Data Visualization & Geospatial Analysis

COGS 108 Fall 2025
Jason Chen
Week 4

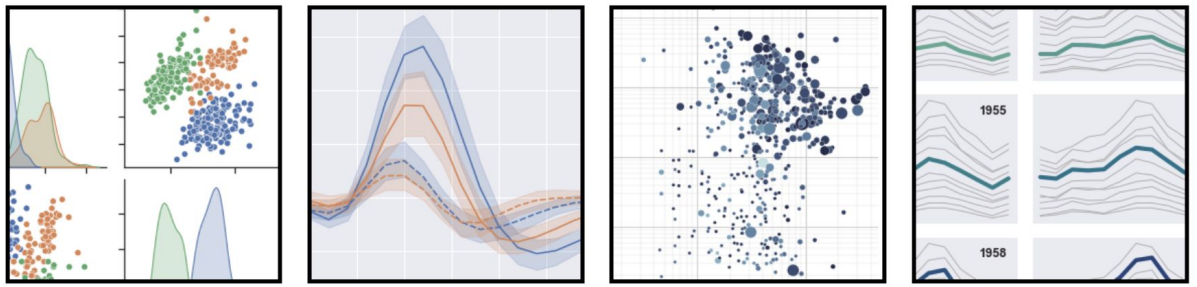
xic007@ucsd.edu

OH: Tue 3-5 pm

Due dates

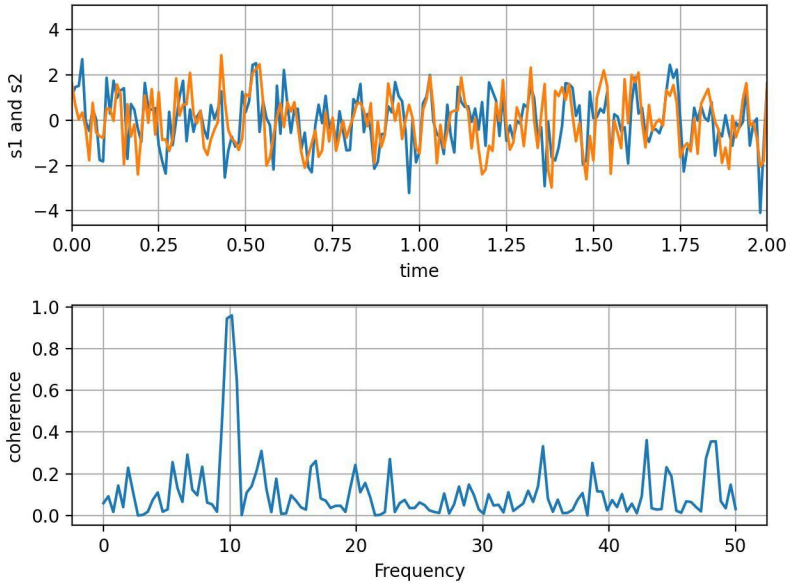
- Q3: Monday (10/20)
- A1 and Project Review: Wednesday (10/22)
- D3: Friday (10/24)

Pyplot and seaborn



Matplotlib (plt)

- A plotting library for Python
- Makes static, animated, and interactive visualizations in Python.
- Usually imported under the plt alias



https://matplotlib.org/stable/tutorials/index

There are so many ways to make a same plot!

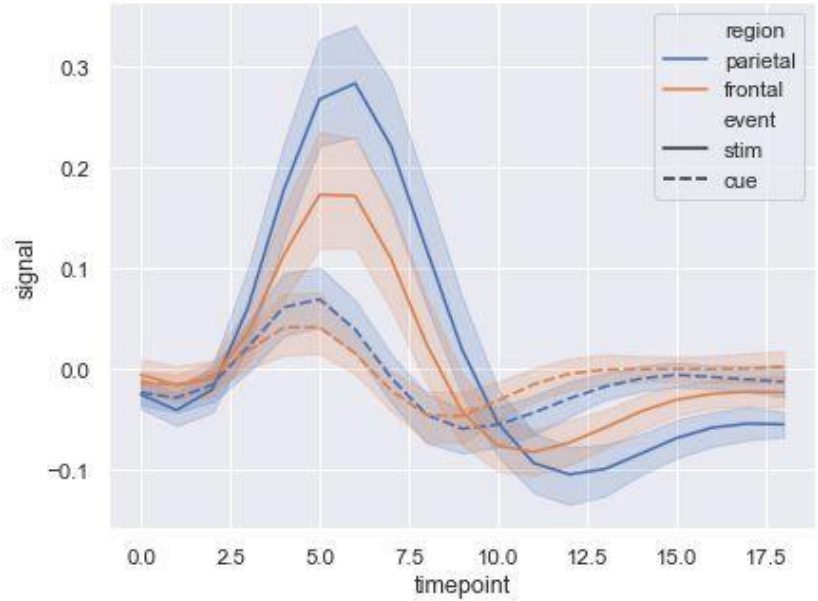
- All of these do the same thing:
  - Line 1: plt.hist(df['income'], 25)
  - Line 2: df['income'].hist(bins=25)
  - Line 3: df.hist('income', bins=25)

- In Python, most image-based plots created using Matplotlib (plt), e.g., plt.hist, plt.bar, plt.plot, etc.

- Pandas gives shortcuts for matplotlib plots. Lines 2 and 3 are shortcuts for line 1.

Seaborn

- Makes common statistical charts easy to create, like bar plots with confidence intervals.
- Again, seaborn is really just a bunch of shortcuts for matplotlib.
- Usually imported under the sns alias



https://seaborn.pydata.org/tutorial.html

It’s time to …

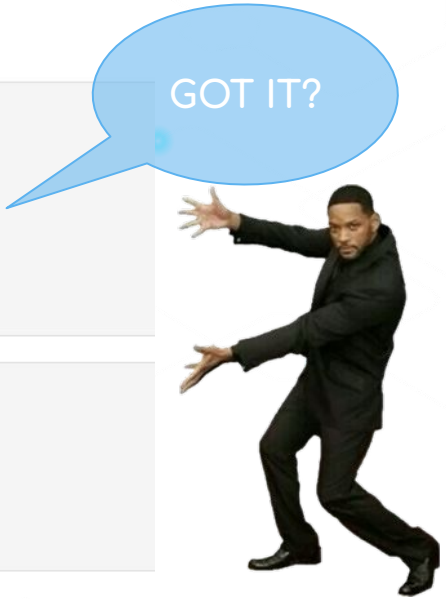
- Go over some demos of D3!

Import all the necessary libraries

- `numpy` (`np`)
- `pandas` (`pd`)
- `matplotlib.pyplot` (`plt`)
- `seaborn` (`sns`)

[1]:
```
# YOUR CODE HERE
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
assert np
assert pd
assert plt
assert sns
```



Part I : Cheating

```
feature_counts =
dataFrame['feature'].value_counts()
```

`df['your_column'].value_counts()` - this will return the count of unique occurences in the specified column.

It is important to note that `value_counts` only works on pandas series, not Pandas dataframes. As a result, we only include one bracket df['your_column'] and not two brackets df[['your_column']].

Parameters

- **normalize (bool, default False) -** If True then the object returned will contain the relative frequencies of the unique values.
- **sort (bool, default True) -** Sort by frequencies.
- **ascending (bool, default False) -** Sort in ascending order.
- **bins (int, optional) -** Rather than count values, group them into half-open bins, a convenience for `pd.cut`, only works with numeric data.
- **dropna (bool, default True) -**Don't include counts of NaN.

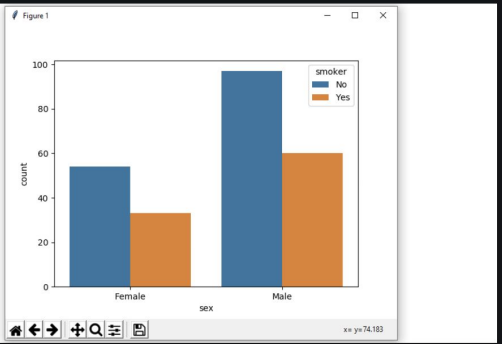
Part I : Cheating

sns.countplot(x, y, hue, data=df);

Python3

```
# importing the required library

import seaborn as sns
import matplotlib.pyplot as plt

# read a tips.csv file from seaborn library
df = sns.load_dataset('tips')

# count plot on two categorical variable
sns.countplot(x ='sex', hue = "smoker", data = df)

# Show the plot
plt.show()
```

Output :



Part I : Cheating

create a DataFrame prop_df with three columns, one for gender, one for cheated, and one including the proportion of respondents who cheated within each gender

```
prop_df = (survey['cheated']
        .groupby(...)
        .value_counts(normalize=True)
        .rename(...)
        .reset_index())
```

Part I : Cheating

Regenerate your barplot using the proportion data you just generated to determine which gender cheats more frequently.

Assign your seaborn plot to a variable named plot_proportion

```
plot_proportion = sns.barplot(x=' ',
        y=' ',
        hue=' ',
        data=dataFrame);
```

Y axis is cheated
Y axis is proportion
The hue is the gender

Swapping: include
hue_order=["Male","Female"],

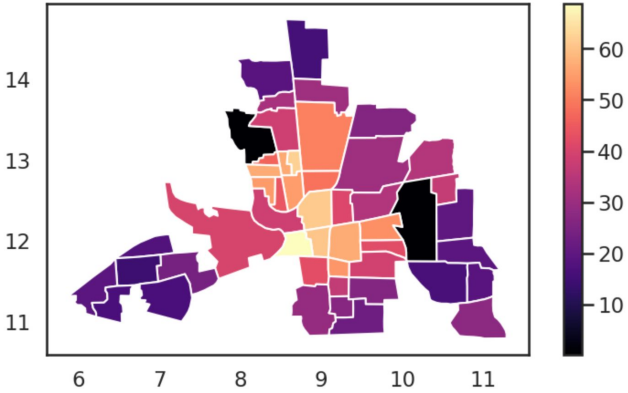
Part IV : Geospatial Graphs

Import geospatial data from file_name.shp:

```
pth =
ps.examples.get_path("file_name.shp")
data =
gpd.GeoDataFrame.from_file(pth)
```

Plot geospatial data with var visualized with color

```
columbus_data.plot(column = 'var',
cmap = …,  legend = True)
```



Section Materials

Section materials can be accessed at:
https://github.com/JasonC1217/COGS108_FA25_B07-B08

THANKS!

Questions on EdStem or office hours

Office hours: Tue, 3-5 PM