Homework 4: Statistics and Visualization

University of Arizona CSC 380: Principles of Data Science

Homework due at 11:59pm on Oct 15

Deliverables Your must make two submissions: (1) your homework as a SINGLE PDF file by the stated deadline to the gradescope (Homework 4). Include your code and output of the code as texts in the PDF. and (2) your codes in HW04.ipynb file to a separate submission (Homework 4 code). Each subproblem is worth 10 points. More instructions:

- You can hand-write your answers and scan them to make it a PDF. If you use your phone camera, I recommend using TurboScan (smartphone app) or similar ones to avoid uploading a slanted image or showing the background. Make sure you rotate it correctly.
- Watch the video and follow the instruction for the submission: https://youtu.be/KMPoby5g_nE
- **Show all work along with answers to get the full credit**.
- **Paste all your codes and outputs in the PDF report to get full credit.**
- Place your final answer into an 'answer box' that can be easily identified.
- Map the questions with your solutions when submitting. Points will be deducted if not following this.
- There will be no late days. Late homeworks result in zero credit.

Failure to follow the submission instructions will result in a minor penalty in credit.

You need to work individually.

- If you have clarification questions, please feel free to post on Piazza so that it can promote discussion.

Problem 1: Bootstrap Confidence Interval for Pearson Correlation (40pts)

This question is a continuation of HW3, Problem 3. Previously, we have built point estimates of the Pearson correlation ρ and examined their variations; this time, we will construct ρ's confidence interval using the bootstrap method. Lecture slides and Chapter 8 of the textbook (Wasserman) will be helpful if you need a refresher. To approach this question, you can build on your own code for HW3 Problem 3, or the code in HW3's solution guide (posted in D2L).

Recall the problem setup from last time: the data is drawn iid from the following Gaussian distribution: $P(X, Y; \mu, \Sigma) = \mathcal{N}(\mu, \Sigma)$, where the population mean is $\mu = (0, 0)^T$ and the population covariance matrix is

$$\Sigma = \begin{pmatrix} \sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2 \end{pmatrix} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}.$$

The covariance matrix can be written in terms of the correlation coefficient,

$$\rho = \frac{\mathrm{Cov}(X, Y)}{\sigma_X\sigma_Y} = \frac{0.5}{1 \cdot 1} = 0.5.$$

Also recall that our estimator of Pearson correlation is defined as:

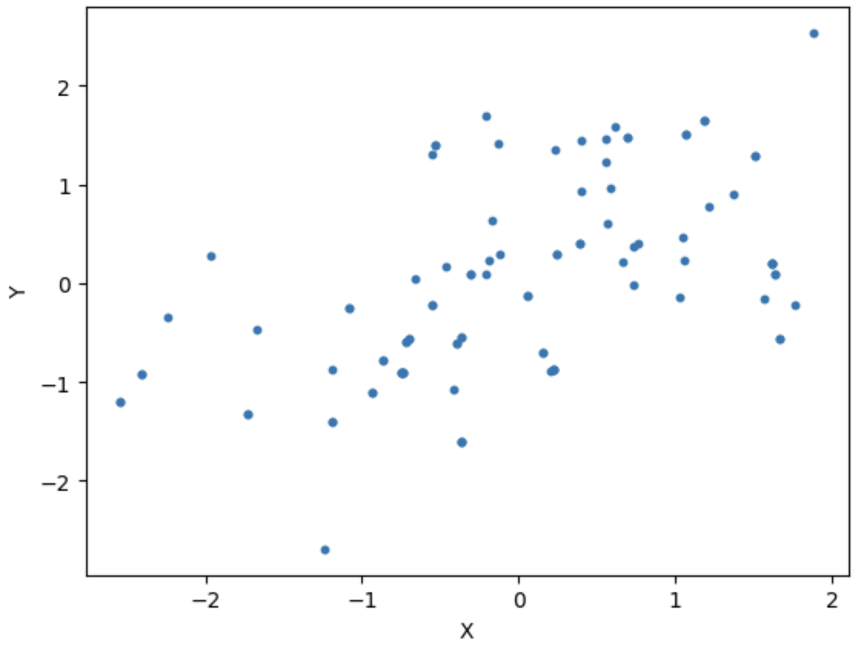
$$\hat{\rho}_N = \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_i (X_i - \bar{X})^2 \sum_j (Y_j - \bar{Y})^2}}$$

Where $\bar{X} = \frac{1}{N}\sum_i X_i$ is the sample mean (and similarly for $\bar{Y}$).

Based on the dataset you generated from your HW3 Problem 3 a), denoted as $S = (Z_1, \ldots, Z_N)$ (here each $Z_i = (X_i, Y_i)$, $N = 100$ and using *seed* $= 0$), answer the following:

a) (5pts) Using **np.random.choice** subsample $M = 100$ points $Z_1^*, \ldots, Z_M^*$ from S **with replacement**. Generate a new estimate from this data from your data $\hat{\rho}(Z_1^*, \ldots, Z_M^*)$ and report.

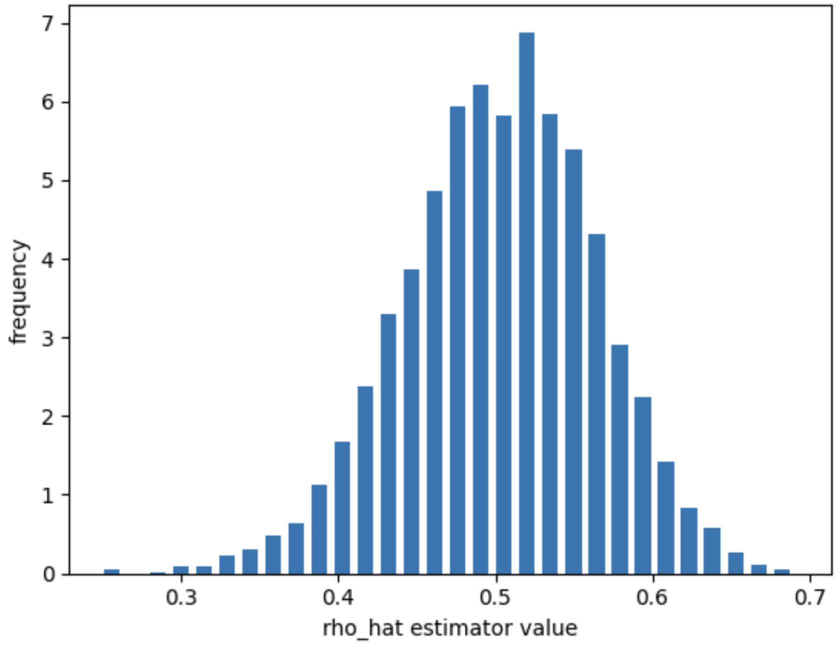
```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> # set seed to 0
>>> np.random.seed(0)
>>> # insert your code for a)
>>> # setting random seed to 0
>>> np.random.seed(0)
>>> # init distribution mean
>>> mean = [0, 0]
>>> # init covariance matrix
>>> cov = [[1, 0.5], [0.5, 1]]
>>> # generate dataset for scatter plot
>>> x, y = np.random.multivariate_normal(mean, cov, size=100).T
>>> # generate subsample of 100 from dataset (with replacement)
>>> sub = np.random.choice(100, size=100, replace=True)
>>> # build x_sub
>>> x_sub = [x[i] for i in sub]
>>> # build y_sub
>>> y_sub = [y[i] for i in sub]
>>> # plot subsample
>>> plt.scatter(x_sub, y_sub, 10)
<matplotlib.collections.PathCollection object at 0x10f1510d0>
>>> # label x axis
>>> plt.xlabel("X")
Text(0.5, 0, 'X')
>>> # label y axis
>>> plt.ylabel("Y")
Text(0, 0.5, 'Y')

>>> # define rho_hat calculation function
>>> def rho_hat(x, y):
...   # compute x_bar
...   x_bar = sum(x) / len(x)
...   # compute y_bar
...   y_bar = sum(y) / len(y)
...   # compute summation terms in numerator
...   num_sum = [(x_val - x_bar)*(y_val - y_bar) for x_val, y_val in zip(x, y)]
...   # compute numerator
...   num = sum(num_sum)
...   # compute summation terms in denominator
...   d0 = [np.square(x_val - x_bar) for x_val in x]
...   d1 = [np.square(y_val - y_bar) for y_val in y]
...   # compute denominator
...   denom = np.sqrt(sum(d0) * sum(d1))
...   # compute and return rho_hat
...   return num / denom
...
>>> # call rho_hat function and write result
>>> print('Plug-in estimator of subsample = ' + str(rho_hat(x_sub, y_sub)))
Plug-in estimator of subsample = 0.5547086113546251
```



b) (10pts) Repeat the above process $B = 5,000$ times to generate bootstrap estimates $\hat{\rho}_{M,1}, \ldots, \hat{\rho}_{M,B}$. Each estimator should be based on a different subsample of M points drawn, with replacement, from the original dataset S. Display a histogram of your bootstrap estimates using **matplotlib.pyplot.hist** with 30 bins. Label your axes.

```
>>> # insert your code for b)

>>> # set seed to 0
>>> np.random.seed(0)

>>> # init array of rho_hats
>>> rho_hats = []

>>> # generate rho_hat for 5000 samples and append to rho_hats
>>> for i in range(5000):
...     # generate subsample of 100 from dataset (with replacement)
...     sub = np.random.choice(100, size=100, replace=True)
...     # build x_sub
...     x_sub = [x[i] for i in sub]
...     # build y_sub
...     y_sub = [y[i] for i in sub]
...     # compute rho_hat for subsample
...     rho_hats.append(rho_hat(x_sub, y_sub))
...
>>> # plot histogram of rho_hats
>>> plt.hist(rho_hats, density=True, bins=30, width = 0.01)
```

```
(array([0.05430911, 0.        , 0.01357728, 0.08146367, 0.08146367,
       0.21723644, 0.29870011, 0.48878199, 0.63813205, 1.12691404,
       1.67000514, 2.37602357, 3.29927845, 3.8695241 , 4.86066537,
       5.93327029, 6.21839312, 5.82465207, 6.88367972, 5.83822935,
       5.39017919, 4.31757426, 2.9055374 , 2.2402508 , 1.41203687,
       0.84179121, 0.58382294, 0.25796827, 0.10861822, 0.05430911]), array([0.2502692
       0.32392172, 0.33865222, 0.35338271, 0.3681132 , 0.3828437 ,
       0.39757419, 0.41230469, 0.42703518, 0.44176567, 0.45649617,
       0.47122666, 0.48595715, 0.50068765, 0.51541814, 0.53014863,
       0.54487913, 0.55960962, 0.57434011, 0.58907061, 0.6038011 ,
       0.6185316 , 0.63326209, 0.64799258, 0.66272308, 0.67745357,
       0.69218406]), <BarContainer object of 30 artists>)
>>> plt.xlabel('rho_hat estimator value')
Text(0.5, 0, 'rho_hat estimator value')
>>> plt.ylabel('frequency')
Text(0, 0.5, 'frequency')
```



c) (5pts) Compare the histogram you obtained from b) with the one you previously got from HW3, Problem 3c). Are they similar? Why?

They are similar in that they both resemble a normal distribution curve that is centered around 0.5 with a slight left skew. They are similar since our new histogram is generated from 5000 subsamples (with replacement) from the original Pearson correlation estimator dataset. Any differences between the histograms is due to the randomness in selecting subsamples.

d) (10pts) Compute and report the 95% confidence interval of ρ based on the bootstrapping method learned in the class. Since what we learned in the class was for the sample mean, you will have to modify it: replace any occurrences of the sample mean with the correlation coefficient. (There are different versions of the bootstrap confidence bounds, but you must use the one from the lecture to receive full credit.) Does the interval contain the true correlation?

```
>>> # insert your code for d)

>>> # sort rho_hats
>>> rho_hats.sort()
>>> # original rho_hat estimator
>>> rh_estimate = 0.5046162424282595
>>> # bootstrap method
>>> bootstrap = [rho_hat - rh_estimate for rho_hat in rho_hats]
>>> # calculate 95 percent confidence interval
>>> q_u = bootstrap[int(np.ceil((1-(0.05/2))*(len(bootstrap) - 1)))]
>>> q_l = bootstrap[int(np.floor((0.05/2)*(len(bootstrap) - 1)))]
>>> # print confidence interval
>>> print("95% confidence interval: [" + str(rh_estimate - q_u) + ", " + str(rh_estima
95% confidence interval: [0.3876405305150534, 0.6294088646368858]
>>> print("This interval does contain the true correlation of 0.5.")
This interval does contain the true correlation of 0.5.
```

Problem 2: Basic data analysis and visualization (60pts)

Please see `380f23_hw04.ipynb`.

HW04_part2

October 16, 2023

1 CSC380 Homework 4 : Data Analysis and Visualization

INDIVIDUAL HOMEWORK The homework is not collaborative anymore. Please respect the academic integrity. **Remember: if you get caught on cheating, you get F.**

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.

Each subproblem is worth 10 pts.

What to turn in: - Please print the notebook containing the answers and results into a pdf file (you can use `File - Print`). Submit the original file as well in the code entry in gradescope. All cells are marked with instructions to insert your code. Please complete all cells as directed. In the worst case where you cannot print it into a pdf file for some reason, you can create a Microsoft word document and then copy paste screenshots showing your code and environment parts by parts. - Should also submit your code separately as usual.

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.

[24]:
```
import pandas as pd
```

1.1 1. Basic Operations and Stats from the DataSet

Download the dataset and read the data as a python 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

[25]:
```
starbucks_df = pd.read_json('starbucks.json')
starbucks_df = pd.DataFrame(starbucks_df)
```

```
starbucks_df
```

[25]:

```
                    Beverage_category  \
0                               Coffee
1             Classic Espresso Drinks
2             Classic Espresso Drinks
3             Classic Espresso Drinks
4             Classic Espresso Drinks
..                                  …
68  Frappuccino  Light Blended Coffee
69         Frappuccino  Blended Crme
70         Frappuccino  Blended Crme
71         Frappuccino  Blended Crme
72         Frappuccino  Blended Crme

                                        Beverage Beverage_prep  Calories  \
0                                  Brewed Coffee         Plain         5
1                                    Caff Latte   Nonfat Milk       130
2                                    Caff Latte       2% Milk       190
3                                    Caff Latte       Soymilk       150
4            Caff Mocha (Without Whipped Cream)   Nonfat Milk       220
..                                            …           …       …
68                                     Java Chip   Nonfat Milk       220
69  Strawberries & Crme (Without Whipped Cream)   Nonfat Milk       230
70  Strawberries & Crme (Without Whipped Cream)    Whole Milk       260
71  Strawberries & Crme (Without Whipped Cream)       Soymilk       240
72          Vanilla Bean (Without Whipped Cream)   Nonfat Milk       240

    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
2             7.0            3.5                0.2           30
3             5.0            0.5                0.0            0
4             2.5            1.5                0.0            5
..            …            …                …          …
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?

[26]:

```
# first two rows of dataframe
starbucks_df.head(2)
# last two rows of dataframe
starbucks_df.tail(2)
```

[26]:

```
             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.

Print the column 'Beverage_prep'

[27]:

```
#insert your code here
starbucks_df[['Beverage_prep']]
```

[27]:

```
   Beverage_prep
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

[73 rows x 1 columns]
```

One beautiful thing about DataScience is that we can answer questions using data we have, but without having to actually manually go through the data. Let's try answering some questions?

1.1.1 a. On an average, how much caffine 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

```
[28]: #insert your code here
      starbucks_df[['Caffeine (mg)']].mean()
      # starbucks_df['Caffeine (mg)'].mean()
```

```
[28]: Caffeine (mg)    95.753425
      dtype: float64
```

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

```
[29]: #insert your code here
      starbucks_df[['Caffeine (mg)']].median()
      # starbucks_df['Caffeine (mg)'].median()
```

```
[29]: Caffeine (mg)    100.0
      dtype: float64
```

1.1.3 b. What is the maximum amount of caffine you can find at starbucks in its drinks?

```
[30]: #insert your code here
      starbucks_df['Caffeine (mg)'].max()
```

```
[30]: 330
```

1.1.4 c. What is the least amount of caffine you can find at starbucks in its drinks?

```
[31]: #insert your code here
      starbucks_df['Caffeine (mg)'].min()
```

```
[31]: 0
```

1.2 2. PieChart

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

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

Hint - Checkout pandas value_counts() function.

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

      #insert your code here
      starbucks_df['Beverage_category'].value_counts()
```

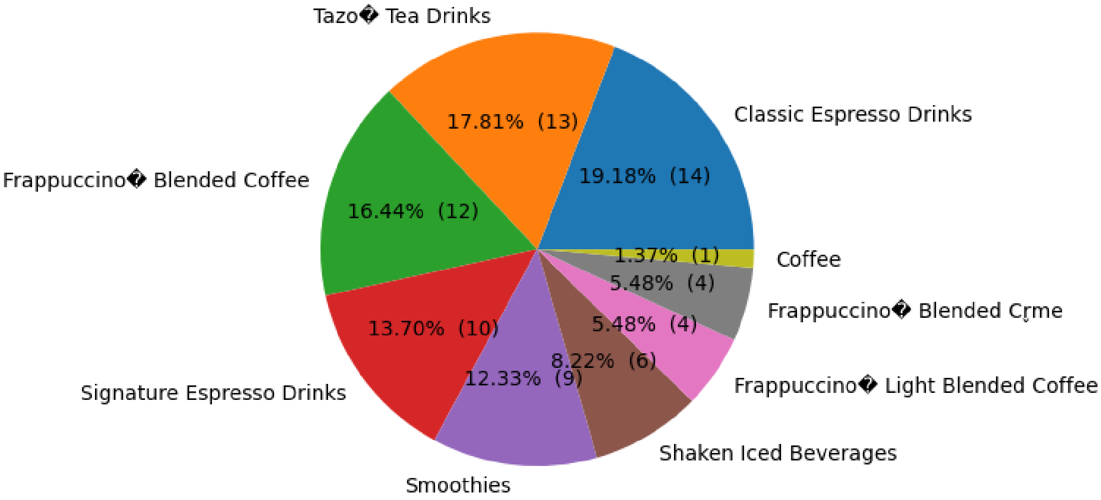
[32]:

```
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

[33]:

```python
import matplotlib.pyplot as plt

beverage_category_counts = starbucks_df['Beverage_category'].value_counts()
labels = beverage_category_counts.index.tolist()
sizes = beverage_category_counts.values

def make_autopct(values):
    def my_autopct(pct):
        total = sum(values)
        val = int(round(pct*total/100.0))
        return '{p:.2f}%  ({v:d})'.format(p=pct,v=val)
    return my_autopct

fig, ax = plt.subplots()
ax.pie(sizes, labels=labels, autopct=make_autopct(sizes))
plt.show()
# beverage_category_counts.plot.pie()
# plt.ylabel('')
```



1.3 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.

1.3.1 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

```
[34]: #insert your code here
      min_cal = starbucks_df['Calories'].min()
      starbucks_df.loc[starbucks_df['Calories'] == min_cal]
```

```
[34]:   Beverage_category   Beverage Beverage_prep  Calories  Total Fat (g)  \
     25  Tazo Tea Drinks  Tazo Tea         Plain         0            0.0

        Trans Fat (g)  Saturated Fat (g)  Sodium (mg)  Total Carbohydrates (g)  \
     25           0.0                0.0            0                        0

        Cholesterol (mg)  Dietary Fibre (g)  Sugars (g)  Protein (g)  \
     25                0                  0           0          0.0

        Vitamin A (fDV)  Vitamin C (fDV)  Calcium (fDV)  Iron (fDV)  Caffeine (mg)
     25             0.0              0.0            0.0         0.0             95
```

#insert your code hereBut 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

[35]:
```
grouped = starbucks_df.groupby(['Beverage_category'])['Calories'].mean().
  ↪sort_values()
```

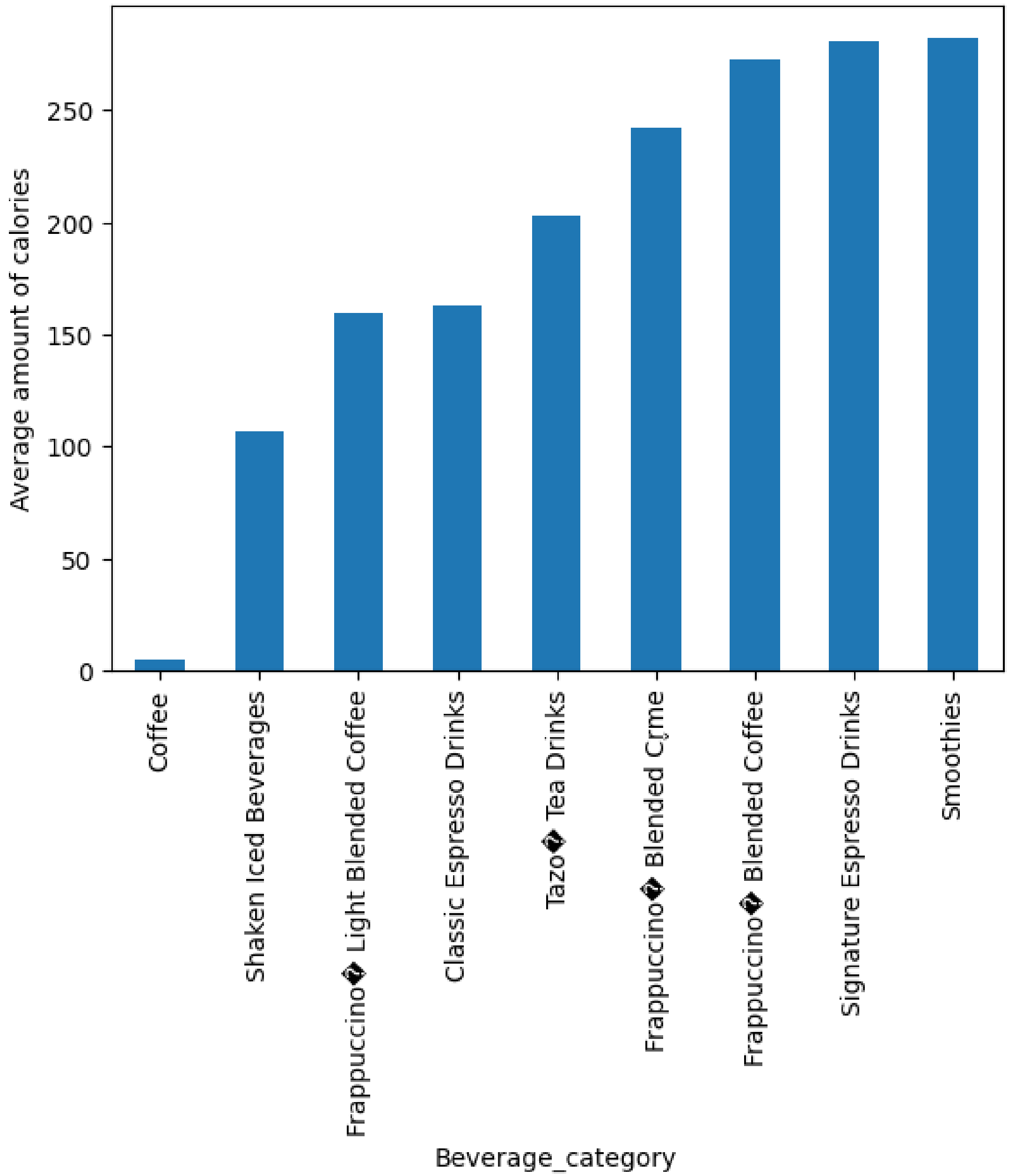
1.3.2 b. Plot a bar Graph

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/

[36]:
```
#insert your code here
grouped.plot.bar()
plt.ylabel('Average amount of calories')
```

[36]:
```
Text(0, 0.5, 'Average amount of calories')
```



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

[37]:
```python
# print the name of the category
print('Coffee')
```

```
Coffee
```

Let’s keep looking

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

[38]:
```
# print the name of the category
print('Shaken Iced Beverages')
```

```
Shaken Iced Beverages
```

This gives us some idea of how much 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.

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

[39]:
```
std = starbucks_df.groupby(['Beverage_category'])['Calories'].std().fillna(0).
  ↪sort_values()
std
```

[39]:
```
Beverage_category
Coffee                                  0.000000
Frappuccino  Blended Crme              12.583057
Smoothies                               13.017083
Shaken Iced Beverages                   18.618987
Frappuccino  Blended Coffee            36.711405
Frappuccino  Light Blended Coffee      42.426407
Signature Espresso Drinks               71.561939
Classic Espresso Drinks                 71.649521
Tazo  Tea Drinks                       87.405627
Name: Calories, dtype: float64
```

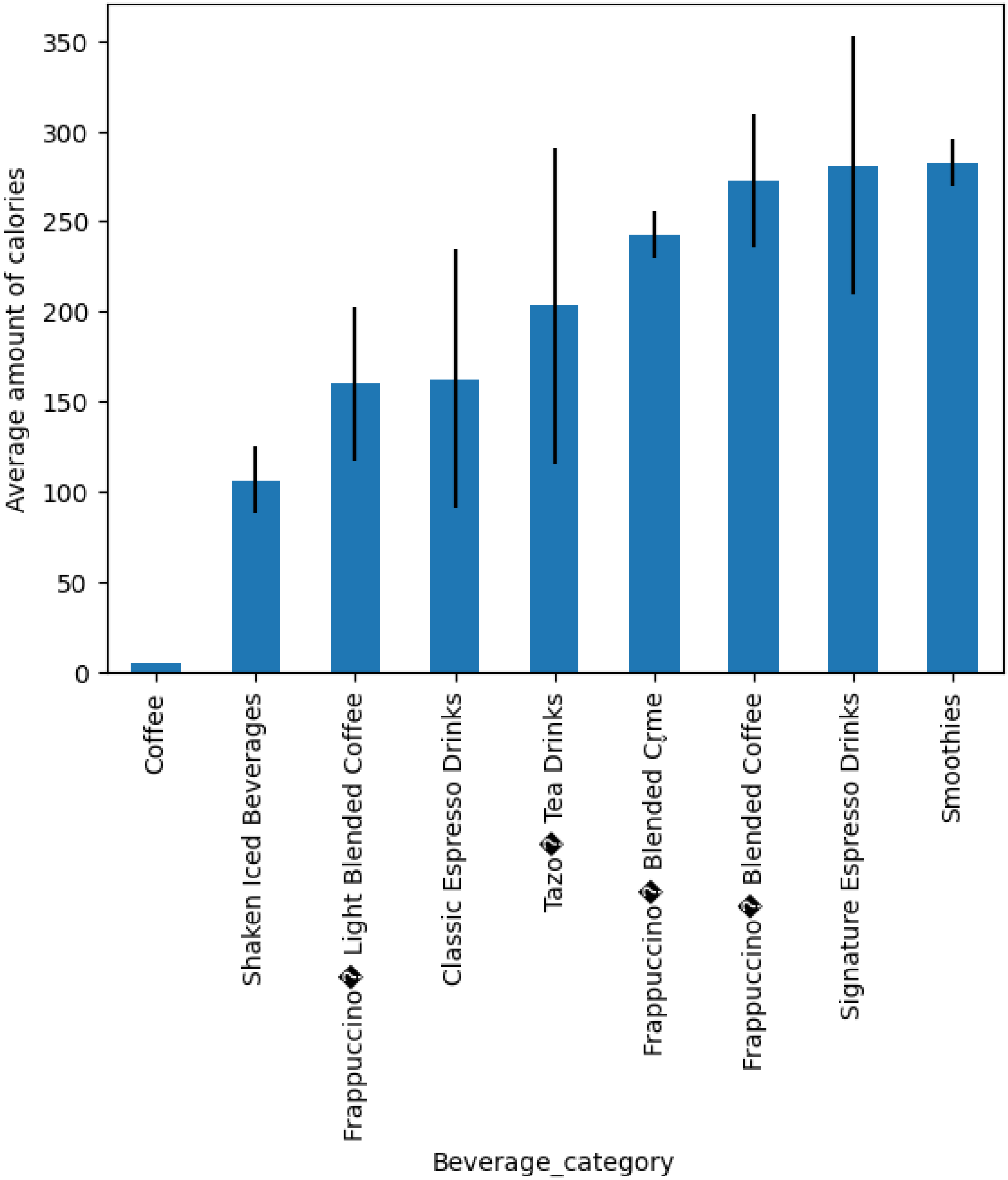
If you get a nan for Coffee in the above cell, just add .fillna(0) at the end. To read more about fillna(0) - https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html

[ ]:
```

```

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 STDEV Hint : go back to https://benalexkeen.com/bar-charts-in-matplotlib/

[40]:
```
#insert your code here
grouped.plot.bar(yerr=std)
plt.ylabel('Average amount of calories')
```

[40]:
```
Text(0, 0.5, 'Average amount of calories')
```



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

Awesome work so far!!

1.4 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 get started.

Let’s sort the dataframe based on Caffeine

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

[41]:
```
starbucks_df.sort_values(by=['Caffeine (mg)'])
```

[41]:
```
             Beverage_category  \
36            Tazo  Tea Drinks
70  Frappuccino  Blended Crme
69  Frappuccino  Blended Crme
52                   Smoothies
51                   Smoothies
..                         …
4      Classic Espresso Drinks
6      Classic Espresso Drinks
5      Classic Espresso Drinks
10     Classic Espresso Drinks
0                       Coffee

                                           Beverage Beverage_prep  Calories  \
36  Tazo  Full-Leaf Red Tea Latte (Vanilla Rooibos)       2% Milk       190
70     Strawberries & Crme (Without Whipped Cream)    Whole Milk       260
69     Strawberries & Crme (Without Whipped Cream)   Nonfat Milk       230
52                       Strawberry Banana Smoothie       Soymilk       290
51                       Strawberry Banana Smoothie       2% Milk       290
..                                              …           …       …
4                Caff Mocha (Without Whipped Cream)   Nonfat Milk       220
6                Caff Mocha (Without Whipped Cream)      Soymilk       230
5                Caff Mocha (Without Whipped Cream)      2% Milk       260
10                                   Caff Americano         Plain        15
0                                     Brewed Coffee         Plain         5

    Total Fat (g)  Trans Fat (g)  Saturated Fat (g)  Sodium (mg)  \
36            4.0            2.0                0.1           15
70            4.0            2.0                0.1           10
69            0.2            0.1                0.0            0
52            2.0            0.4                0.0            5
51            2.0            1.0                0.0            5
..            …            …                …          …
4             2.5            1.5                0.0            5
6             7.0            2.0                0.0            0
5             8.0            4.5                0.2           25
10            0.0            0.0                0.0            0
0             0.1            0.0                0.0            0

    Total Carbohydrates (g)  Cholesterol (mg)  Dietary Fibre (g)  Sugars (g)  \
36                      95              31                0          30
70                     190              53                0          52
69                     190              53                0          52
52                     120              58                8          40
51                     125              58                7          41
..                       …            …                …        …
4                      125              43                2          34
6                      105              37                3          26
5                      140              42                2          34
10                      15               3                0           0
0                       10               0                0           0

    Protein (g)  Vitamin A (fDV)  Vitamin C (fDV)  Calcium (fDV)  Iron (fDV)  \
36          7.0             0.10             0.00           0.25        0.00
70          4.0             0.06             0.06           0.15        0.04
69          4.0             0.08             0.06           0.15        0.04
52         16.0             0.02             1.00           0.10        0.08
51         16.0             0.04             1.00           0.10        0.08
..            …                …              …              …           …
4          13.0             0.20             0.00           0.35        0.25
6          11.0             0.10             0.00           0.35        0.40
5          13.0             0.15             0.02           0.35        0.25
10          1.0             0.00             0.00           0.02        0.00
0           1.0             0.00             0.00           0.00        0.00

    Caffeine (mg)
36              0
70              0
69              0
52              0
51              0
..            …
4             175
6             175
5             175
10            225
0             330

[73 rows x 18 columns]
```

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

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

[42]:

```
top10_Caf_Drink = starbucks_df.sort_values(by=['Caffeine (mg)']).tail(10)
top10_Caf_Drink
```

[42]:

```
            Beverage_category                                       Beverage  \
20  Signature Espresso Drinks  White Chocolate Mocha (Without Whipped Cream)
19  Signature Espresso Drinks  White Chocolate Mocha (Without Whipped Cream)
18  Signature Espresso Drinks  White Chocolate Mocha (Without Whipped Cream)
17  Signature Espresso Drinks                              Caramel Macchiato
38      Shaken Iced Beverages        Iced Brewed Coffee (With Classic Syrup)
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)
10    Classic Espresso Drinks                                Caff Americano
0                      Coffee                                  Brewed Coffee

   Beverage_prep  Calories  Total Fat (g)  Trans Fat (g)  Saturated Fat (g)  \
20       Soymilk       370           10.0            5.0                0.0
19       2% Milk       400           11.0            7.0                0.2
18   Nonfat Milk       350            6.0            4.5                0.0
17       Soymilk       200            5.0            1.0                0.0
38         Plain        90            0.1            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
10         Plain        15            0.0            0.0                0.0
0          Plain         5            0.1            0.0                0.0

    Sodium (mg)  Total Carbohydrates (g)  Cholesterol (mg)  Dietary Fibre (g)  \
20            0                      220                56                  1
19           25                      250                61                  0
18           10                      240                61                  0
17            5                      115                29                  1
38            0                        5                21                  0
4             5                      125                43                  2
6             0                      105                37                  3
5            25                      140                42                  2
10            0                       15                 3                  0
0             0                       10                 0                  0

    Sugars (g)  Protein (g)  Vitamin A (fDV)  Vitamin C (fDV)  Calcium (fDV)  \
20          51         13.0             0.10             0.02           0.45
19          58         15.0             0.15             0.02           0.45
18          58         15.0             0.20             0.02           0.45
17          24          9.0             0.10             0.00           0.35
38          21          0.3             0.00             0.00           0.00
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
10           0          1.0             0.00             0.00           0.02
0            0          1.0             0.00             0.00           0.00
```

```
    Iron (fDV)  Caffeine (mg)
20        0.15            150
19        0.00            150
18        0.02            150
17        0.15            150
38        0.00            165
4         0.25            175
6         0.40            175
5         0.25            175
10        0.00            225
0         0.00            330
```

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

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

```
[43]:
                                         Beverage  Sugars (g)  Caffeine (mg)
20  White Chocolate Mocha (Without Whipped Cream)          51            150
19  White Chocolate Mocha (Without Whipped Cream)          58            150
18  White Chocolate Mocha (Without Whipped Cream)          58            150
17                              Caramel Macchiato          24            150
38        Iced Brewed Coffee (With Classic Syrup)          21            165
4            Caff Mocha (Without Whipped Cream)          34            175
6            Caff Mocha (Without Whipped Cream)          26            175
5            Caff Mocha (Without Whipped Cream)          34            175
10                             Caff Americano           0            225
0                                   Brewed Coffee           0            330
```

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

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

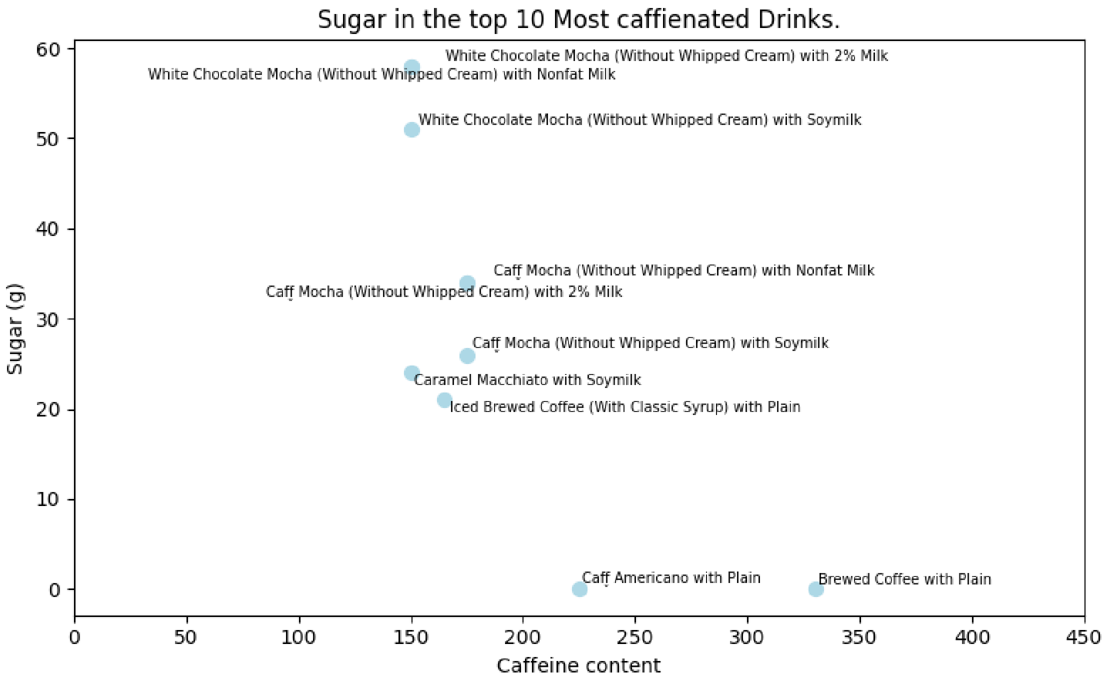
```
[44]:
                                         Beverage Beverage_prep  Sugars (g)  \
20  White Chocolate Mocha (Without Whipped Cream)       Soymilk          51
19  White Chocolate Mocha (Without Whipped Cream)       2% Milk          58
18  White Chocolate Mocha (Without Whipped Cream)   Nonfat Milk          58
17                              Caramel Macchiato       Soymilk          24
38        Iced Brewed Coffee (With Classic Syrup)         Plain          21
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
10                             Caff Americano         Plain           0
0                                   Brewed Coffee         Plain           0

    Caffeine (mg)
```

```
20            150
19            150
18            150
17            150
38            165
4             175
6             175
5             175
10            225
0             330
```

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.

[45]:

```
x = top10_Caf_Drink['Caffeine (mg)'].to_list()
y = top10_Caf_Drink['Sugars (g)'].to_list()

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))]

plt.figure(figsize=(8, 5))
plt.scatter(x, y,s = 50, c='lightblue')

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

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

from adjustText import adjust_text

# for i, txt in enumerate(labels):
#     plt.annotate(txt, (x[i], y[i]), fontsize=8)

texts = [plt.text(x[i], y[i], '%s' %labels[i], ha='center', va='center',␣
 ↪fontsize=7) for i in range(len(labels))]
adjust_text(texts)

plt.tight_layout()
```



Nice work!!

2 Conclusion

In this assignment, we were able to dowmload 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!!