Project Title: Infobeermatics

Team Members: Ali, Emmanuel, Graham, Shakeel

Project Description/Outline: Looking at different beer relationships

Research Questions to answer:

1. ABV vs Rating
2. avb vs rating grouped by country
3. type vs rating
4. type vs rating grouped by country
5. brewery country of origin vs rating
6. Calories?
7. Type of brewery vs rating? (e.g. micro)

Break Down of Tasks:

1. Pull of raw data OpenBrewery API - Ali
2. Pull of raw data RateBeer API - Graham
3. Merge of data sets (pandas)- Emmanuel
4. ABV vs Rating and ABV vs rating grouped by country - Shakeel
5. Type vs rating and type vs rating grouped by country - Graham
6. Brewery country of origin vs rating - Emmanuel
7. Clean data (pandas)
8. Analysis of data (matplotlib)
9. Gmaps visualization of brewery locations (heatmap) - Ali
10. Bonus: User input to predict favorite beer/suggest beer to try

Data sets to be used: apis: OpenBrewery DB, RateBeer

Plan B tasks:

Brewery Ratings, google places, - Graham

Brewery Data, open brewery db, - Ali

Merge brewery data - Emmanuel

Brewery type vs. rating - Shakeel

State vs. rating (what state have the best ratings) - Graham

Gather weather data - Emmanuel

Merge weather data to brewery data - Shakeel

Brewery rating by max temp - Ali

Plan B goals:

Heat Map of Breweries - done

Regression correlation between Weather and Brewery rating

Breakdown of all states - done

Graphs:

Heat Map by density

Rating by State

Rating by type

Rating by City

Scatter of Rating by Temp

Data sets to be used: apis: OpenBrewery DB, Google Maps, Google Places

# Group DataFrame by Brewery Type

group\_by\_type = dataset\_clean2.groupby('brewery\_type')

means = group\_by\_type['rating'].mean().tolist()

rounded\_means = [round(means[x],1) for x in range(len(means))]

#Plot Average Rating By Brewery Type

plt.figure(figsize = (10,5))

plt.bar(np.arange(0,len(group\_by\_type),1),height = group\_by\_type['rating'].mean(), color = 'lightblue', align = 'center', tick\_label = group\_by\_type['rating'].mean().index, width = 0.8,)

plt.grid()

plt.title("Different Brewery Types")

plt.xlabel("Brewery Type")

plt.ylabel("Average Brewery Rating")

plt.xticks(rotation = 45)

plt.yticks(np.arange(3.5,5.1,0.1))

plt.ylim(3.5,5.1)

[plt.annotate(rounded\_means[x],[x - 0.1,group\_by\_type['rating'].mean()[x]+0.01], color = 'black', fontsize=15) for x in range(len(group\_by\_type['rating'].mean()))]

plt.savefig("Average Brewery Rating vs Brewery Type")

#Importing Stats

from scipy.stats import linregress

#Making Line of Best Fit for City Temp

(slope, intercept, \_, \_, \_) = linregress(dataset\_clean2['temp'], dataset\_clean2['rating'])

fit = slope \* dataset\_clean2['temp'] + intercept

#Making plot

fig, ax = plt.subplots()

fig.suptitle("Brewery Rating vs. City Temperature °F", fontsize=12)

ax.set\_xlabel("Temperature °F")

ax.set\_ylabel("Rating")

corr\_coef = np.corrcoef(dataset\_clean2['temp'], dataset\_clean2['rating'])[0, 1]

ax.plot(dataset\_clean2['temp'], dataset\_clean2['rating'], color = 'red', linewidth=0, marker='o')

ax.plot(dataset\_clean2['temp'], fit, 'b--')

plt.grid()

plt.show()

print('The slope of the line of best fit is ' + str(slope))

print(f'The correlation coefficient is {corr\_coef}')

#Pull state temp data into df

state\_weather = pd.read\_csv('weatherData.csv')

state\_weather.head()

#Making Line of Best Fit for State Temp

(slope2, intercept2, \_, \_, \_) = linregress(dataset\_clean2['Average Yearly Temp'],

dataset\_clean2['rating'])

fit2 = slope2 \* dataset\_clean2['Average Yearly Temp'] + intercept2

#Making plot

fig2, ax2 = plt.subplots()

fig2.suptitle("Brewery Rating vs. State Temperature °F", fontsize=12)

ax2.set\_xlabel("Temperature °F")

ax2.set\_ylabel("Rating")

corr\_coef2 = np.corrcoef(dataset\_clean2['Average Yearly Temp'], dataset\_clean2['rating'])[0, 1]

ax2.plot(dataset\_clean2['Average Yearly Temp'], dataset\_clean2['rating'],

color = 'red', linewidth=0, marker='o')

ax2.plot(dataset\_clean2['temp'], fit, 'b--')

plt.grid()

plt.show()

print('The slope of the line of best fit is ' + str(slope2))

print(f'The correlation coefficient is {corr\_coef2}')