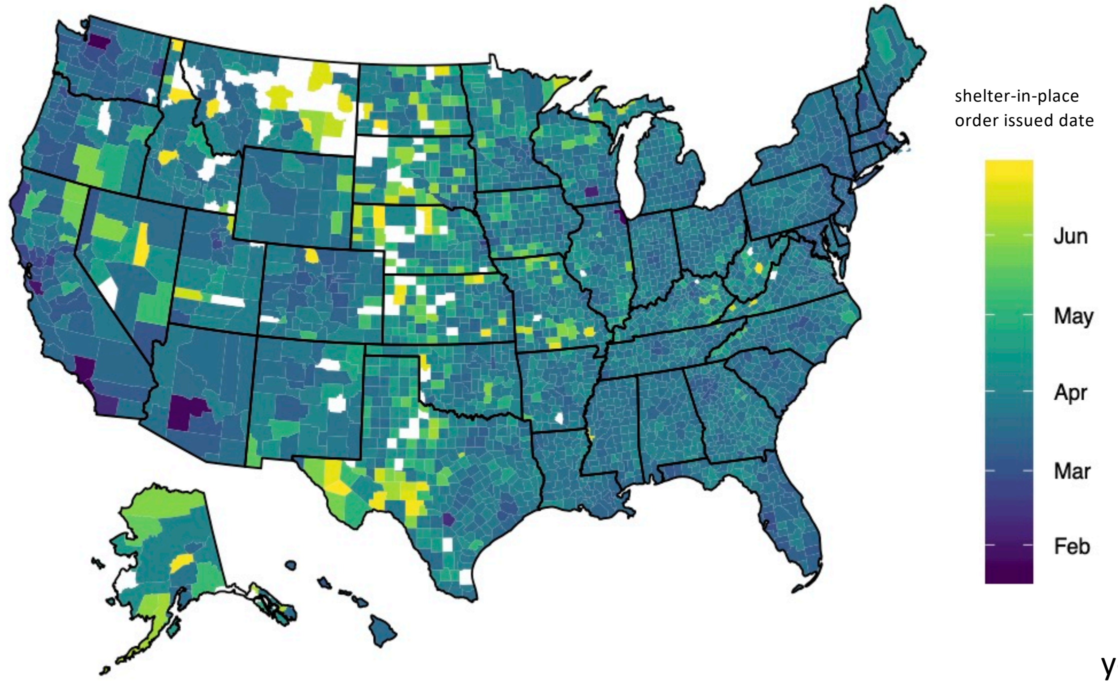
S&DS 177
YData: COVID-19 Behavioral Impacts



Youpei Yan
youpei.yan@yale.edu

Lecture 5 (Mar 4, Thursday 9:25 – 11:15)

Overview:

- Select rows based on column values in pandas
- Plot and compare various datasets to draw plausible conclusions
- Create maps! (But not this cute)
  - Install Plotly
  - import plotly.express as px

Lecture 5 (Mar 4, Thursday 9:25 – 11:15) 1. Data selection in Pandas

- Why do we want to select certain data within the dataset?

It can have various reasons due to the purpose of the study.
Try to explain why the research might be interesting when:

- We want to focus on the epidemic performance in certain states.
- We want to focus on the states with really high cases or high death rates.
- Only states with high average trips to stores matter to our study.
- Only high precipitation counties matter because we suspect humidity influence transmission.
- Etc.

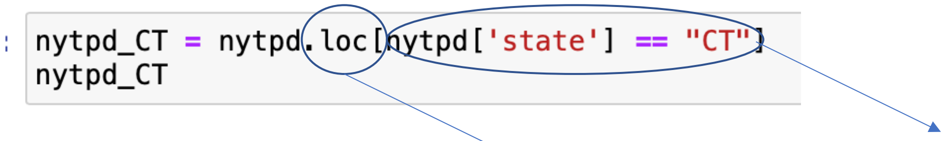
Lecture 5 (Mar 4, Thursday 9:25 – 11:15) 1. Data selection in Pandas

- Why do we want to select certain data within the dataset?

It can have various reasons due to the purpose of the study.
Try to explain why the research might be interesting when:

- We want to focus on the epidemic performance in certain states.
- We want to focus on the states with really high cases or high death rates.
- Only states with high average trips to stores matter to our study.
- Only high precipitation counties matter because we suspect humidity influence transmission.
- Etc.

- Recall what we did to select rows based on values using Numpy.
- Here is a new function to select using Pandas.



```
nytpd_CT = nytpd.loc[nytpd['state'] == "CT"]
nytpd_CT
```

.loc function

Select rows with state equals to CT

Lecture 5 (Mar 4, Thursday 9:25 – 11:15) 1. Data selection in Pandas

- Why do we want to select certain data within the dataset?

It can have various reasons due to the purpose of the study.
Try to explain why the research might be interesting when:

- We want to focus on the epidemic performance in certain states.
- We want to focus on the states with really high cases or high death rates.
- Only states with high average trips to stores matter to our study.
- Only high precipitation counties matter because we suspect humidity influence transmission.
- Etc.

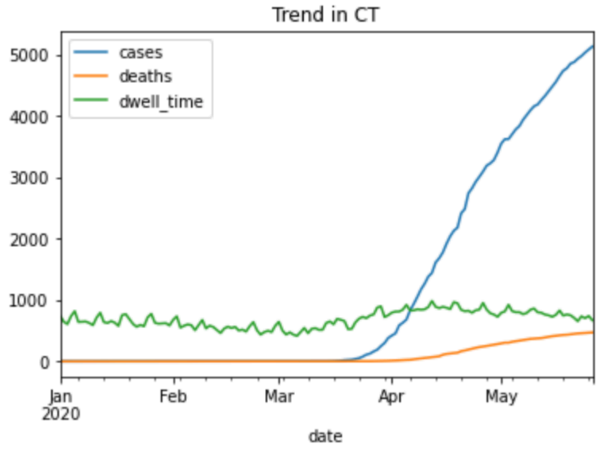
- Modify the .loc function to create the above 4 datasets.
- For cases above a certain level (say 500), we can use:

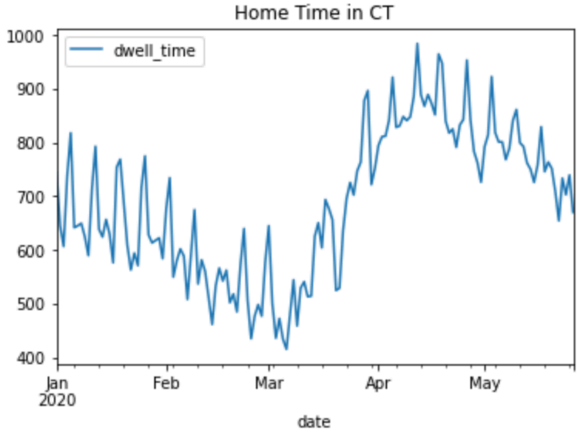
```
df.loc(df[‘cases’] > 500)
```

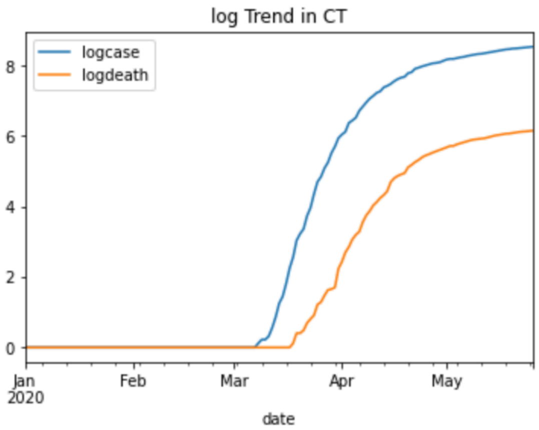
where df is the dataframe’s name, such as *nytpd*.

Lecture 5 (Mar 4, Thursday 9:25 – 11:15) 2. More visualization under Pandas

- Figures can provide more information than staring at tables. But two things we need to pay attention to







- The magnitude, or scale of the data, can be easily modified. It looks like the curve is flatten for home dwelling time in figure 1, is this true if we look at figure 2?
- The transformation of data matters more. Cases and deaths are increasing in an exponential way at the beginning. If we converted to the log form, we can see a linear increase at the beginning instead.
- What else can you tell from the figure?

Lecture 5 (Mar 4, Thursday 9:25 – 11:15) 2. More visualization under Pandas

- Using only cases, deaths, and dwelling time at home to explain each other are not a good way to find a correct relationship of cause-and-effect.

- What else may influence home dwelling time on average?
  - Import weather data and plot to see if temperature or precipitation has any co-movement with our home dwelling time data.
  - Import POI data and plot some industries to see if it has any co-movement with case reports.

What are the steps to do this?

1. Import data. – which function under pandas?
2. Aggregate by group, because figures can only show 2-dimention information, but our panel data provides multi-dimension information. - which function?
3. Plot with x-axis as date. – Did you convert date from string to date first?
4. What can you tell from the figure?

Lecture 5 (Mar 4, Thursday 9:25 – 11:15) 3. Creating maps

- Install new module V.S. Import

  Python is an open-source programming language. Some functions are built-in and we can use them directly. But Python has an active supporting community of contributors and users that also make their software available for other Python developers to use under open source license terms. If someone writes a new module for us to do a group of things easier, all we need to do is installing the user-written module, then import.

- Basic Usage (Reference: https://packaging.python.org/tutorials/installing-packages/)
  - pip install or conda install
    - Conda and pip are often considered as being nearly identical. Although some of the functionality of these two tools overlap, they were designed and should be used for different purposes.

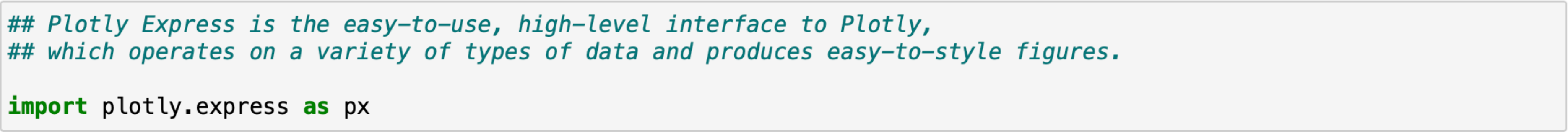
    - Pip is the Python Packaging Authority’s recommended tool for installing packages from the Python Package Index, PyPI. Pip installs Python software packaged as wheels or source distributions. The latter may require that the system have compatible compilers, and possibly libraries, installed before invoking pip to succeed.

    - Conda is a cross platform package and environment manager that installs and manages conda packages from the Anaconda repository as well as from the Anaconda Cloud. Conda packages are binaries. There is never a need to have compilers available to install them. Additionally conda packages are not limited to Python software. They may also contain C or C++ libraries, R packages or any other software.

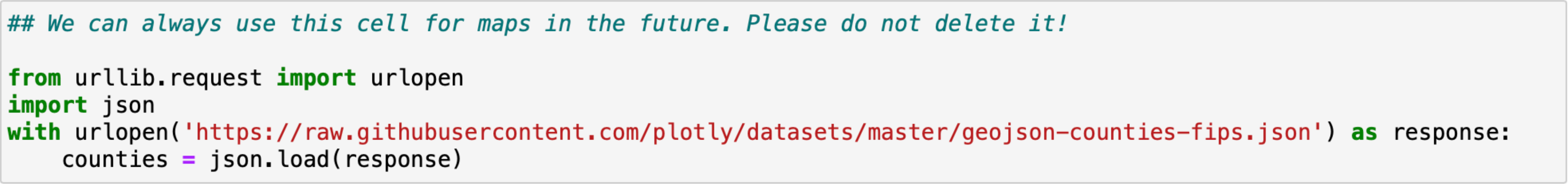
Lecture 5 (Mar 4, Thursday 9:25 – 11:15) 3. Creating maps

- Currently, we don’t have to know any of these differences. We simply need to know that we want to **intall plotly** first before creating beautiful maps.

- All the required code can be pasted to your own file (please don’t change any lines), and we borrow it directly for our own dataset to plot!



```
## Plotly Express is the easy-to-use, high-level interface to Plotly,
## which operates on a variety of types of data and produces easy-to-style figures.

import plotly.express as px
```



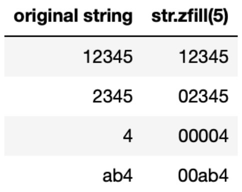
```
## We can always use this cell for maps in the future. Please do not delete it!

from urllib.request import urlopen
import json
with urlopen('https://raw.githubusercontent.com/plotly/datasets/master/geojson-counties-fips.json') as response:
    counties = json.load(response)
```

- Basically, it provides a connection between our geoid (5 digit county code) to a county map.
- With just one line of code, we can change the colors of a county map to be based on the values in our dataset.

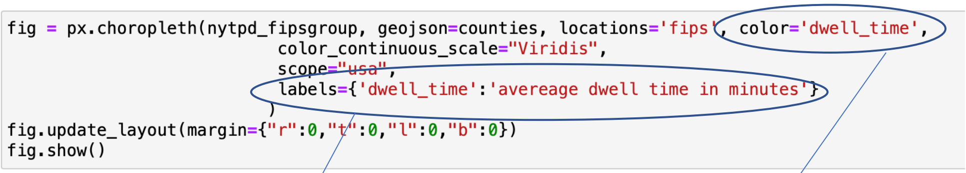
Lecture 5 (Mar 4, Thursday 9:25 – 11:15) 3. Creating maps

- Generate readable FIPS code for Python to process:
  - Convert geoid from numerical values to string
  - Fill the 4 digits to 5 digits with 0 at front.

`str.zfill(#)` fills strings to # digits. If the string is less than # digits, it fills zero(s) in front of it. Here are some quick examples for `str.zfill(5)`



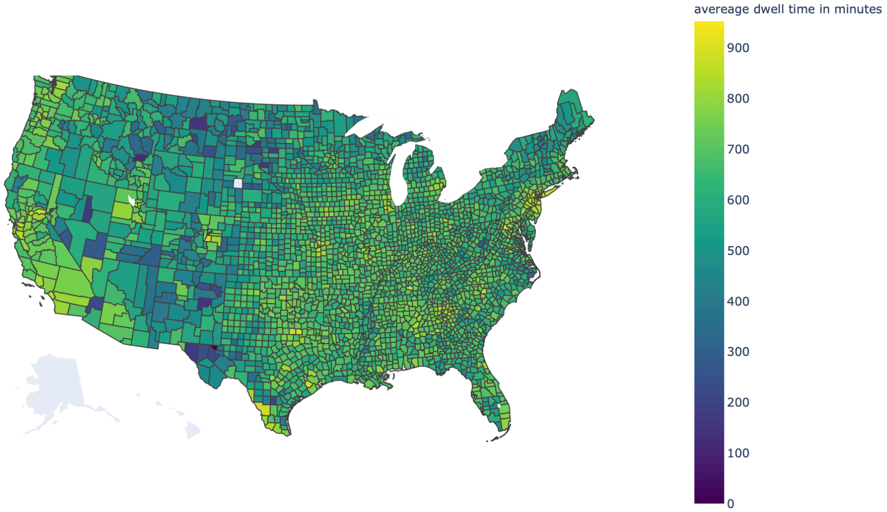
original string	str.zfill(5)
12345	12345
2345	02345
4	00004
ab4	00ab4



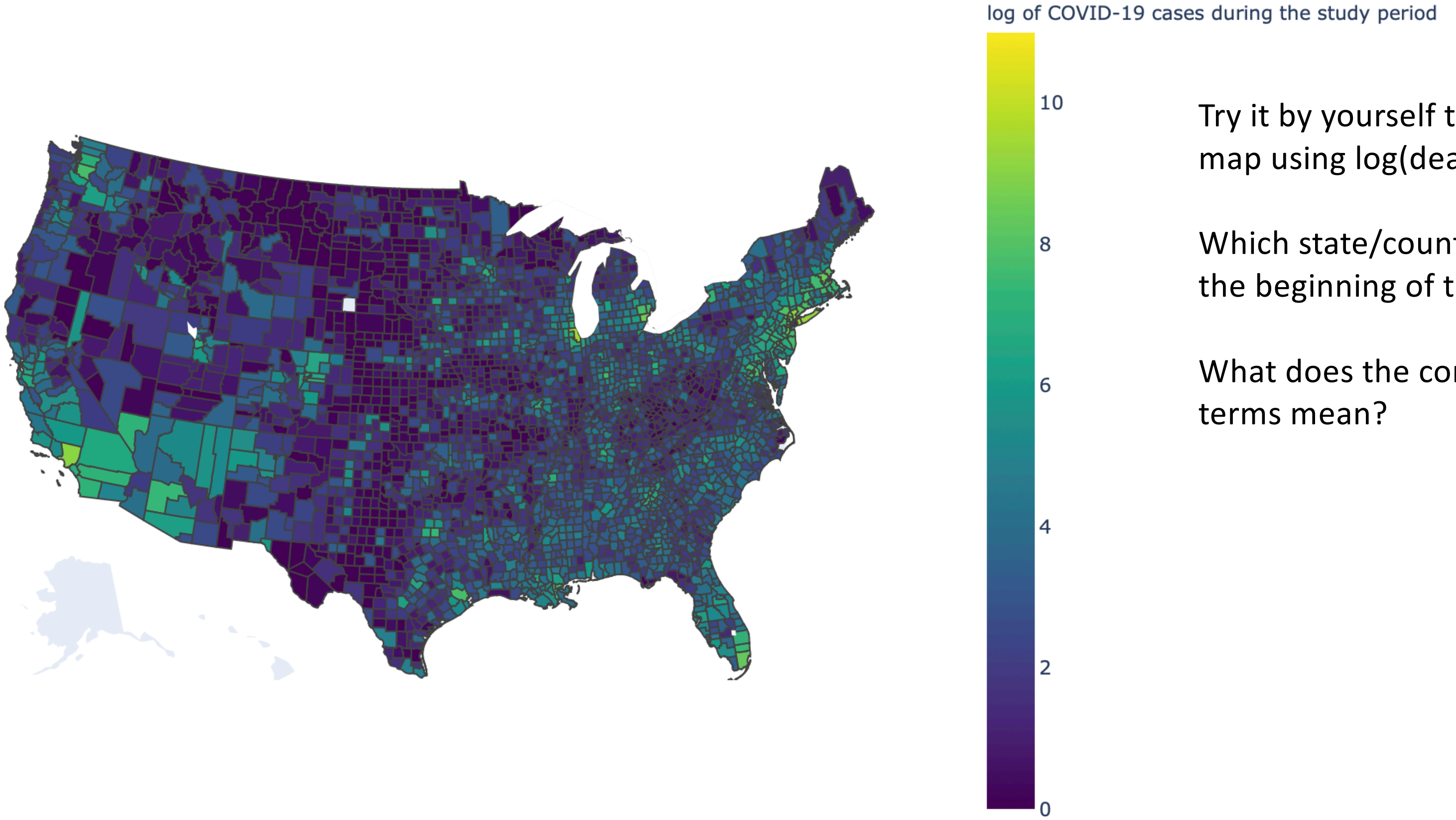
```
fig = px.choropleth(nytpd_fipsgroup, geojson=counties, locations='fips', color='dwell_time',
                           color_continuous_scale="Viridis",
                           scope="usa",
                           labels={'dwell_time':'avereage dwell time in minutes'}
                          )
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```

Change here to our designed variable

Also modify the labels accordingly.



Lecture 5 (Mar 4, Thursday 9:25 – 11:15) 3. Creating maps



Try it by yourself to create a similar map using log(death + 1).

Which state/county is worrisome at the beginning of the pandemic?

What does the comparison in log terms mean?

Lecture 5 (Mar 4, Thursday 9:25 – 11:15) 3. Creating maps

- Are we plotting a panel data result? Or we are plotting a cross-sectional result?

- Try to modify the code to show a group of maps reflecting average case reports in multiple time periods

- To do so:

1. Import raw csv data and modify geoid to FIPS and save it as the original panel dataset.
2. Convert cases value to log(cases + 1) to better reflect the data range.
3. Select date(1) from the original panel dataset into a new dataset(1).
4. Plot the map for this date.
5. Repeat step 3 & 4 with different dates.