Information Technology Agency
City of Los Angeles

Data + Donuts 12/10/19

Tiffany Chu and Ian Rose

Citywide Data Science and Predictive Analytics

Partner with various City departments to do multi-departmental data analytics projects

- Transportation
- Office of Finance
- Street Services
- Sanitation
- Housing + Community Investment Development
- City Planning
- Mayor's initiatives

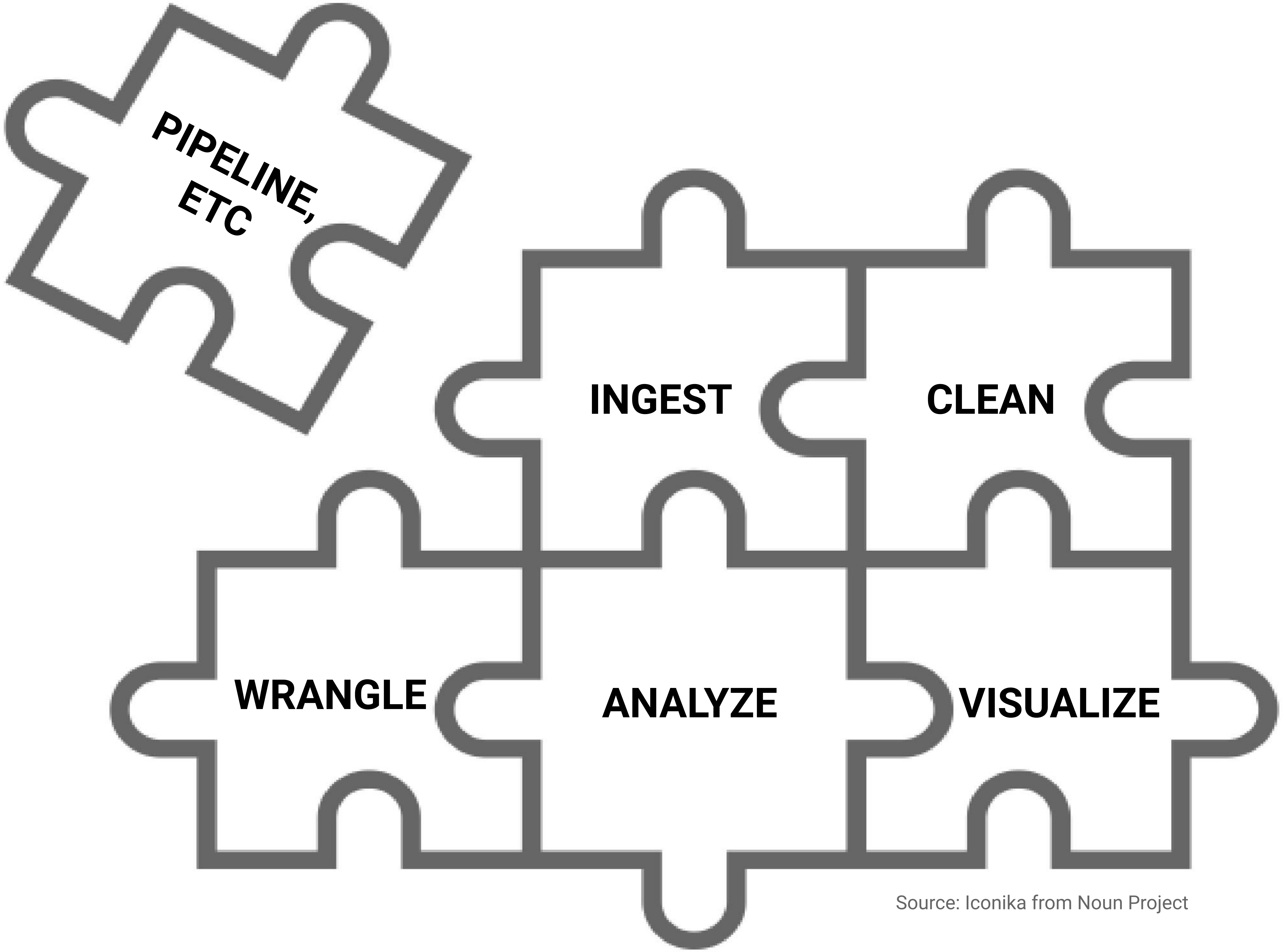
Best Practices

Our hard-earned lessons and standard we're striving for:

- Goal: Reproducibility
- GitHub
- Data Pipelines
- Data Management
- Shared platform for analysis (JupyterHub)
- Tutorials

https://cityoflosangeles.github.io/best-practices/

Data Analysis

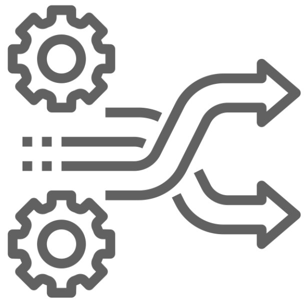


Source: Iconika from Noun Project

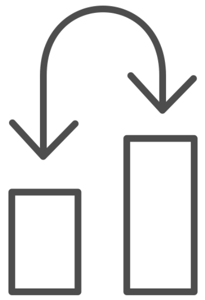
Motivation

- What's the research question?
  - Policy interventions, factors / mechanisms, outcomes
- So many datasets, mix & match
- Merge data to compare and see what's going on

Observe trends



Current resource allocation vs "need"-driven allocation



Normalize by population or area

Sources: Sachin Modgekar, Nithinan Tatah, and Andriwidodo, from Noun Project

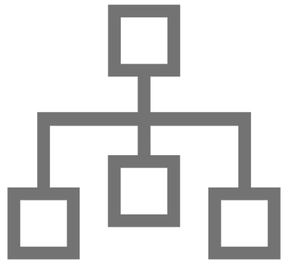
Ingest and Clean

- Multiple data sources
- Identify the unit of analysis

Geography
tract, council district

Time
year, quarter

Other Category
treatment, intervention



Mixed
tract-year,
tract-treatment-year

- Clean each data source to the common unit (least common factor)
    - Aggregate
    - Spatial join + dissolve

Sources: Randomhero, Alice Design, Calvin Goodman, and Fahmi from Noun Project

Merge

- **one-to-one (1:1)**
  merge on GEOID and Year
  GEOID and Year appear once in both dfs

collisions

GEOID	Year	# collisions
A	2018	10
A	2019	8
B	2018	15

traffic volume

GEOID	Year	traffic vol
A	2018	2,000
A	2019	1,500
B	2018	2,500

- **many-to-one (m:1)**
  merge on GEOID
  GEOID appears multiple times in left df; once in right df

collisions

GEOID	Year	# collisions
A	2018	10
A	2019	8
B	2018	15

GEOID characteristics

GEOID	area (sqmi)
A	5
B	4

- **one-to-many (1:m)**
  merge on Year
  Year appears once in left df; multiple times in right df

collisions by year

Year	# collisions
2018	30
2019	40

traffic volume by quarter

Year	Qtr	traffic vol
2018	1	500
2018	2	700
2019	1	450

Crosswalks

- Correspondence tables that link elements together

tracts to council districts crosswalk
lists the CD associated with each tract

GEOID	CD
A	1
B	5
C	14
D	5

streets to tracts crosswalk
lists the tract associated with each street segment

Segment	Tract
Wilshire1	1000
Main4	2000
Hoover2	3000
Pico5	4000

- Spatial join + clipping to create crosswalk
- Facilitates the merging and aggregating across multiple dfs

Putting it Together

df1 = pd.merge(collisions, traffic_volume, on = ['GEOID', 'Year'],
how = 'inner', validate = '1:1')

collisions

GEOID	Year	# collisions
A	2018	10
A	2019	8
B	2018	15
C	2017	20

traffic volume

GEOID	Year	traffic vol
A	2018	2,000
A	2019	1,500
B	2018	2,500
C	2017	2,500

df1

GEOID	Year	# collisions	traffic vol
A	2018	10	2,000
A	2019	8	1,500
B	2018	15	2,500
C	2017	20	2,500

1:1 merge on GEOID, Year

Putting it Together

```
df2 = pd.merge(df1, tract_characteristics, on = 'GEOID',
               how = 'inner', validate = 'm:1')
```

df1

GEOID	Year	# collisions	traffic vol
A	2018	10	2,000
A	2019	8	1,500
B	2018	15	2,500
C	2017	20	2,500

tract characteristics

GEOID	area
A	5
B	4
C	6

df2

GEOID	Year	# collisions	traffic vol	area
A	2018	10	2,000	5
A	2019	8	1,500	5
B	2018	15	2,500	4
C	2017	20	2,500	6

m:1 on GEOID

Putting it Together

```
df3 = pd.merge(df2, crosswalk, on = 'GEOID',
               how = 'inner', validate = 'm:1')
```

df2

GEOID	Year	# collisions	traffic vol	area
A	2018	10	2,000	5
A	2019	8	1,500	5
B	2018	15	2,500	4
C	2017	20	2,500	6

crosswalk

GEOID	CD
A	1
B	5
C	14

df3

GEOID	Year	# collisions	traffic vol	area	CD
A	2018	10	2,000	5	1
A	2019	8	1,500	5	1
B	2018	15	2,500	4	5
C	2017	20	2,500	6	14

m:1 on GEOID

df3 is the merged combination of collisions, traffic_volume, tract_characteristics, and crosswalk

Additional Resources

- Merging dfs
  - https://guides.nyu.edu/quant/merge
    (Tableau, SPSS, JMP, Stata, SAS, R, Matlab, Python)
  - https://www.shanelynn.ie/merge-join-dataframes-python-pandas-index-1/
  - https://towardsdatascience.com/why-and-how-to-use-merge-with-pandas-in-python-548600f7e738

- Clip congressional_districts to City of LA boundary and create new geometry column
  - df = gpd.sjoin(congressional_districts, city_boundary, how = ‘inner’, op = ‘intersects’)
  - boundary = city_boundary.geometry.iloc[0]
  - df[‘new_geom’] = df[df.intersects(boundary)].intersection(boundary)

Infrastructure for Civic Data Teams

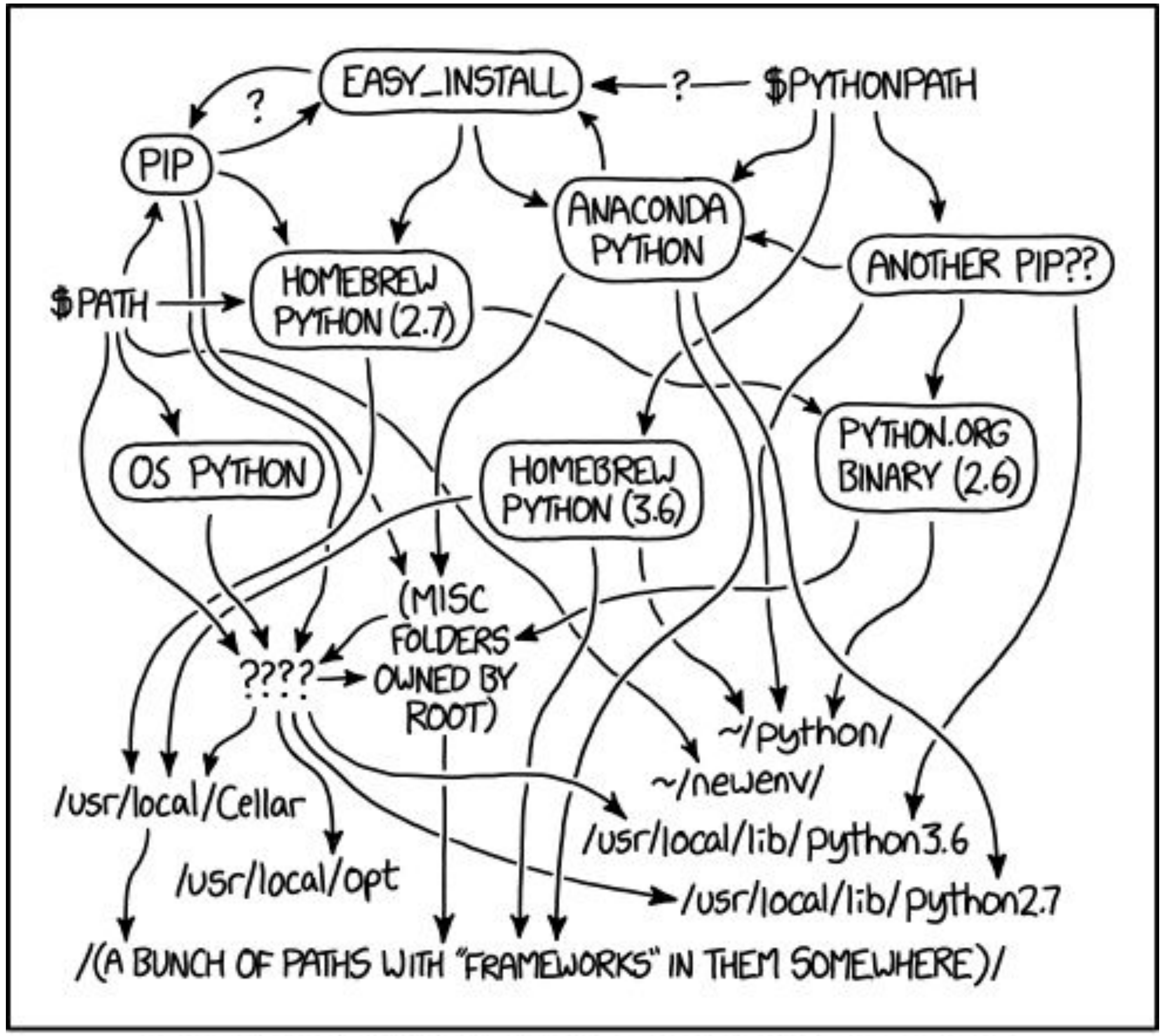
Data analysis is hard: how do we set up the infrastructure for our analysts to spend more time doing that, and less time fighting their hardware/software?

Best Practices: Reproducible Environments

Reproducibility is an unsolved problem in science!

Tools that can help:

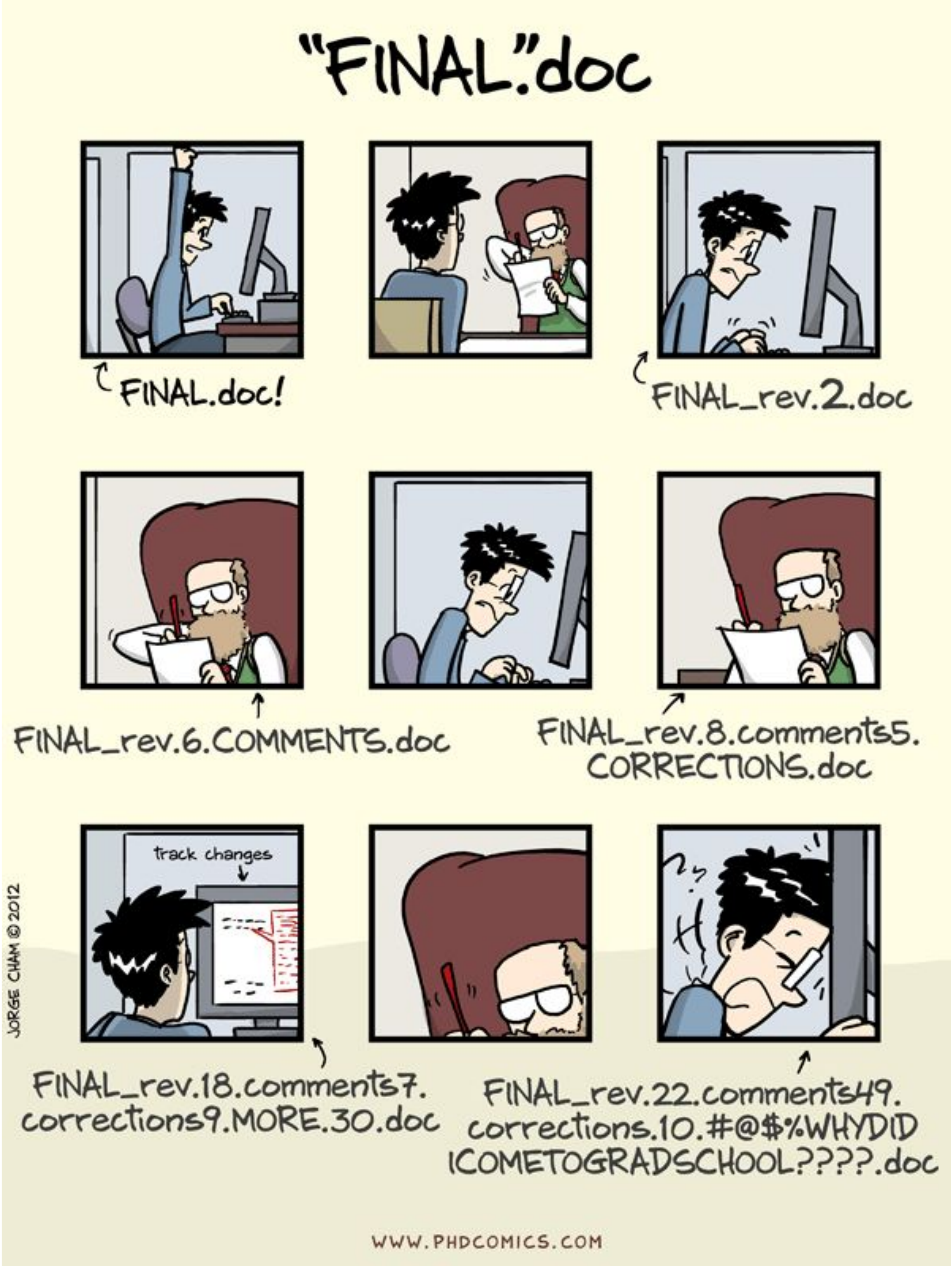
- Documentation, documentation, documentation
- Dockerfiles / requirements.txt / environment.yml
- Shared cloud compute
- CI/CD



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

Explication of this comic by Brett Cannon: https://snarky.ca/deconstructing-xkcd-com-1987/

Best Practices: Version Control



git

GitHub

Best Practices: Continuous Integration

AKA: “catch errors before they are a problem”

Travis CI

circleci

| Cloud-Native Infrastructure

AKA "someone else's computer"

Pros:

- Scalable
- Reproducible
- Reliable

Cons:

- Can have a steep learning curve
- Networking can be difficult
- "Oops, my cluster is down, now I can't work."

aws

Microsoft Azure

Google Cloud

| Cloud-Friendly Formats

Use file formats that are open, standardized, and easy to share on cloud resources:

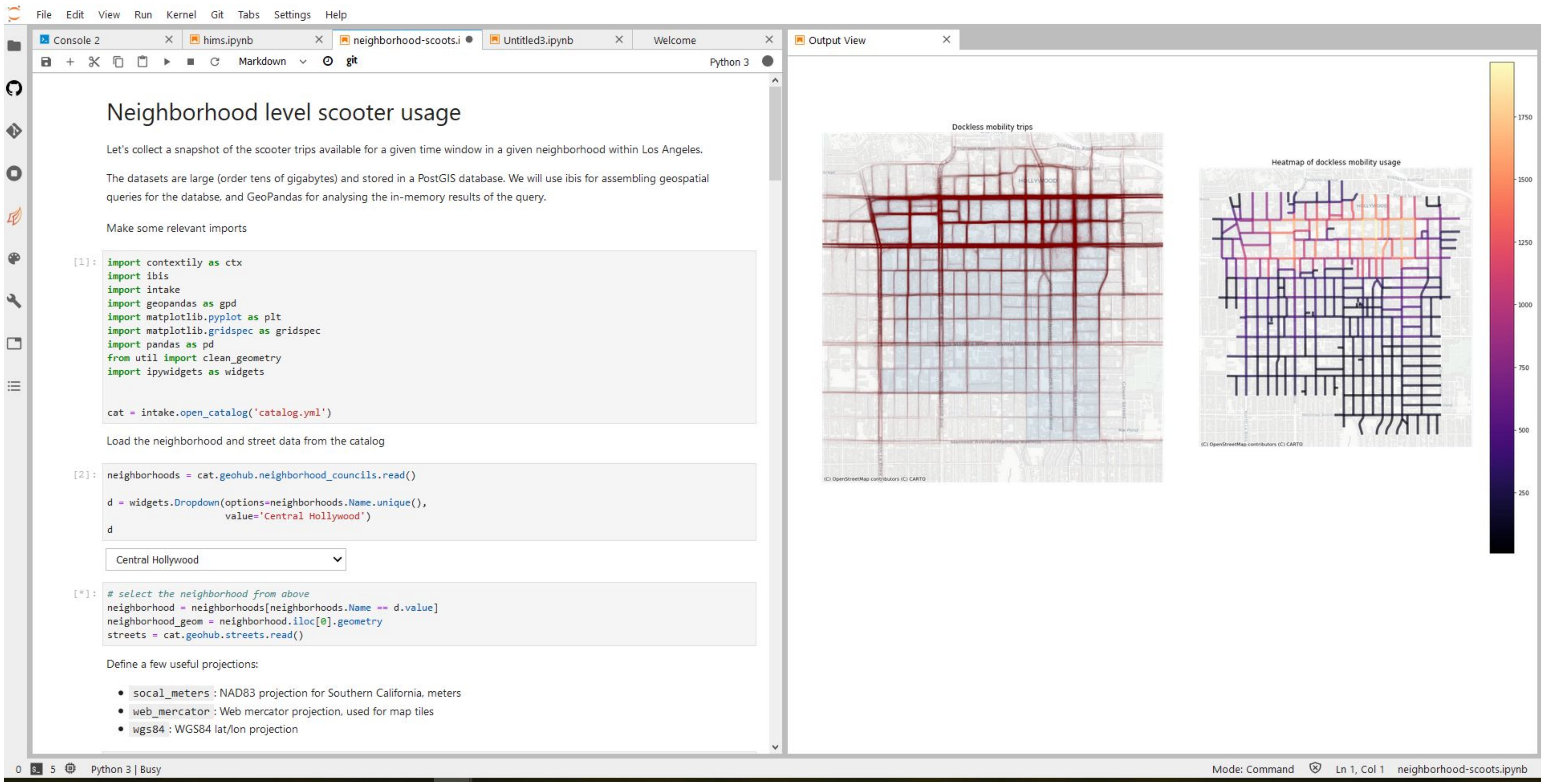
- Parquet
- GeoJSON
- GeoTIFF
- Zarr
- CSV*
- Shapefile*

* Caveats apply

Try to avoid:

- Excel (cf. The Excel Error that Changed History)
- PDF (please please please)
- ../../ian-rose/files/some-forgotten-file.csv

Literate computing: Jupyter and RMarkdown



| A Civic Data Science Tech Stack

Analysis Environment

RStudio lab

Data Analysis

GDAL matplotlib pandas tidyverse

Data Access

Socrata esri Amazon S3 PostgreSQL

Infrastructure

docker kubernetes Google Cloud aws HTML 5

| Thanks for listening!

Questions?

ITA