

Name: Kelly Joseph Calvadores

Course and Section: CPE 019 - CPE32S3

Date of Submission: February 7, 2024

Instructor: Engr. Roman M. Richard

Objectives

- Part 1: The Dataset
- Part 2: Scatterplot Graphs and Correlatable Variables
- Part 3: Calculating Correlation with Python
- Part 4: Visualizing

Scenario/Background

Correlation is an important statistical relationship that can indicate whether the variable values are linearly related.

In this lab, you will learn how to use Python to calculate correlation. In Part 1, you will setup the dataset. In Part 2, you will learn how to identify if the variables in a given dataset are correlatable. Finally, in Part 3, you will use Python to calculate the correlation between two sets of variable.

✓ Part 1: The Dataset

You will use a dataset that contains a sample of 40 right-handed Anglo Introductory Psychology students at a large Southwestern university. Subjects took four subtests (Vocabulary, Similarities, Block Design, and Picture Completion) of the Wechsler (1981) Adult Intelligence Scale-Revised. The researchers used Magnetic Resonance Imaging (MRI) to determine the brain size of the subjects. Information about gender and body size (height and weight) are also included. The researchers withheld the weights of two subjects and the height of one subject for reasons of confidentiality. Two simple modifications were applied to the dataset:

1. Replace the question marks used to represent the withheld data points described above by the 'NaN' string. The substitution was done because Pandas does not handle the question marks correctly.
2. Replace all tab characters with commas, converting the dataset into a CSV dataset.

The prepared dataset is saved as brainsize.txt.

✓ *tep 1: Loading the Dataset From a File.

```
import pandas as pd
```



```
BrainFile = 'brainsize.txt'
```

```
BrainFrame = pd.read_csv(BrainFile, '\t')
```

```
<ipython-input-7-729c6881f4c2>:4: FutureWarning: In a future version of pandas all arguments to read_csv will be keyword arguments (by default)
BrainFrame = pd.read_csv(BrainFile, '\t')
```

✓ Step 2: Verifying the dataframe.

```
BrainFrame.head(10)
```

	Gender	FSIQ	VIQ	PIQ	Weight	Height	MRI_Count	
0	Female	133	132	124	118.0	64.5	816932	
1	Male	140	150	124	NaN	72.5	1001121	
2	Male	139	123	150	143.0	73.3	1038437	
3	Male	133	129	128	172.0	68.8	965353	
4	Female	137	132	134	147.0	65.0	951545	
5	Female	99	90	110	146.0	69.0	928799	
6	Female	138	136	131	138.0	64.5	991305	
7	Female	92	90	98	175.0	66.0	854258	
8	Male	89	93	84	134.0	66.3	904858	
9	Male	133	114	147	172.0	68.8	955466	

Part 2: Scatterplot Graphs and Correlatable Variables

✓ Step 1: The pandas describe() method.

```
BrainFrame.describe()
```

	FSIQ	VIQ	PIQ	Weight	Height	MRI_Count
count	40.000000	40.000000	40.000000	38.000000	39.000000	4.000000e+01
mean	113.450000	112.350000	111.02500	151.052632	68.525641	9.087550e+05
std	24.082071	23.616107	22.47105	23.478509	3.994649	7.228205e+04
min	77.000000	71.000000	72.00000	106.000000	62.000000	7.906190e+05
25%	89.750000	90.000000	88.25000	135.250000	66.000000	8.559185e+05
50%	116.500000	113.000000	115.00000	146.500000	68.000000	9.053990e+05
75%	135.500000	129.750000	128.00000	172.000000	70.500000	9.500780e+05
max	144.000000	150.000000	150.00000	192.000000	77.000000	1.079549e+06



✓ Step 2: Scatterplot graphs

a. Load the required modules.

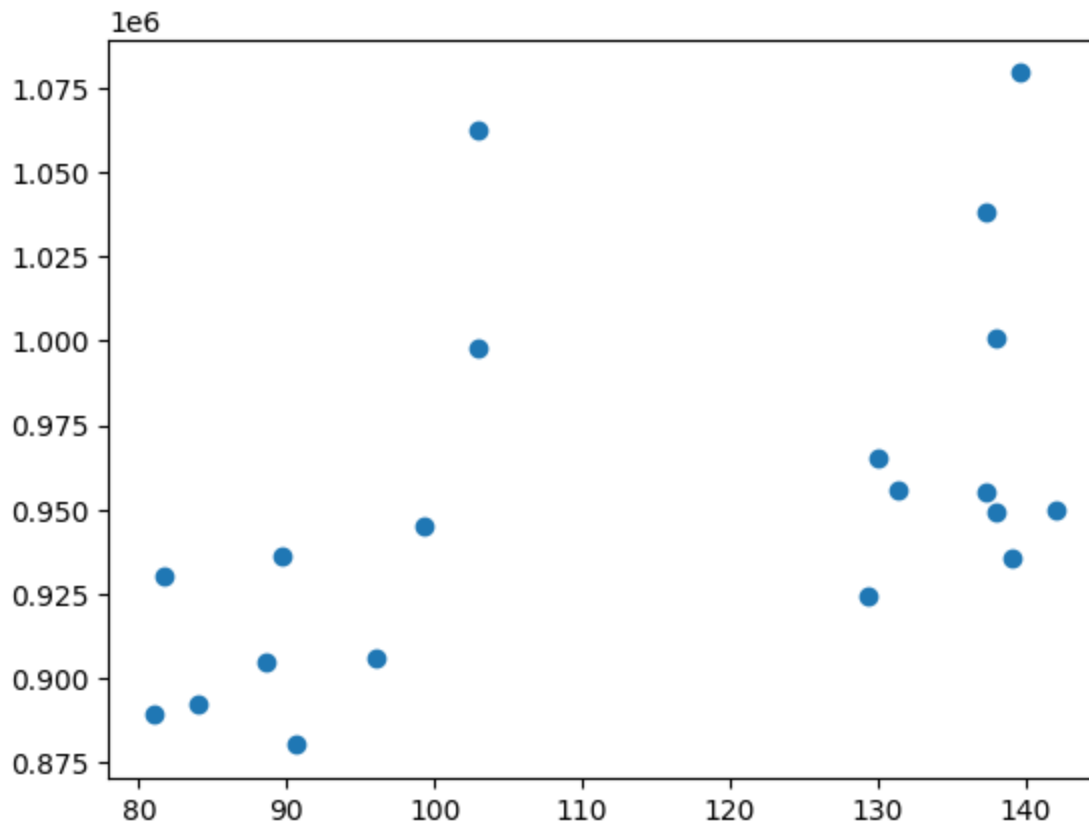
```
import numpy as np
import matplotlib.pyplot as plt
```

b. Separate the data

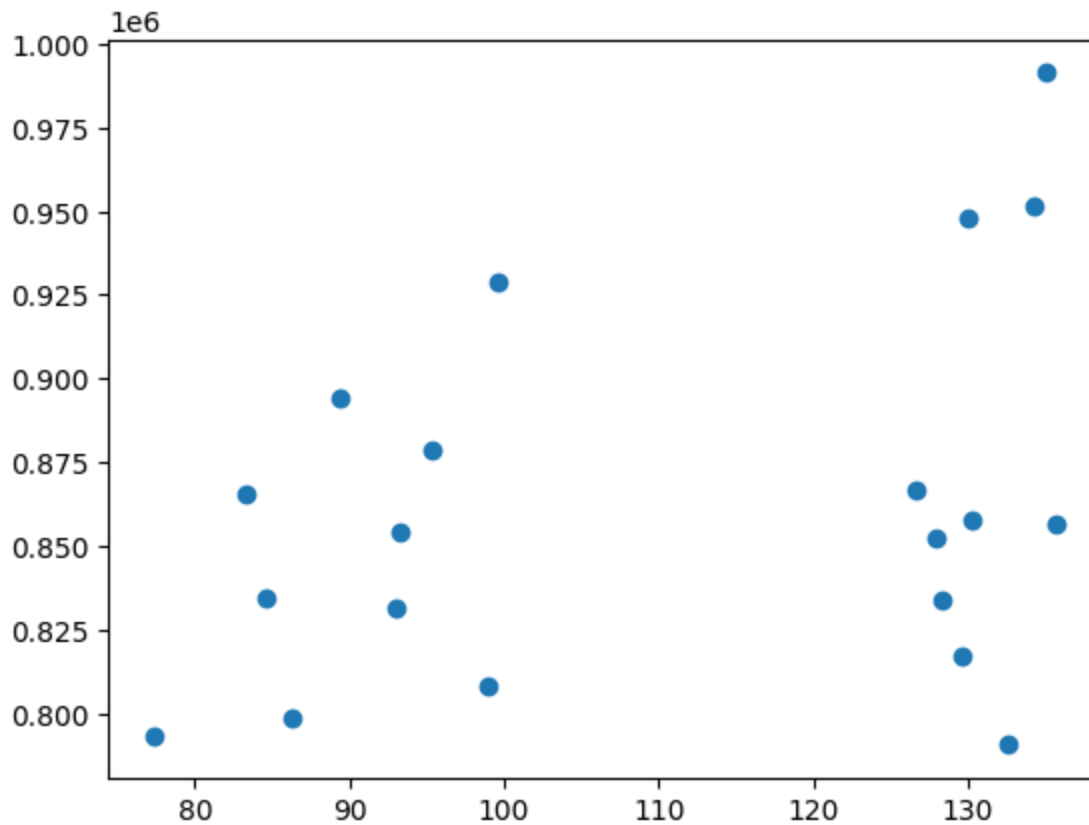
```
MenDF = BrainFrame[(BrainFrame.Gender == 'Male')]
WomenDF = BrainFrame[(BrainFrame.Gender == 'Female')]
```

c. Plot the graphs.

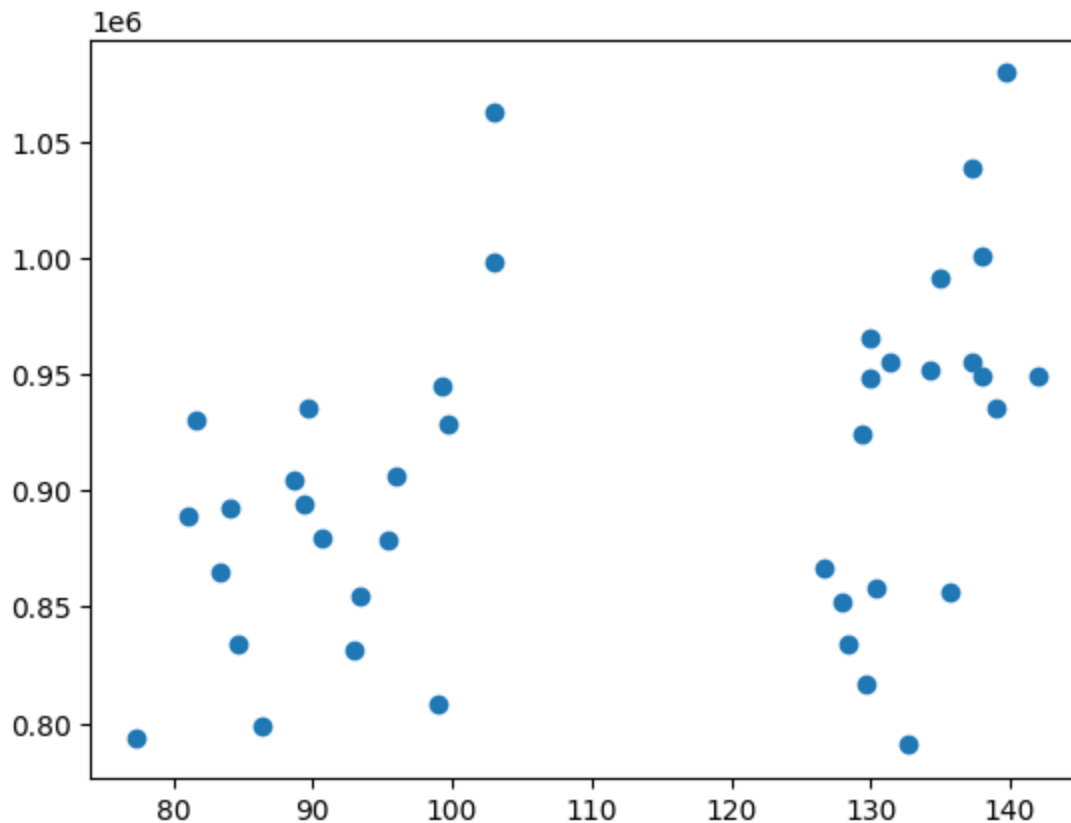
```
#Men Only Filter
MenMeanSmart = MenDF[["PIQ", "FSIQ", "VIQ"]].mean(axis = 1)
plt.scatter(MenMeanSmart, MenDF["MRI_Count"])
plt.show()
#Use %matplotlib inline when using jupyter
```



```
#Women Only Filter
WomenMeanSmart = WomenDF[["PIQ", "FSIQ", "VIQ"]].mean(axis = 1)
plt.scatter(WomenMeanSmart, WomenDF["MRI_Count"])
plt.show()
```



```
#All Gender
AllGenderSmart = BrainFrame[["PIQ", "FSIQ", "VIQ"]].mean(axis = 1)
plt.scatter(AllGenderSmart, BrainFrame["MRI_Count"])
plt.show()
```



✓ Part 3: Calculating Correlation with Python

Step 1: Calculate correlation against brainFrame.

```
BrainFrame.corr(method = 'pearson')
```

```
<ipython-input-22-da64797702a4>:1: FutureWarning: The default value of numeric_only in
BrainFrame.corr(method = 'pearson')
```

	FSIQ	VIQ	PIQ	Weight	Height	MRI_Count	
FSIQ	1.000000	0.946639	0.934125	-0.051483	-0.086002	0.357641	
VIQ	0.946639	1.000000	0.778135	-0.076088	-0.071068	0.337478	
PIQ	0.934125	0.778135	1.000000	0.002512	-0.076723	0.386817	
Weight	-0.051483	-0.076088	0.002512	1.000000	0.699614	0.513378	
Height	-0.086002	-0.071068	-0.076723	0.699614	1.000000	0.601712	
MRI_Count	0.357641	0.337478	0.386817	0.513378	0.601712	1.000000	

Notice at the left-to-right diagonal in the correlation table generated above. Why is the diagonal filled with 1s? Is that a coincidence? Explain.



- It is not coincidence, because variable itself is a perfect correlated.

Still looking at the correlation table above, notice that the values are mirrored; values below the 1 diagonal have a mirrored counterpart above the 1 diagonal. Is that a coincidence? Explain.

- Not coincidence, because either variable such as "VIQ and PIQ" or "PIQ and VIQ" will remain same result due to their same correlation, just in reverse.

```
WomenDF.corr(method = 'pearson')
```

```
<ipython-input-23-d94969706107>:1: FutureWarning: The default value of numeric_only in
WomenDF.corr(method = 'pearson')
```

	FSIQ	VIQ	PIQ	Weight	Height	MRI_Count	
FSIQ	1.000000	0.955717	0.939382	0.038192	-0.059011	0.325697	
VIQ	0.955717	1.000000	0.802652	-0.021889	-0.146453	0.254933	
PIQ	0.939382	0.802652	1.000000	0.113901	-0.001242	0.396157	
Weight	0.038192	-0.021889	0.113901	1.000000	0.552357	0.446271	
Height	-0.059011	-0.146453	-0.001242	0.552357	1.000000	0.174541	
MRI_Count	0.325697	0.254933	0.396157	0.446271	0.174541	1.000000	

```
#The Default corr() method is pearson
MenDF.corr()
```

```
<ipython-input-24-ea4e39d6026f>:1: FutureWarning: The default value of numeric_only in
MenDF.corr()
```

	FSIQ	VIQ	PIQ	Weight	Height	MRI_Count	
FSIQ	1.000000	0.944400	0.930694	-0.278140	-0.356110	0.498369	
VIQ	0.944400	1.000000	0.766021	-0.350453	-0.355588	0.413105	
PIQ	0.930694	0.766021	1.000000	-0.156863	-0.287676	0.568237	
Weight	-0.278140	-0.350453	-0.156863	1.000000	0.406542	-0.076875	
Height	-0.356110	-0.355588	-0.287676	0.406542	1.000000	0.301543	
MRI_Count	0.498369	0.413105	0.568237	-0.076875	0.301543	1.000000	

✓ Part 4: Visualizingt

Step 1: Install Seaborn.

```
!pip install seaborn
```

```
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packages (0.13
Requirement already satisfied: numpy!=1.24.0,>=1.20 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: pandas>=1.2 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in /usr/local/lib/python3.10/dis
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (fro
```



Step 2: Plot the correlation heatmap.

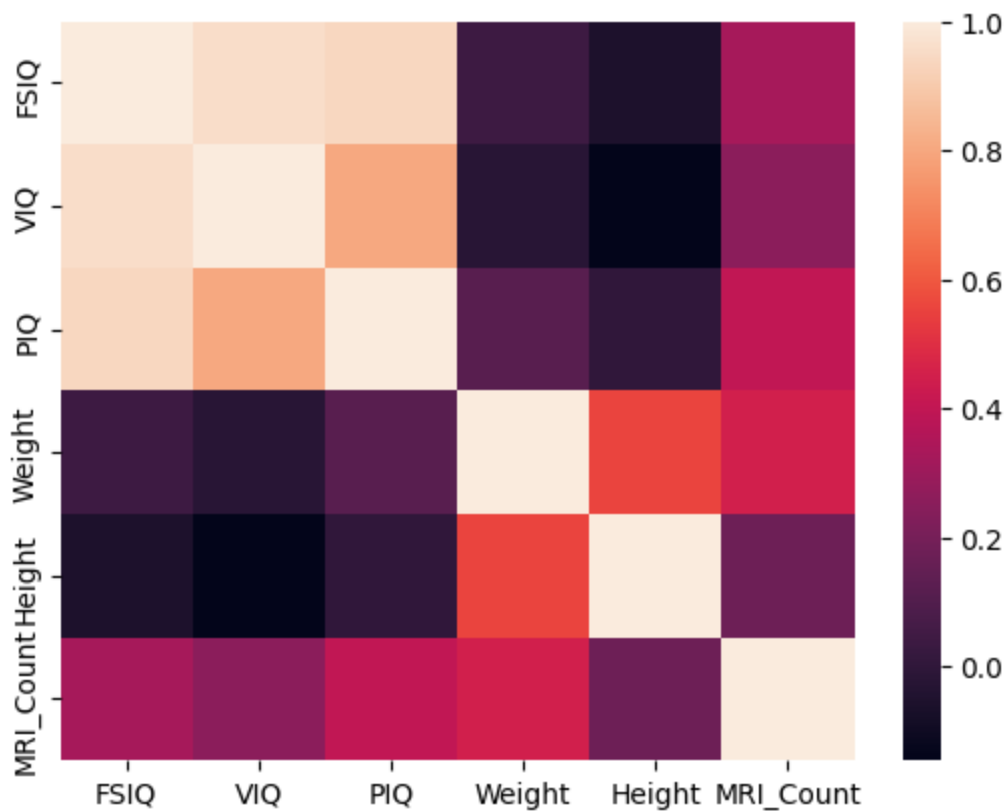
```
import seaborn as sns

WCorr = WomenDF.corr()
sns.heatmap(WCorr)
```



```
<ipython-input-26-d957e9e548a5>:3: FutureWarning: The default value of numeric_only in  
WCorr = WomenDF.corr()
```

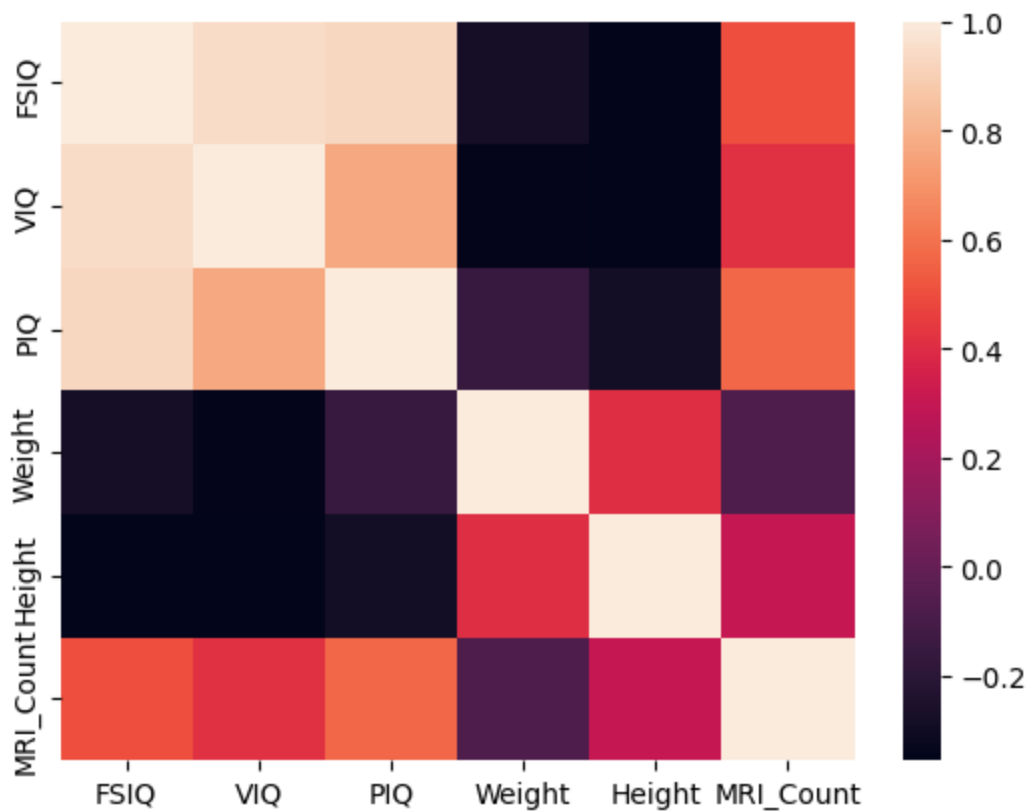
```
<Axes: >
```



```
MCorr = MenDF.corr()  
sns.heatmap(MCorr)
```

```
<ipython-input-27-4f9ef728bd03>:1: FutureWarning: The default value of numeric_only in  
MCorr = MenDF.corr()
```

```
<Axes: >
```



```
BFCorr = BrainFrame.corr()  
sns.heatmap(BFCorr)
```

```
<ipython-input-28-18d6cb08a70f>:1: FutureWarning: The default value of numeric_only in
  BFCorr = BrainFrame.corr()
<Axes: >
```



Many variable pairs present correlation close to zero. What does that mean?

- It means that those pairs of variable that close to zero are weak to no correlated to the brain size, meaning that they are no irrelevant when it comes to brain size.



Why separate the genders?

- Separating the gender due to circumstances of chances, if the brain size is related when it comes to gender, and also easy to organize the data that has been gather.



What variables have stronger correlation with brain size (MRI_Count)? Is that expected? Explain.

- The variables that has a stronger correlation with the brain size are FSIQ, VIQ, and PIQ. As my observation in the data that the three variable has the most correlate to the brain size, their value is almost 1 or closer to 1 which giving a result.

✓ Supplementary Activity