380s25_hw01

January 30, 2025

1 CSC380 Homework 1 Problem 1: Data Science Stories

Find two news stories where: 1. One shows that using data science contributes positively to the world 2. The other shows that using data science without caution resulted in negative consequences.

For each story explain: 1. What questions were answered/decisions were made using data. 2. What data was used.

1.1 Story 1: Positive, Using Data to find the source of a Cholera Outbreak

doi: 10.1016/B978-0-12-804571-8.00017-2

This is a classic example of using data science for a beneficial outcome. This is not a recent news story but one I have heard of before and wanted to analyze more closely.

John Snow was a researcher investigating a Cholera epidemic in the Soho district of London in 1854. He developed a novel method where he took a map of London and plotted the locations of where the people who had Cholera lived. From there, he determined that the people who lived close to one particular well and drank from that well were getting sick. But people who did not live near the well and thus did not drink from it were not getting sick. This demonstrated that that particular well was the source of the Cholera outbreak. The city removed the pump handle from the well, and the Cholera outbreak ceased.

1. What questions were answered/decisions were made using data?

   What was the source of the cholera epidemic?

   The pump handle was removed because it was believed to have contaminated the water and caused the epidemic

2. What data was used?

   Data on the spatial location of where the people who were sick lived as well as the location of the wells in London.

2 Story 2: Negative, Growth in a Time of Debt

doi: 10.1257/aer.100.2.573

For a modern example of bad statistics leading to negative outcomes, I found a story about a paper titled "Growth in a Time of Debt," by Reinhart and Rogoff published in 2010. This paper was published in a non peer-reviewed journal called *American Economic Review*. This paper suggested

that if a country's external debts exceeded 90% of the country's GDP that would lead to economic collapse. This then lead to country's adopting "pro-austerity" policies. Austerity policies are ones which attempt to reduce government budget deficits through a combination of techniques. Because of this move to austerity, policy makers "left behind" many groups that the government should have been protecting, such as the unemployed.

However this study was founded on bad data and other economists were not able to replicate their results. In fact one study found that the excel sheet that Reinhart and Rogoff used contained multiple coding errors. As well, the data used for the study itself was biased as it ommitted data for Australia and other countries when they had high economic growth and large public debts. Lastly it also appears to be a case of correlation being mixed with causation. Reihart and Rogoff claim that the large public debts lead to slow economic growth, but in reality it could be the opposite, or there is some third variable that links them.

1. What questions were answered/decisions were made using data?

   Because of this paper many countries adopted pro-austerity policies at many levels of national debt. This paper was influential for the United States budget in 2014, referred to as the "Paul Ryan Budget." It influenced economic decisions made by the EU. A British member of Parliament even said "As Rogoff and Reinhart demonstrate convincingly, all financial crises ultimately have their origins in one thing."

2. What data was used?

   Reinhart and Rogoff analyzed historic data from multiple countries on the public debts. From this dataset they compared it with historic data on public debt levels, inflation and growth.

3 CSC380 Homework 1 Problem 2: Data Analysis and Visualization

Overview This homework will familiarize you with the basic steps involved in reading, analyzing, and visualizing data. We will use the Starbucks Nutrition Dataset which itemizes most of the food and drink (12oz) options available at the Starbucks coffee chain. To simplify things we have processed the data for you into a JSON file distributed with the homework (filename: starbucks.json). We will be using the Pandas library to load and manipulate data. I briefly introduced all of the Pandas functionality that will need in class and additional links are provided inline below.

The problem has a maximum score of 70 points.

Installing Pandas To install any python library just type:

!pip3 install "library name"

Or, if you are using Anaconda then type:

!conda install "library name"

The cell below can be used to install Pandas. Or you can do it on the command line.

[1]:
```
# Uncomment and run the line below to install Pandas using pip
#!pip3 install pandas
```

```
# Uncomment and run the line below to install Pandas using Anaconda
#!conda install pandas
```

[2]:
```
import pandas as pd
```

3.1 Problem 1 : Basic Operations and Stats from the dataset

Download the data "starbucks.json" and load it to create a Pandas DataFrame.

What is a python DataFrame ? - https://www.geeksforgeeks.org/python-pandas-dataframe/

Hint : Check out the read_json function - https://www.w3schools.com/python/pandas/pandas_json.asp

[3]:
```
starbucks_df = pd.read_json("starbucks.json")
starbucks_df
```

[3]:

	Beverage_category	Beverage	Beverage_prep	Calories	Total Fat (g)	Trans Fat (g)	Saturated Fat (g)	Sodium (mg)
0	Coffee	Brewed Coffee	Plain	5	0.1	0.0	0.0	0
1	Classic Espresso Drinks	Caff Latte	Nonfat Milk	130	0.3	0.2	0.0	5
2	Classic Espresso Drinks	Caff Latte	2% Milk	190	7.0	3.5	0.2	30
3	Classic Espresso Drinks	Caff Latte	Soymilk	150	5.0	0.5	0.0	0
4	Classic Espresso Drinks	Caff Mocha (Without Whipped Cream)	Nonfat Milk	220	2.5	1.5	0.0	5
..	…	…	…	…				
68	Frappuccino Light Blended Coffee	Java Chip	Nonfat Milk	220				
69	Frappuccino Blended Crme	Strawberries & Crme (Without Whipped Cream)	Nonfat Milk	230				
70	Frappuccino Blended Crme	Strawberries & Crme (Without Whipped Cream)	Whole Milk	260				
71	Frappuccino Blended Crme	Strawberries & Crme (Without Whipped Cream)	Soymilk	240				
72	Frappuccino Blended Crme	Vanilla Bean (Without Whipped Cream)	Nonfat Milk	240				

```
..            …            …                 …          …
68           4.0          3.0               0.0          0
69           0.2          0.1               0.0          0
70           4.0          2.0               0.1         10
71           2.0          0.2               0.0          0
72           0.1          0.1               0.0          5

    Total Carbohydrates (g)  Cholesterol (mg)  Dietary Fibre (g)  Sugars (g)  \
0                        10                 0                  0           0
1                       150                19                  0          18
2                       170                19                  0          17
3                       130                13                  1           8
4                       125                43                  2          34
..                      …                …                   …          …
68                      240                43                  2          39
69                      190                53                  0          52
70                      190                53                  0          52
71                      180                51                  1          49
72                      230                56                  0          55

    Protein (g)  Vitamin A (fDV)  Vitamin C (fDV)  Calcium (fDV)  Iron (fDV)  \
0           1.0            0.00             0.00           0.00        0.00
1          13.0            0.20             0.00           0.40        0.00
2          12.0            0.20             0.02           0.40        0.00
3          10.0            0.15             0.00           0.40        0.15
4          13.0            0.20             0.00           0.35        0.25
..          …              …                …              …          …
68          5.0            0.06             0.00           0.10        0.25
69          4.0            0.08             0.06           0.15        0.04
70          4.0            0.06             0.06           0.15        0.04
71          3.0            0.04             0.06           0.15        0.08
72          5.0            0.08             0.00           0.15        0.00

    Caffeine (mg)
0             330
1             150
2             150
3             150
4             175
..            …
68            105
69              0
70              0
71              0
72              0

[73 rows x 18 columns]
```

Printing the entire dataframe looks cumbersome. How can we look at the first and last **two** rows of a dataframe?

Check out .head() and .tail() - https://www.tutorialspoint.com/python_pandas/python_pandas_basic_functiona

What are the first two and last two rows on the dataframe?

[4]:
```
starbucks_df.head(2)
```

[4]:
```
          Beverage_category       Beverage Beverage_prep  Calories  \
0                    Coffee  Brewed Coffee         Plain         5
1  Classic Espresso Drinks    Caff Latte   Nonfat Milk       130

   Total Fat (g)  Trans Fat (g)  Saturated Fat (g)  Sodium (mg)  \
0            0.1            0.0                0.0            0
1            0.3            0.2                0.0            5

   Total Carbohydrates (g)  Cholesterol (mg)  Dietary Fibre (g)  Sugars (g)  \
0                       10                 0                  0           0
1                      150                19                  0          18

   Protein (g)  Vitamin A (fDV)  Vitamin C (fDV)  Calcium (fDV)  Iron (fDV)  \
0          1.0              0.0              0.0            0.0         0.0
1         13.0              0.2              0.0            0.4         0.0

   Caffeine (mg)
0            330
1            150
```

[5]:
```
starbucks_df.tail(2)
```

[5]:
```
             Beverage_category                                       Beverage  \
71  Frappuccino  Blended Crme  Strawberries & Crme (Without Whipped Cream)
72  Frappuccino  Blended Crme           Vanilla Bean (Without Whipped Cream)

   Beverage_prep  Calories  Total Fat (g)  Trans Fat (g)  Saturated Fat (g)  \
71       Soymilk       240            2.0            0.2                0.0
72   Nonfat Milk       240            0.1            0.1                0.0

    Sodium (mg)  Total Carbohydrates (g)  Cholesterol (mg)  Dietary Fibre (g)  \
71            0                      180                51                  1
72            5                      230                56                  0

    Sugars (g)  Protein (g)  Vitamin A (fDV)  Vitamin C (fDV)  Calcium (fDV)  \
71          49          3.0             0.04             0.06           0.15
72          55          5.0             0.08             0.00           0.15

    Iron (fDV)  Caffeine (mg)
71        0.08              0
```

```
72        0.00                0
```

How can we access just a column of a dataset in pandas? https://cmdlinetips.com/2020/04/3-ways-to-select-one-or-more-columns-with-pandas/.

It is okay if while printing you only see first and last few element and dots in between; this is Python's way of summarizing the output.

Print the column 'Beverage_prep'

[6]:
```
starbucks_df["Beverage_prep"]
```

[6]:
```
0            Plain
1      Nonfat Milk
2          2% Milk
3          Soymilk
4      Nonfat Milk
           …
68     Nonfat Milk
69     Nonfat Milk
70      Whole Milk
71         Soymilk
72     Nonfat Milk
Name: Beverage_prep, Length: 73, dtype: object
```

One goal of data science is to use data in order to answer questions. This is done in an automated way, without us having to manually go through the data. Let's try answering some simple questions about Starbucks' menu items.

3.1.1 a. On an average, how much caffeine does a starbucks drink have?

Hint: Checkout the math functions of a pandas dataframe.

https://erikrood.com/Python_References/pandas_column_average_median_final.html

[7]:
```
print(starbucks_df["Caffeine (mg)"].mean())
```

```
95.75342465753425
```

3.1.2 b. What is the *typical* (median) amount of caffeine in a starbucks drink?

[8]:
```
print(starbucks_df["Caffeine (mg)"].median())
```

```
100.0
```

3.1.3 c. What is the maximum amount of caffeine you can find at starbucks in its drinks?

[9]:
```
print(starbucks_df["Caffeine (mg)"].max())
```

```
330
```

3.1.4 d. What is the least amount of caffeine you can find at starbucks in its drinks?

[10]:
```
print(starbucks_df["Caffeine (mg)"].min())
```

```
0
```

3.2 Problem 2 : Pie Chart

Let's explore the dataset we have a bit more further

3.2.1 a. What are the different type of Drinks (ie Beverage Category)that Starbucks has? How much of each?

Hint - Checkout pandas value_counts() function.

[11]:
```
#print the different beverage category and how much of each here

starbucks_df["Beverage_category"].value_counts()
```

[11]:
```
Beverage_category
Classic Espresso Drinks              14
Tazo  Tea Drinks                     13
Frappuccino  Blended Coffee          12
Signature Espresso Drinks            10
Smoothies                             9
Shaken Iced Beverages                 6
Frappuccino  Light Blended Coffee     4
Frappuccino  Blended Crme            4
Coffee                                1
Name: count, dtype: int64
```

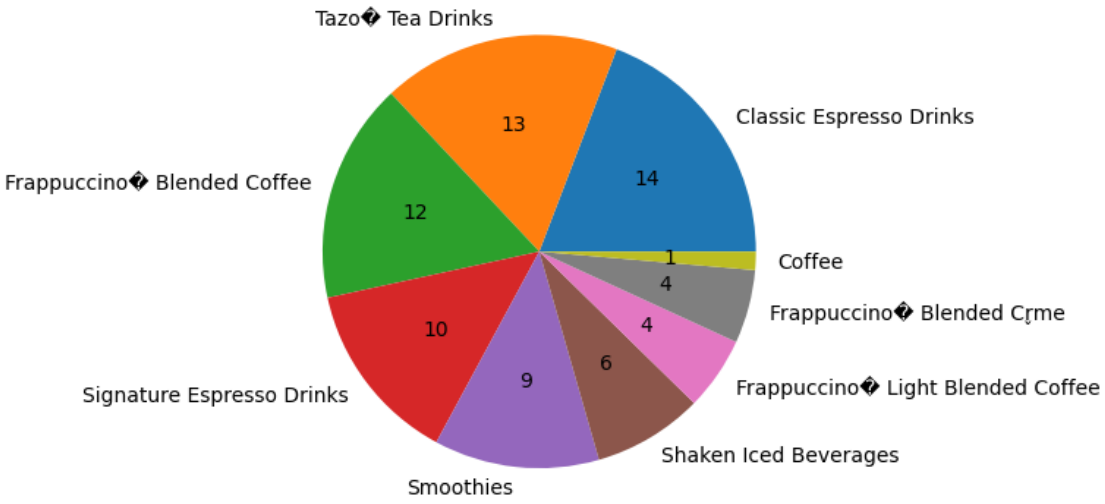
Let's make these more appealing. Plot these as a pie chart

[12]:
```
import matplotlib.pyplot as plt

beverage_category_counts = starbucks_df["Beverage_category"].value_counts()
labels = beverage_category_counts.keys()
sizes = beverage_category_counts

fig, ax = plt.subplots()
ax.pie(sizes, labels=labels, autopct= lambda x: '{:.0f}'.format(x*sizes.sum()/
 ↪100))

plt.show()
```



3.3 Problem 3 : Bar Chart

Suppose you have a very calorie conscious friend. But they really like to get the drinks at Starbucks. As a budding Data Scientist, you want to help them out.

3.3.1 a. What is the drink with the least amount of calories at Starbucks?

Hint : Check this out ==> https://www.interviewqs.com/ddi-code-snippets/rows-cols-python

[13]:
```
#you can print the entire row or just the name

starbucks_df["Beverage"][starbucks_df["Calories"] == starbucks_df["Calories"].
 ↪min()]
```

[13]:
```
25    Tazo  Tea
Name: Beverage, dtype: object
```

But they are quickly bored of this drink. I mean, it's only natural.

So, let's recommend them a beverage category instead.

First let's find on an average how much calories do each beverage category have?

Hint - Checkout groupby function. The first example in this page is what we are trying to do. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.groupby.html

[14]:
```
starbucks_df[["Beverage_category", "Calories"]].groupby("Beverage_category").
 ↪mean()
```

[14]:
```
                                Calories
Beverage_category
```

```
Classic Espresso Drinks            162.500000
Coffee                               5.000000
Frappuccino  Blended Coffee        272.500000
Frappuccino  Blended Crme          242.500000
Frappuccino  Light Blended Coffee  160.000000
Shaken Iced Beverages              106.666667
Signature Espresso Drinks          281.000000
Smoothies                          282.222222
Tazo  Tea Drinks                   203.076923
```

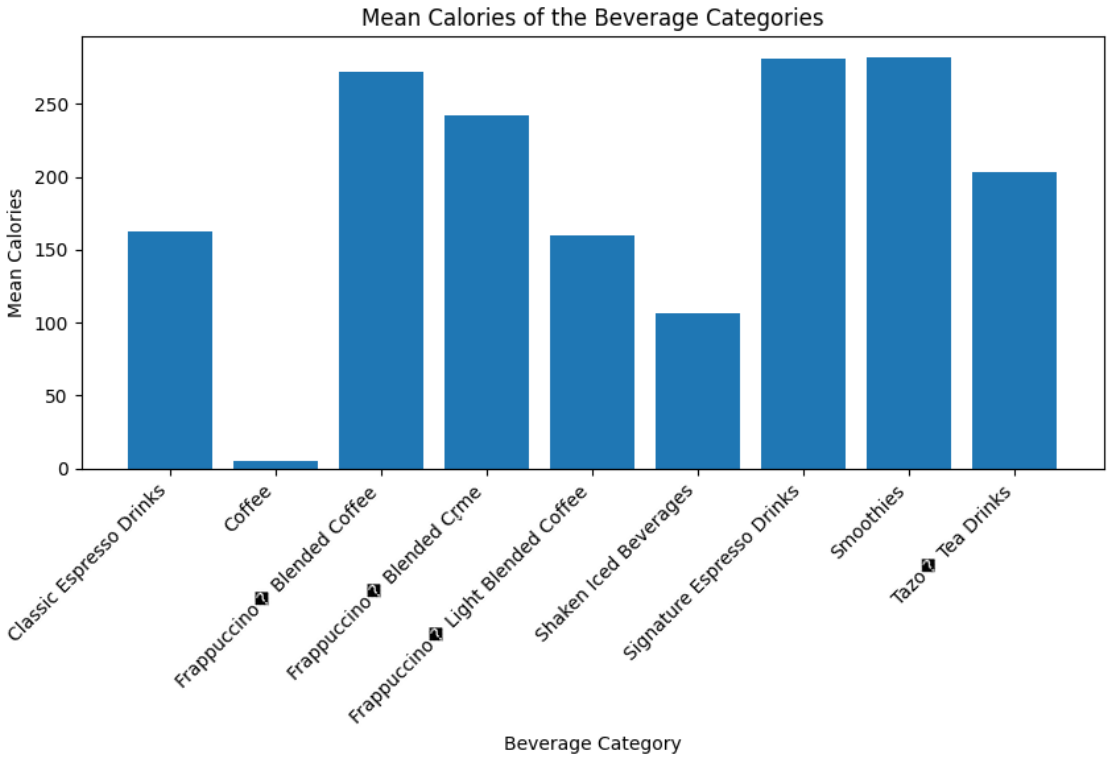
3.3.2 b. Plot a bar Graph

Let's make this visually appealing by plotting a bar graph, where the height of the bar plot is average amount of calories.

Hint : Check this out -> https://benalexkeen.com/bar-charts-in-matplotlib/

[15]:
```
data = starbucks_df[["Beverage_category", "Calories"]].
 ↪groupby("Beverage_category").mean()
categories = data.index.tolist()
means = data["Calories"]

fig, ax = plt.subplots(figsize=(10, 4.25))
ax.bar(categories, means)
ax.set_xlabel("Beverage Category")
ax.set_ylabel("Mean Calories")
ax.set_title("Mean Calories of the Beverage Categories")
plt.xticks(rotation=45, ha="right")
plt.show()
```



3.3.3 By looking at the graph, which beverage category has the least average calories?

[16]:
```
print('Coffee')
```

```
Coffee
```

Let's keep looking

3.3.4 By looking at the graph, which beverage category has the second least average calories?

[17]:
```
print('Shaken Iced Beverages')
```

```
Shaken Iced Beverages
```

This gives us some idea of how many calories to expect in each beverage category. But we know from our previous classes that taking just the mean is not a good representation of how the values are spread. In this case, while the average is useful, we need to know how it is spread across various drinks within a beverage category.

3.3.5 What is the standard deviation of calories within each beverage categories?

Standard deviation is a concept we haven't yet covered in class (as of 1/22). But its idea is simple: similar to the range of the data we saw, it measures the "spread" of the data.

Hint: try replacing the aggregate function after groupby() from mean() to std().

[18]:
```
starbucks_df[["Beverage_category", "Calories"]].groupby("Beverage_category").
↪std()
```

[18]:

Beverage_category	Calories
Classic Espresso Drinks	71.649521
Coffee	NaN
Frappuccino Blended Coffee	36.711405
Frappuccino Blended Crme	12.583057
Frappuccino Light Blended Coffee	42.426407
Shaken Iced Beverages	18.618987
Signature Espresso Drinks	71.561939
Smoothies	13.017083
Tazo Tea Drinks	87.405627

If you get a nan for Coffee in the above cell, just add .fillna(0) at the end. To read more about fillna(0) - https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html

Else skip the below cell

[19]:
```
starbucks_df[["Beverage_category", "Calories"]].groupby("Beverage_category").
↪std().fillna(0)
```

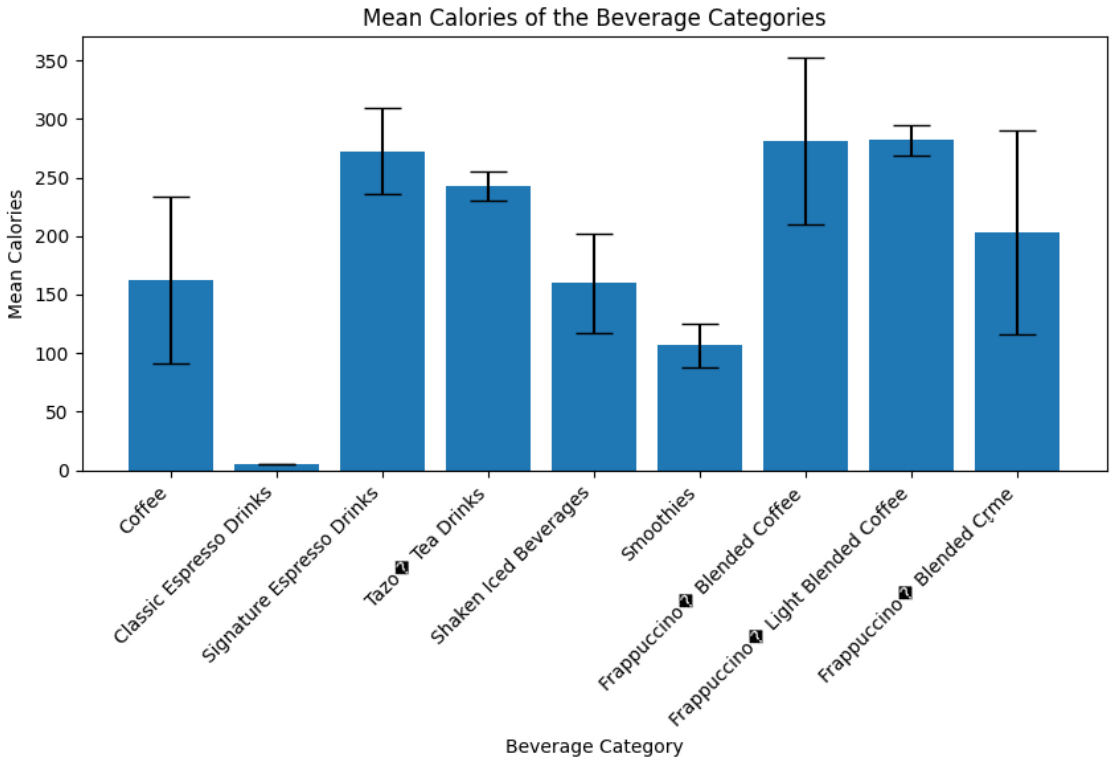
[19]:

Beverage_category	Calories
Classic Espresso Drinks	71.649521
Coffee	0.000000
Frappuccino Blended Coffee	36.711405
Frappuccino Blended Crme	12.583057
Frappuccino Light Blended Coffee	42.426407
Shaken Iced Beverages	18.618987
Signature Espresso Drinks	71.561939
Smoothies	13.017083
Tazo Tea Drinks	87.405627

Now Let's incorporate this info into the bar chart as well. We want a bar chart where there is 1 bar for each beverage category, the height is average calories, and error bars representing +/- sample standard deviation. Hint: go back to https://benalexkeen.com/bar-charts-in-matplotlib/

[20]:
```
data = starbucks_df[["Beverage_category", "Calories"]].
↪groupby("Beverage_category").mean()
categories = starbucks_df["Beverage_category"].unique()
means = data["Calories"]
std = starbucks_df[["Beverage_category", "Calories"]].
↪groupby("Beverage_category").std().fillna(0)["Calories"]
```

```
fig, ax = plt.subplots(figsize=(10, 4.25))
ax.bar(categories, means, yerr = std, capsize=10)
ax.set_xlabel("Beverage Category")
ax.set_ylabel("Mean Calories")
ax.set_title("Mean Calories of the Beverage Categories")
plt.xticks(rotation=45, ha="right")
plt.show()
plt.show()
```



Look how easy it is to understand that many numbers when visualised well!

Awesome work so far!!

3.4 Problem 4 : Scatter plot

Now another friend of yours, who absolutely loves Caffeine came to you for a recommendation.They want to know what are the top drinks with the most Caffeine in Starbucks. They would like to know how much sugar each of them may have too, since they would like to reduce that.They don't like numbers much,so we want to present this to them in a attractive way. Let's start by sorting the DataFrame based on Caffeine

Hint: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sort_values.html

[21]:
```
starbucks_df.sort_values("Caffeine (mg)", ascending=False)
```

[21]:

	Beverage_category	Beverage	\
0	Coffee	Brewed Coffee	
10	Classic Espresso Drinks	Caff Americano	
4	Classic Espresso Drinks	Caff Mocha (Without Whipped Cream)	
6	Classic Espresso Drinks	Caff Mocha (Without Whipped Cream)	
5	Classic Espresso Drinks	Caff Mocha (Without Whipped Cream)	
..	…	…	
52	Smoothies	Strawberry Banana Smoothie	
69	Frappuccino Blended Crme	Strawberries & Crme (Without Whipped Cream)	
70	Frappuccino Blended Crme	Strawberries & Crme (Without Whipped Cream)	
71	Frappuccino Blended Crme	Strawberries & Crme (Without Whipped Cream)	
72	Frappuccino Blended Crme	Vanilla Bean (Without Whipped Cream)	

	Beverage_prep	Calories	Total Fat (g)	Trans Fat (g)	Saturated Fat (g)	\
0	Plain	5	0.1	0.0	0.0	
10	Plain	15	0.0	0.0	0.0	
4	Nonfat Milk	220	2.5	1.5	0.0	
6	Soymilk	230	7.0	2.0	0.0	
5	2% Milk	260	8.0	4.5	0.2	
..	…	…	…	…	…	
52	Soymilk	290	2.0	0.4	0.0	
69	Nonfat Milk	230	0.2	0.1	0.0	
70	Whole Milk	260	4.0	2.0	0.1	
71	Soymilk	240	2.0	0.2	0.0	
72	Nonfat Milk	240	0.1	0.1	0.0	

	Sodium (mg)	Total Carbohydrates (g)	Cholesterol (mg)	Dietary Fibre (g)	\
0	0	10	0	0	
10	0	15	3	0	
4	5	125	43	2	
6	0	105	37	3	
5	25	140	42	2	
..	…	…	…	…	
52	5	120	58	8	
69	0	190	53	0	
70	10	190	53	0	
71	0	180	51	1	
72	5	230	56	0	

	Sugars (g)	Protein (g)	Vitamin A (fDV)	Vitamin C (fDV)	Calcium (fDV)	\
0	0	1.0	0.00	0.00	0.00	
10	0	1.0	0.00	0.00	0.02	
4	34	13.0	0.20	0.00	0.35	
6	26	11.0	0.10	0.00	0.35	
5	34	13.0	0.15	0.02	0.35	
..	…	…	…	…	…	
52	40	16.0	0.02	1.00	0.10	

```
69          52          4.0             0.08            0.06            0.15
70          52          4.0             0.06            0.06            0.15
71          49          3.0             0.04            0.06            0.15
72          55          5.0             0.08            0.00            0.15

    Iron (fDV)  Caffeine (mg)
0         0.00            330
10        0.00            225
4         0.25            175
6         0.40            175
5         0.25            175
..         …              …
52        0.08              0
69        0.04              0
70        0.04              0
71        0.08              0
72        0.00              0

[73 rows x 18 columns]
```

What are the top 10 **drinks** with the most caffeine in them?

Hint : Remember.head() from earlier. Use that.

[22]:
```
top10_Caf_Drink = starbucks_df.sort_values("Caffeine (mg)", ascending=False).
↪head(10)
top10_Caf_Drink
```

[22]:
```
          Beverage_category                                  Beverage  \
0                    Coffee                             Brewed Coffee
10  Classic Espresso Drinks                            Caff Americano
4   Classic Espresso Drinks       Caff Mocha (Without Whipped Cream)
6   Classic Espresso Drinks       Caff Mocha (Without Whipped Cream)
5   Classic Espresso Drinks       Caff Mocha (Without Whipped Cream)
38    Shaken Iced Beverages   Iced Brewed Coffee (With Classic Syrup)
2   Classic Espresso Drinks                               Caff Latte
3   Classic Espresso Drinks                               Caff Latte
1   Classic Espresso Drinks                               Caff Latte
8   Classic Espresso Drinks  Vanilla Latte (Or Other Flavoured Latte)

   Beverage_prep  Calories  Total Fat (g)  Trans Fat (g)  Saturated Fat (g)  \
0          Plain         5            0.1            0.0                0.0
10         Plain        15            0.0            0.0                0.0
4    Nonfat Milk       220            2.5            1.5                0.0
6        Soymilk       230            7.0            2.0                0.0
5        2% Milk       260            8.0            4.5                0.2
38         Plain        90            0.1            0.0                0.0
2        2% Milk       190            7.0            3.5                0.2
```

```
3        Soymilk       150            5.0            0.5                0.0
1     Nonfat Milk      130            0.3            0.2                0.0
8        2% Milk       250            6.0            3.5                0.2

    Sodium (mg)  Total Carbohydrates (g)  Cholesterol (mg)  Dietary Fibre (g)  \
0             0                       10                 0                  0
10            0                       15                 3                  0
4             5                      125                43                  2
6             0                      105                37                  3
5            25                      140                42                  2
38            0                        5                21                  0
2            30                      170                19                  0
3             0                      130                13                  1
1             5                      150                19                  0
8            25                      150                37                  0

    Sugars (g)  Protein (g)  Vitamin A (fDV)  Vitamin C (fDV)  Calcium (fDV)  \
0            0          1.0             0.00             0.00           0.00
10           0          1.0             0.00             0.00           0.02
4           34         13.0             0.20             0.00           0.35
6           26         11.0             0.10             0.00           0.35
5           34         13.0             0.15             0.02           0.35
38          21          0.3             0.00             0.00           0.00
2           17         12.0             0.20             0.02           0.40
3            8         10.0             0.15             0.00           0.40
1           18         13.0             0.20             0.00           0.40
8           35         12.0             0.20             0.02           0.35

    Iron (fDV)  Caffeine (mg)
0         0.00            330
10        0.00            225
4         0.25            175
6         0.40            175
5         0.25            175
38        0.00            165
2         0.00            150
3         0.15            150
1         0.00            150
8         0.00            150
```

We don't really care about the other nutritions at this point. Let's just print what is needed.

[23]:
```
top10_Caf_Drink[['Beverage','Sugars (g)','Caffeine (mg)']]
```

[23]:
```
                                 Beverage  Sugars (g)  Caffeine (mg)
0                           Brewed Coffee           0            330
10                         Caff Americano           0            225
4        Caff Mocha (Without Whipped Cream)          34            175
```

```
6        Caff Mocha (Without Whipped Cream)          26            175
5        Caff Mocha (Without Whipped Cream)          34            175
38   Iced Brewed Coffee (With Classic Syrup)          21            165
2                                Caff Latte          17            150
3                                Caff Latte           8            150
1                                Caff Latte          18            150
8   Vanilla Latte (Or Other Flavoured Latte)          35            150
```

Oops, why does the same drink keep repeating but with different calories and caffeine? Give yourself a minute before reading the next line for the answer.

Yes, they are prepared differently. Let's add that too, since it is relevent information

[24]:
```
top10_Caf_Drink[['Beverage','Beverage_prep','Sugars (g)','Caffeine (mg)']]
```

[24]:
```
                                  Beverage Beverage_prep  Sugars (g)  \
0                              Brewed Coffee         Plain           0
10                            Caff Americano         Plain           0
4        Caff Mocha (Without Whipped Cream)   Nonfat Milk          34
6        Caff Mocha (Without Whipped Cream)       Soymilk          26
5        Caff Mocha (Without Whipped Cream)       2% Milk          34
38   Iced Brewed Coffee (With Classic Syrup)         Plain          21
2                                Caff Latte       2% Milk          17
3                                Caff Latte       Soymilk           8
1                                Caff Latte   Nonfat Milk          18
8   Vanilla Latte (Or Other Flavoured Latte)       2% Milk          35

    Caffeine (mg)
0             330
10            225
4             175
6             175
5             175
38            165
2             150
3             150
1             150
8             150
```

Now that we have the beverages with the prep, sugar and caffeine,we need to show this to our friend. Let's plot them as a need scatter plot. Caffeine on x, Sugars on y.

[25]:
```
x = top10_Caf_Drink["Caffeine (mg)"].tolist()
y = top10_Caf_Drink["Sugars (g)"].tolist()

beverages = top10_Caf_Drink['Beverage'].to_list()
beverage_prep = top10_Caf_Drink['Beverage_prep'].to_list()
labels = [ str(beverages[i]) + ' with' + str(beverage_prep[i]) for i in␣
 ↪range(len(top10_Caf_Drink))]
```

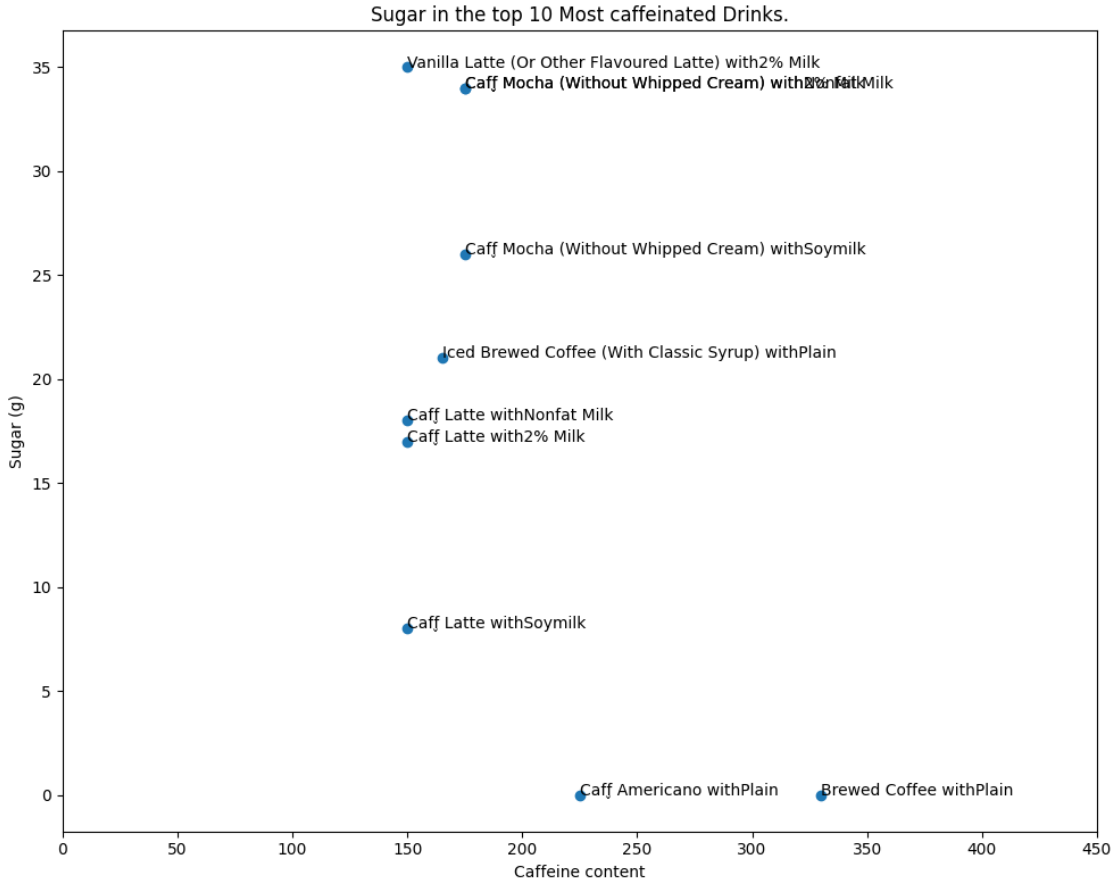
```
#insert your code here to plot the graph here
plt.figure(figsize=(10, 8))
plt.scatter(x, y)

plt.xlabel("Caffeine content")
plt.ylabel("Sugar (g)")
plt.title("Sugar in the top 10 Most caffeinated Drinks.")

axes = plt.gca()
axes.set_xlim([0,450])

for i, txt in enumerate(labels):
    plt.annotate(txt, (x[i], y[i]))

plt.tight_layout()
plt.show()
```



Nice work!!

4 Conclusion

In this assignment, we were able to download a dataset, load it as a pandas dataframe, explore the dataset with basic statistical functions and visulaise many specifc examples to answer relevent queries from the topic.

Congragulations!!