STAT 207 Homework 2 Solutions [50 points]

Data Frame Manipulations and Analyses

Due: Friday, January 27 by noon (11:59 am) CST*

*Late Submissions: accepted until Thursday, February 2 at 11:59 pm

---

1. Imports [4 points]

First, we'll import three Python packages that we've used so far and will need for this assignment. Those packages are pandas, matplotlib.pyplot and seaborn.

In the cell below, type the commands and run them in order to import the packages into our workspace.

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

2. Read in the Data [2 points]

For most of this analysis, we'll work with a data set about the world's billionaires from 2021 (https://www.forbes.com/real-time-billionaires/#49c85f6e3d78). This data is contained in the "Billionaire.csv" file located in the same folder.

Read the file into a pandas data frame. You can give it any name, including "df".

In [2]:
```
df = pd.read_csv('Billionaire.csv')
```

3. Understanding the Billionaires Data [6 points]

a) Use the `.head()` function to display the first several lines of the data frame.

In [3]:
```
df.head()
```

Out[3]:

	Name	NetWorth	Country	Source	Rank	Age	Industry
0	Jeff Bezos	$177 B	United States	Amazon	1	57.0	Technology
1	Elon Musk	$151 B	United States	Tesla, SpaceX	2	49.0	Automotive
2	Bernard Arnault & family	$150 B	France	LVMH	3	72.0	Fashion & Retail
3	Bill Gates	$124 B	United States	Microsoft	4	65.0	Technology
4	Mark Zuckerberg	$97 B	United States	Facebook	5	36.0	Technology

b) Use the `.shape` attribute to determine how many observations (rows) and columns (variables) there are.

In [4]:

```
df.shape
```

Out[4]:
```
(2755, 7)
```

c) What is an observational unit for the Billionaire data? That is, what does each row in this data contain information about.

Each row in this data contains information about a billionaire.

d) For each variable in the Billionaire data, what is its type?

Hint: I am asking about the conceptual variable type, not the programming variable type. There is no code needed to answer this question.

- Name: categorical
- Net Worth: quantitative (or numerical)
- Country: categorical
- Source: categorical
- Rank: quantitative
- Age: quantitative
- Industry: categorical

4. Residence of Billionaires [6 points]

The data contains several variables describing each of the billionaires. The third column, 'Country', lists where the billionaire lived in 2021.

a) Determine how many billionaires there are from each country, using Python commands.

In [5]:
```
df['Country'].value_counts()
```

Out[5]:
```
United States           724
China                   626
India                   140
Germany                 136
Russia                  118
                       ...
Venezuela                 1
Eswatini (Swaziland)      1
Algeria                   1
Liechtenstein             1
Nepal                     1
Name: Country, Length: 70, dtype: int64
```

b) What proportion of billionaires lived in the United States in 2021? In China?

Be sure to type your answer in the Markdown cell below.

In [6]:
```
df['Country'].value_counts(normalize=True)
```

Out[6]:
```
United States           0.262795
China                   0.227223
India                   0.050817
Germany                 0.049365
Russia                  0.042831
                            ...
Venezuela               0.000363
Eswatini (Swaziland)    0.000363
```

```
Algeria                0.000363
Liechtenstein          0.000363
Nepal                  0.000363
Name: Country, Length: 70, dtype: float64
```

26.28% of the billionaires lived in the United States. 22.72% lived in China.

5. Industry Source of Income of Billionaires [12 points]

The seventh column of the dataset, 'Industry', provides the industry that the billionaire derived their income from.

a) Determine the **counts** of billionaires that derived their income from each industry, using Python commands.

In [7]:
```
industry_series = df['Industry'].value_counts()
industry_series
```

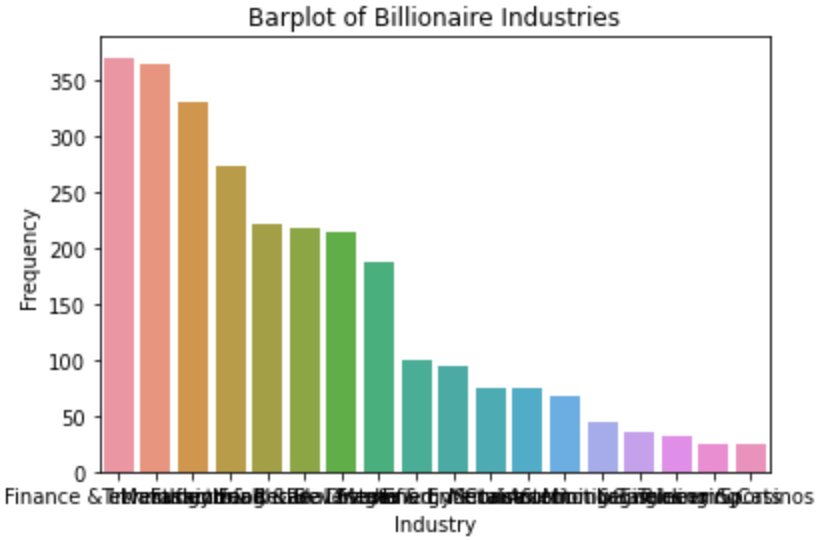
Out[7]:
```
Finance & Investments         371
Technology                    365
Manufacturing                 331
Fashion & Retail              273
Healthcare                    221
Food & Beverage               219
Real Estate                   215
Diversified                   188
Energy                        100
Media & Entertainment          95
Service                        75
Metals & Mining                74
Automotive                     68
Construction & Engineering     44
Logistics                      35
Telecom                        32
Gambling & Casinos             25
Sports                         24
Name: Industry, dtype: int64
```

b) Make a barplot that shows the counts how many billionaires there are that get their income from each type of industry.

Be sure to include:

- a title
- an appropriate label for your x-axis
- an appropriate label for your y-axis.

In [8]:
```
sns.barplot(x=industry_series.index, y=industry_series)
plt.title('Barplot of Billionaire Industries')
plt.xlabel('Industry')
plt.ylabel('Frequency')
plt.show()
```



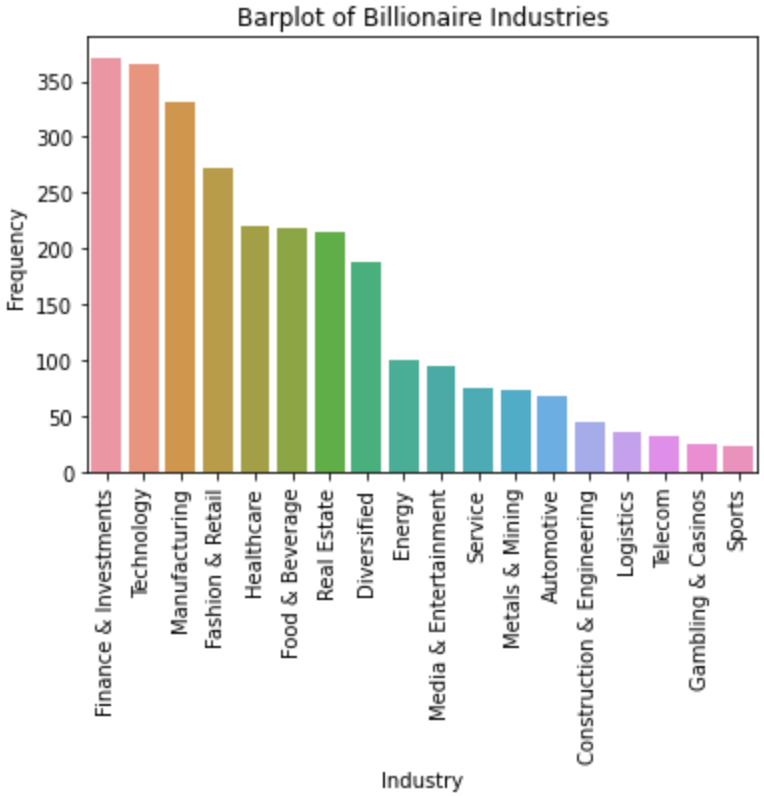
c) Most likely you were unable to to see the labels on the x-axis in your plot above. Run the same code that you have in 5b below, but add the following line of code just above the **plt.show()** line.

```
plt.xticks(rotation = 90)
```

This will rotate your x-axis labels by 90 degrees.

In [9]:

```
sns.barplot(x=industry_series.index, y=industry_series)
plt.title('Barplot of Billionaire Industries')
plt.xlabel('Industry')
plt.ylabel('Frequency')
plt.xticks(rotation = 90)
plt.show()
```



d) Of the billionaires that derived their income from the Technology industry, what **proportion** lived in the United States in 2021? In China? Compare these numbers to what you found in **4b** above.

Use the code cell below to calculate your answer. Be sure to type the relevant numbers in the Markdown cell below that.

In [10]:
```
pd.crosstab(df['Industry'], df['Country'], normalize='index')
```

Out[10]:

Country	Algeria	Argentina	Australia	Austria	Belgium	Brazil	Canada	Chile	China
Industry									
Automotive	0.000000	0.000000	0.014706	0.029412	0.000000	0.000000	0.000000	0.000000	0.250000
Construction & Engineering	0.000000	0.000000	0.000000	0.022727	0.000000	0.000000	0.022727	0.000000	0.068182
Diversified	0.000000	0.005319	0.005319	0.005319	0.000000	0.037234	0.015957	0.010638	0.132979
Energy	0.000000	0.020000	0.000000	0.000000	0.000000	0.020000	0.020000	0.000000	0.210000
Fashion & Retail	0.000000	0.000000	0.018315	0.007326	0.000000	0.054945	0.029304	0.003663	0.120879
Finance & Investments	0.000000	0.000000	0.018868	0.000000	0.002695	0.029650	0.029650	0.010782	0.026954
Food & Beverage	0.004566	0.000000	0.009132	0.004566	0.000000	0.036530	0.022831	0.000000	0.228311
Gambling & Casinos	0.000000	0.000000	0.040000	0.040000	0.000000	0.000000	0.040000	0.000000	0.000000
Healthcare	0.000000	0.004525	0.000000	0.000000	0.000000	0.040724	0.022624	0.000000	0.357466
Logistics	0.000000	0.000000	0.028571	0.028571	0.000000	0.000000	0.000000	0.000000	0.257143
Manufacturing	0.000000	0.000000	0.012085	0.006042	0.003021	0.018127	0.018127	0.000000	0.456193
Media & Entertainment	0.000000	0.000000	0.021053	0.000000	0.000000	0.031579	0.031579	0.000000	0.157895
Metals & Mining	0.000000	0.000000	0.108108	0.000000	0.000000	0.000000	0.000000	0.027027	0.216216
Real Estate	0.000000	0.000000	0.037209	0.004651	0.004651	0.000000	0.032558	0.000000	0.255814
Service	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.013333	0.000000	0.360000
Sports	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Technology	0.000000	0.002740	0.008219	0.000000	0.000000	0.010959	0.027397	0.000000	0.304110
Telecom	0.000000	0.000000	0.031250	0.000000	0.000000	0.000000	0.031250	0.000000	0.125000

18 rows × 70 columns

In [11]:
```
df_tech = df[df['Industry'] == 'Technology']
df_tech['Country'].value_counts(normalize=True)
```

Out[11]:
```
United States     0.383562
China             0.304110
Germany           0.041096
South Korea       0.032877
India             0.032877
Japan             0.032877
Taiwan            0.030137
Canada            0.027397
Israel            0.013699
```

```
Brazil            0.010959
United Kingdom    0.010959
Hong Kong         0.010959
Russia            0.010959
Australia         0.008219
Sweden            0.008219
Netherlands       0.008219
Singapore         0.005479
Switzerland       0.005479
Ireland           0.005479
Czechia           0.005479
France            0.002740
Romania           0.002740
Argentina         0.002740
Iceland           0.002740
Name: Country, dtype: float64
```

Either of the above two approaches are possible. In the first example, we used `pd.crosstab` to create a table that counts the number of billionaires that lived in each country and derived their income from each industry. Using the parameter `normalize='index'` calculated the proportion of billionaires from each industry that lived in a given Country. By looking at the Technology row at the bottom of the table, I can identify that 30.4% lived in China and 38.35% lived in the United States.

Alternatively, I could filter my data to contain only those billionaires who derived their income from Technology. Then, I can calculate the proportion of the Technology billionaires that lived in each country in 2021. I can confirm my that 38.35% lived in the United States, and 30.4% lived in China.

By comparing to 4b, I see that both of these proportions are larger than for all billionaires. This indicates that technology billionaires are more likely to live in the US or China, compared to the billionaires from other industries.

6. Billionaires in Sweden [6 points]

We'd like to learn more about the billionaires that lived in Sweden in 2021.

a) First, create a new version of the data that only contains the billionaires that lived in Sweden. Be sure to give this data frame a different name, so you can still access the original data. How many billionaires lived in Sweden?

In [12]:
```
df_sweden = df[df['Country'] == 'Sweden']
df_sweden.shape
```

Out[12]:
```
(41, 7)
```

41 billionaires lived in Sweden

b) Then, how many industries did the Swedish billionaires derive their income from? That is, what is the number of distinct industries that these billionaires used? Compare this to the number of industries that appeared in the full data.

In [13]:
```
df_sweden['Industry'].value_counts()
```

Out[13]:
```
Finance & Investments    10
Diversified               9
Fashion & Retail          5
Real Estate               4
Food & Beverage           3
```

```
Technology                      3
Media & Entertainment           2
Healthcare                      1
Service                         1
Energy                          1
Manufacturing                   1
Construction & Engineering      1
Name: Industry, dtype: int64
```

I can see that there are 12 distinct industries that the Swedish billionaires derived their income from. There were 18 total industries for all billionaires, so that indicates that there are 6 industries that do not have any billionaires living in Sweden.

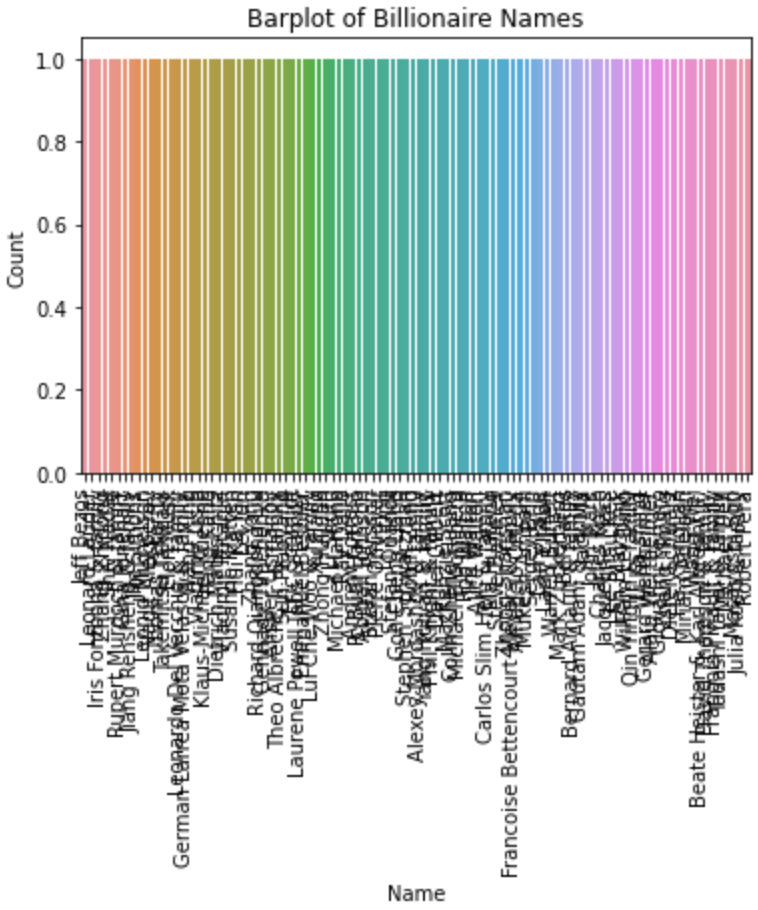
7. Top 100 Barplot [5 points]

The first column in the billionaires data contains the 'Name' of each billionaire. Use Python to create a barplot of the first 100 values of this variable.

Does this barplot contain helpful information?

In [14]:

```
df_100 = df.iloc[:100,:]
name_series = df_100['Name'].value_counts()
sns.barplot(x=name_series.index, y=name_series)
plt.title('Barplot of Billionaire Names')
plt.xlabel('Name')
plt.ylabel('Count')
plt.xticks(rotation = 90)
plt.show()
```



This barplot does not contain helpful information. In essence, it shows that there is no name that is shared by the top 100 billionaires. It's challenging to read any full name, and we can't read almost any of the

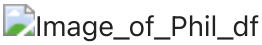
names. If we wanted to share this information more clearly, we could simply provide a list of the top 100 billionaires.

One reason that this graph is not especially helpful is that the variable we are summarizing corresponds to the identifier for each observational unit. These are in essence row identifiers, so we don't have any information to summarize across the variable.

8. Creating a Data Frame [9 points]

On Groundhog Day (February 2), a groundhog in Pennsylvania named Punxsutawney Phil is observed while leaving his burrow. The legend is that if the groundhog can see his shadow on a clear day, that winter will continue for 6 weeks. If clouds prevent him from seeing his shadow, spring will come early.

We'd like to analyze some data about Phil, but unfortunately we misplaced our original csv file. Luckily, we have a picture of the data frame.



Using the tools that you've learned so far, recreate this data frame in Python below. Be sure to print your recreated data frame below.

In [15]:
```
year = [2012, 2013, 2014, 2015, 2016]
shadow = ['yes', 'no', 'yes', 'yes', 'no']
winter = ['more', 'less', 'more', 'more', 'less']
feb = [27.4, 22.8, 16.1, 30.8, 26.5]
mar = [33.9, 30.3, 31.6, 43.4, 35.9]
phil_dict = {'Year': year, 'Shadow': shadow, 'Feb_Temp': feb, 'Mar_Temp': mar}
df = pd.DataFrame(phil_dict)
df
```

Out[15]:

	Year	Shadow	Feb_Temp	Mar_Temp
0	2012	yes	27.4	33.9
1	2013	no	22.8	30.3
2	2014	yes	16.1	31.6
3	2015	yes	30.8	43.4
4	2016	no	26.5	35.9

Remember to keep all your cells and hit the save icon above periodically to checkpoint (save) your results on your local computer. Once you are satisified with your results restart the kernel and run all (Kernel -> Restart & Run All). **Make sure nothing has changed**. Checkpoint and exit (File -> Save and Checkpoint + File -> Close and Halt). Follow the instructions on the Homework 2 Canvas Assignment to submit your notebook to GitHub.