DATA SCIENCE IN PRACTICE

Discussion Section - Week 4

Slide Credit: Akshay N (FA22)

AGENDA FOR TODAY

01 ANNOUNCEMENTS

02 DEADLINES/DATES

03 PROJECT REVIEW

04 D3

Note: Section A05 is podcasted!

DEADLINES

DUE DATES

- Quiz 3 is due Oct 23, 11:59PM
- Project Review is due Oct 25, 11:59PM (Wednesday)
- Discussion lab 3 is due Oct 27, 11:59PM (Friday)

COMING UP

- Project Proposal is due next Wednesday (11/1)

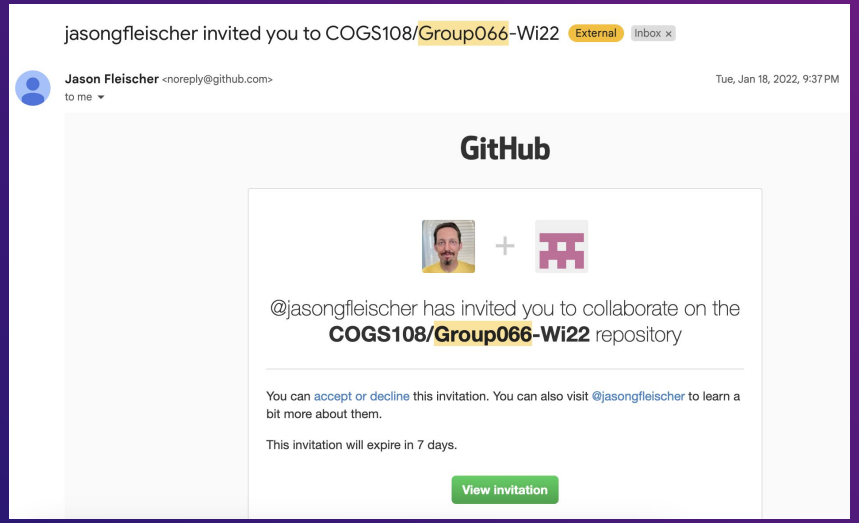
ANNOUNCEMENTS

Project Updates:

- You have all been assigned a Group and Github Repo
- IMPORTANT: If you do NOT have access to your Group Repo, you need to fill out the Github Username Quiz ASAP
- Github Repos are under the Cogs108 account on Github (check your notifications)
- Groups are also on Canvas => People
  - https://canvas.ucsd.edu/groups

ANNOUNCEMENTS

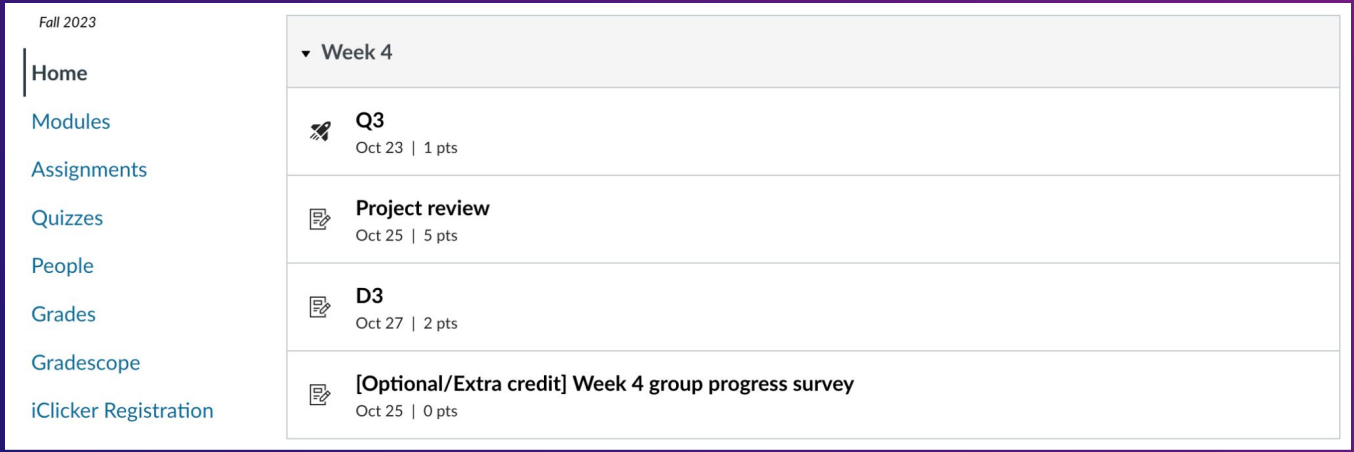
- Project Groups have been released and invites have been sent
  - PLEASE accept the invite as soon as possible (expires on Tuesday)



Looks like this!

PROJECT REVIEW

- Due: Wednesday (10/25)
- One submission per group



PROJECT REVIEW

- PLEASE reach out to your Group
- You should have all received an email with your Group info
- You can also find your group on Canvas or respond via the Discussion

Final Project Group -- COGS108_FA23

File Edit View Insert Format Data Tools Extensions Help

100% View only

D2

	A	B
1	**Group Name**	**Group Member**
2	Group001_FA23	['bfernandez', 'dam001', 'kkray', 'asagundo', 'shz054']
3	Group002_FA23	['kgugnani', 'rpeng', 'srote', 'e7song', 'e2sun']
4	Group003_FA23	['kekhlassi', 'pking', 'spuri', 'arsureshkumar', 'liz009']
5	Group004_FA23	['swchan', 'k1chang', 'e8garcia', 's4kumar', 'hrodriguezterry']
6	Group005_FA23	['sfrianeza', 'yuj036', 'emiranda', 'x1shu', 'svallumchetla']
7	Group006_FA23	['suaye', 'ejao', 'sbnorthrop', 'j1tsai', 'eyenikomshian']
8	Group007_FA23	['a1agrawal', 'mbach', 'kggarcia', 'anprasad', 'x7wen']

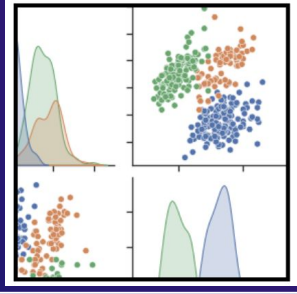
DISCUSSION LAB 3

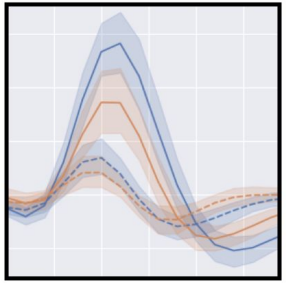
DATA VISUALIZATION AND EXPLORATORY DATA ANALYSIS

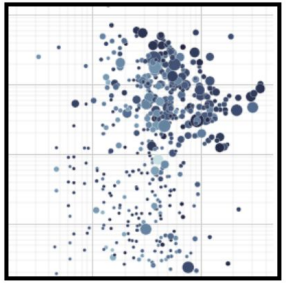
PYPLOT AND SEABORN

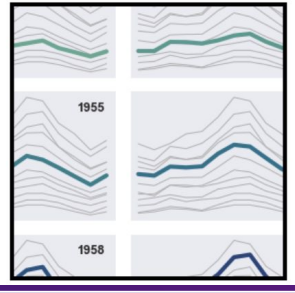
Matplotlib is a library for creating static, animated, and interactive visualizations in Python. Most of the Matplotlib utilities lies under the pyplot submodule, and are usually imported under the plt alias

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Alias : sns









PANDAS SERIES AND DATAFRAMES

Python3

```python
#importing pandas library
import pandas as pd

#Creating a list
author = ['Jitender', 'Purnima', 'Arpit', 'Jyoti']
#Creating a Series by passing list variable to Series() function
auth_series = pd.Series(author)
#Printing Series
print(auth_series)
```

Output:

```
0    Jitender
1     Purnima
2       Arpit
3       Jyoti
dtype: object
```

PANDAS SERIES AND DATAFRAMES

We have created two lists ‘author’ and article’ which have been passed to Series() functions to create two Series.

After creating Series, we have created a dictionary and passed Series objects as values of the dictionary and keys of the dictionary will be served as Columns of the dataframe.

Python3

```python
#Importing Pandas library
import pandas as pd

#Creating two lists
author = ['Jitender', 'Purnima', 'Arpit', 'Jyoti']
article = [210, 211, 114, 178]
#Creating two Series by passing lists
auth_series = pd.Series(author)
article_series = pd.Series(article)
#Creating a dictionary by passing Series objects as values
frame = { 'Author': auth_series, 'Article': article_series }
#Creating DataFrame by passing Dictionary
result = pd.DataFrame(frame)
#Printing elements of Dataframe
print(result)
```

Output:

```
     Author  Article
0  Jitender      210
1   Purnima      211
2     Arpit      114
3     Jyoti      178
```

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

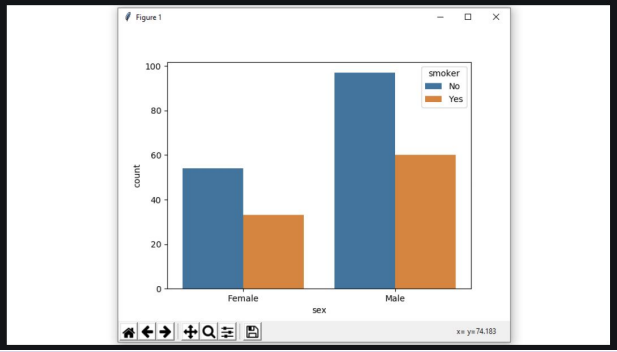
sns.countplot(x, y, hue, data=df);

***The first “plot_cheated” is looking for COUNTS, not proportions!

Python3

```python
# 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 :



Ref: https://www.geeksforgeeks.org/countplot-using-seaborn-in-python/

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())
```

Ref: https://www.geeksforgeeks.org/countplot-using-seaborn-in-python/

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);
```

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

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

Ref: https://www.geeksforgeeks.org/countplot-using-seaborn-in-python/

THANKS!

Questions on campuswire or office hours
Office hours: -