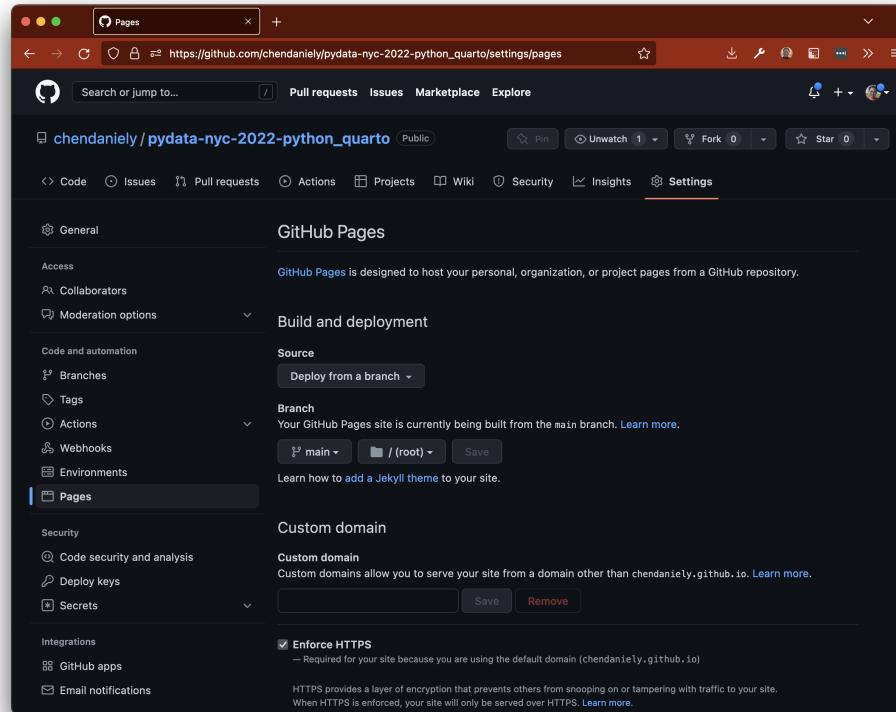
The screenshot shows a web browser window displaying the Quarto Get Started page at <https://quarto.org/docs/get-started/>. The page has a dark header with the Quarto logo and navigation links for Overview, Get Started, Guide, Extensions, Reference, Gallery, Blog, and Help. Below the header, there's a sidebar with 'Get Started' links: Tutorial: Hello, Quarto; Tutorial: Computations; and Tutorial: Authoring. The main content area is titled 'Get Started' with the sub-instruction 'Install Quarto, then check out the tutorials to learn the basics.' It features a 'Step 1' section for 'Install Quarto' with a 'Download Quarto CLI' button (1.2.269 Release Candidate for Mac OS). A download table lists packages for Ubuntu 18+/Debian 10+, Mac OS, and Windows. Below the table is a link to 'Release notes and more downloads...'. The 'Step 2' section, titled 'Choose your tool and get started', shows icons for VS Code, RStudio, Jupyter, and Text Editor.

| Platform | Download | Size | SHA-256 |
|-----------------------|--|-----------|-------------------------|
| Ubuntu 18+/Debian 10+ | quarto-1.2.269-linux-amd64.deb | 73.89 MB | d3283b6 |
| Mac OS | quarto-1.2.269-macos.pkg | 106.47 MB | 124563e |
| Windows | quarto-1.2.269-win.msi | 73.49 MB | 6673bec |

- <https://quarto.org/docs/get-started/>
- <https://quarto.org/docs/guide/>

Share: Github

- Build from a branch
- `index.qmd`

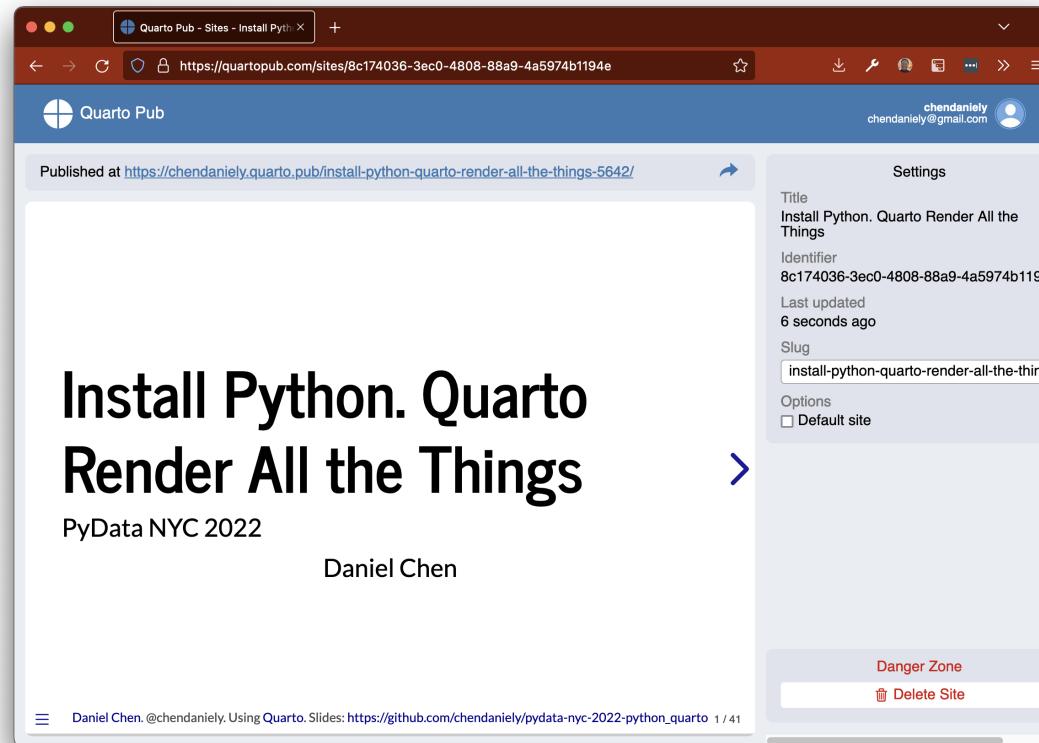


- Isabel's slides repo + auto build: <https://github.com/isabelizimm/pydata-nyc2022>

Daniel Chen. @chendaniely. Using Quarto. Slides: https://github.com/chendaniely/pydata-nyc-2022-python_quarto

Share: Quartopub

- <https://quartopub.com/>
- `quarto publish quarto-pub index.qmd`



Learn more

- Tom Mock: Getting Started with Quarto workshop
- Allison Hill: The Happiest Notebooks on Earth
- Gordon Shotwell: An overview of Quarto, and Jupyter
- Mine Çetinkaya-Rundel & Julia Stewart Lowndes: Hello Quarto: Share, Collaborate, Teach, Reimagine

Try Quarto!

1. <https://quarto.org/>
2. Get Started (aka install)
3. Guides (Pick a project)
4. Website: `quarto create-project mysite --type website`
5. `quarto preview mysite`
6. Profit?