Name: Danica Marie L. Beato

Course and Section: CPE 019 - CPE32S3

Date of Submission: 02/07/2024

Instructor: Engr. Roman Richard

Objectives

Part 1: The Dataset

Part 2: Scatterplot Graphs and Correlatable Variables

Part 3: Calculating Correlation with Python

Part 4: Visualizing

Required Resources

1 PC with Internet access

Raspberry Pi version 2 or higher

Python libraries: pandas, numpy, matplotlib, seaborn

Datafiles: brainsize.txt

Part 1: The Dataset

```
# Loading and creation of the dataset
import pandas as pd
brainFile = './brainsize.txt'
brainFrame = pd.read_csv(brainFile, '\t')
```

```
<ipython-input-1-30e01f2295ac>:3: FutureWarning: In a future version of pandas all ar
  brainFrame = pd.read_csv(brainFile, '\t')
```

```
# Displays first five entries inside the dataframe.
brainFrame.head()
```

	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

Part 2: Scatterplot Graphs and Correlatable Variables

Part 2: Scatterplot Graphs and Correlatable Variables

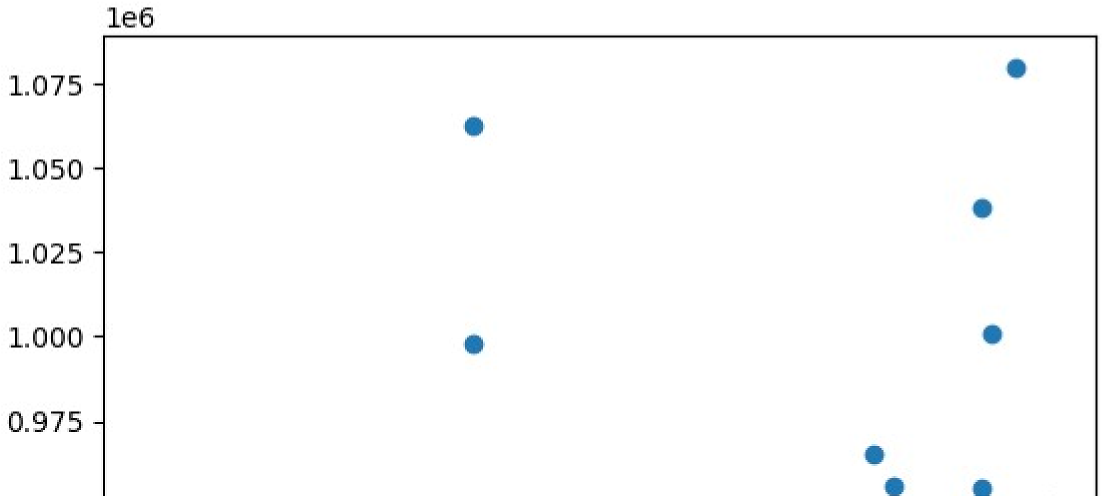
```
# Statistically describes the values in the dataframe
brainFrame.describe()
```

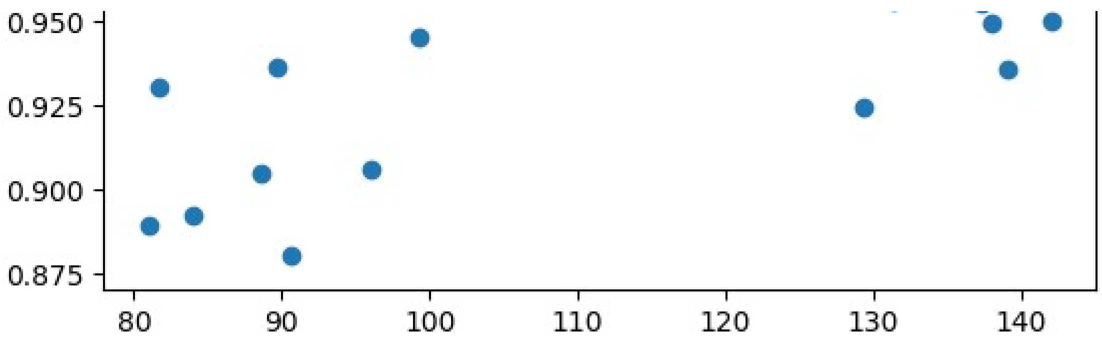
	FSIQ	VIQ	PIQ	Weight	Height	MRI_Count
count	40.000000	40.000000	40.00000	38.000000	39.000000	4.000000e+01
mean	113.450000	112.350000	111.02500	151.052632	68.525641	9.087550e+05
std	24.082071	23.616107	22.47105	23.478509	3.994649	7.228205e+04
min	77.000000	71.000000	72.00000	106.000000	62.000000	7.906190e+05
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

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

menDf = brainFrame[(brainFrame.Gender == 'Male')]
womenDf = brainFrame[(brainFrame.Gender == 'Female')]
```

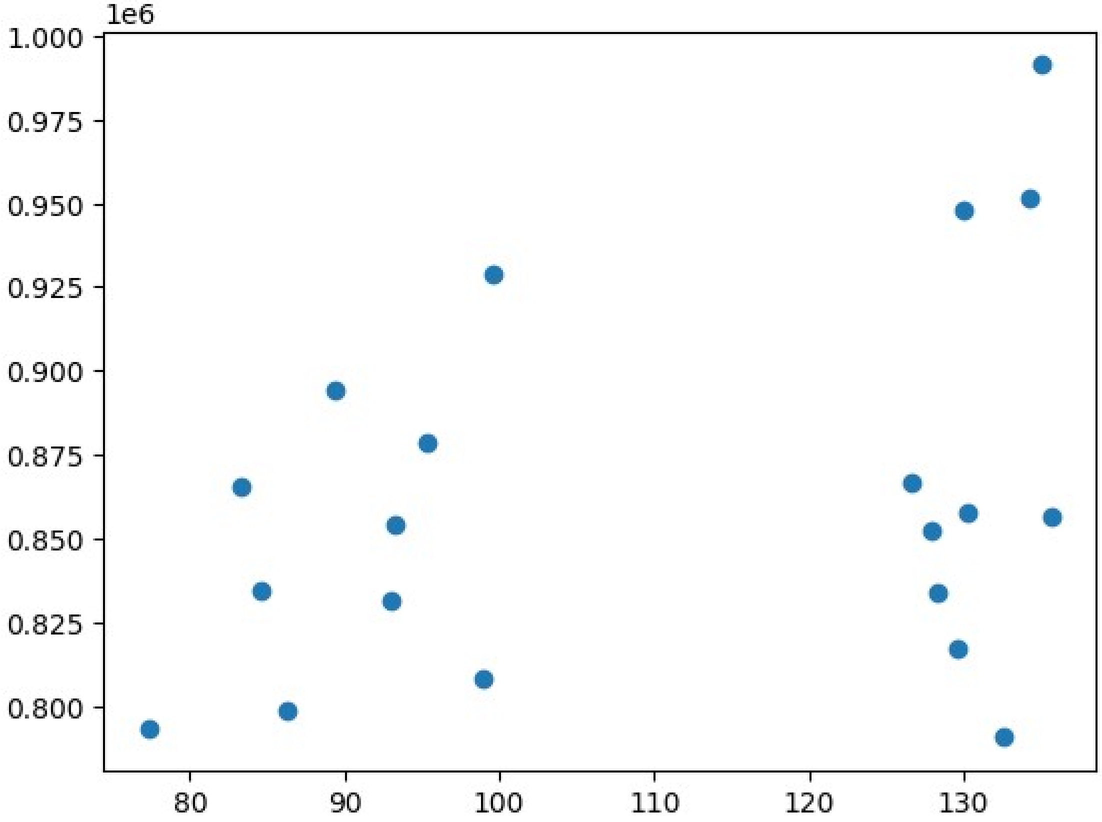
Scatterplot Graphs

```
menMeanSmarts = menDf[["PIQ", "FSIQ", "VIQ"]].mean(axis=1)
plt.scatter(menMeanSmarts, menDf["MRI_Count"])
plt.show()
%matplotlib inline
```





```
womenMeanSmarts = womenDf[["PIQ", "FSIQ", "VIQ"]].mean(axis=1)
plt.scatter(womenMeanSmarts, womenDf["MRI_Count"])
plt.show()
%matplotlib inline
```



Calculating Correlation with Python

```
# Used pearson correlation coefficient to make correlation on the dataset
brainFrame.corr(method='pearson')
```

```
<ipython-input-8-4d3089cc6357>:1: FutureWarning: The default value of numeric_only ir
  brainFrame.corr(method='pearson')
```

	FSIQ	VIQ	PIQ	Weight	Height	MRI_Count
FSIQ	1.000000	0.946639	0.934125	0.051483	0.086992	0.357641

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

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

```
<ipython-input-69-01fad84dd5db>:1: FutureWarning: The default value of numeric_only i
  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

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

```
<ipython-input-11-4396b7a1db7e>:1: FutureWarning: The default value of numeric_only i
  menDf.corr(method='pearson')
```

	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

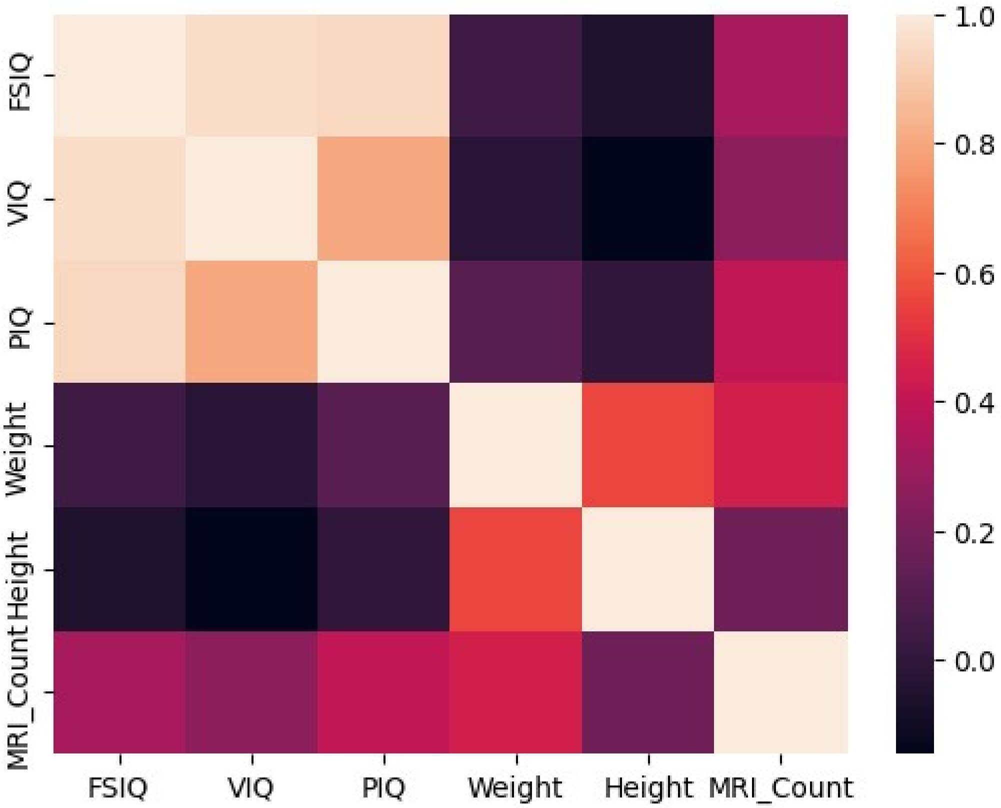
Visualization

```
!pip install seaborn
```

```
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packages (0.
Requirement already satisfied: numpy!=1.24.0,>=1.20 in /usr/local/lib/python3.10/dist
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/c
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (f
```

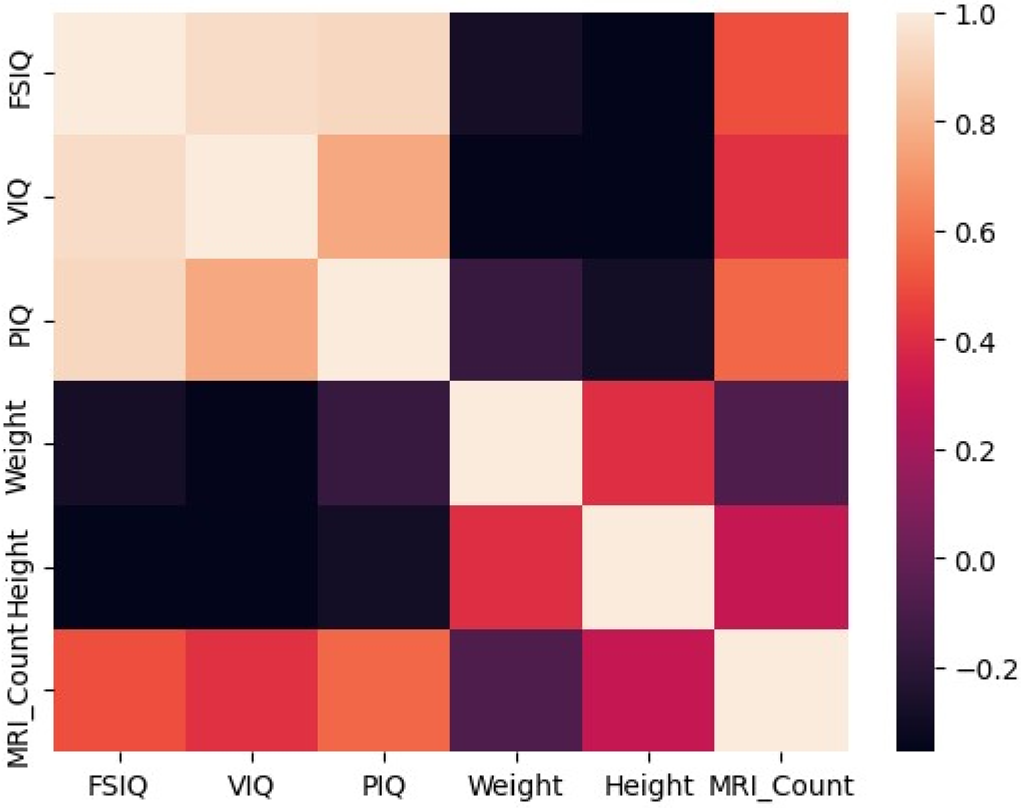
```
import seaborn as sns
wcorr = womenDf.corr()
sns.heatmap(wcorr)
#plt.savefig('attribute_correlations.png', tight_layout=True)
```

```
<ipython-input-13-424452bfc0e4>:2: FutureWarning: The default value of numeric_only i
  wcorr = womenDf.corr()
<Axes: >
```



```
mcorr = menDf.corr()
sns.heatmap(mcorr)
sns.heatmap(mcorr)
#plt.savefig('attribute_correlations.png', tight_layout=True)
```

```
<ipython-input-14-77e80db358d6>:1: FutureWarning: The default value of numeric_only i
  mcorr = menDf.corr()
<Axes: >
```



Questions:

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

> This means that there are close to no correlation between the variable and the size of a subjects's brain. For this dataset, it shows that there are no correlation between a subject's height and weight with their brain size.

2. Why separate the genders?

> By separating genders, it helps generate cleaner and precise data as each gender can have significant differences in their physical characteristics that can affect/ alter the results of the data.

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

> FSIQ, VIQ, and PIQ are the variables that presents stronger correlation with the brain size of the subjects. Yes, I think it is expected. These variables are directly

> brain size of the subjects. Yes, I think it is expected. These variables are directly connected with intelligence from which measures full scale, verbal, and performance IQ's of the subject. On the other hand, the variables that generated close to zero or negative values are focused more on the physical aspect of the subject which deemed no correlation with the brain size.

Supplementary Activity:

Student Exam Performance Prediction

The dataset is designed for predicting whether a student will pass or fail an exam based on the number of study hours and their scores in the previous exam.

```
examPerf = '/content/student_exam_data.csv'
examdf = pd.read_csv(examPerf)
```

```
examdf.head(10)
```

	Study Hours	Previous Exam Score	Pass/Fail
0	4.370861	81.889703	0
1	9.556429	72.165782	1
2	7.587945	58.571657	0
3	6.387926	88.827701	1
4	2.404168	81.083870	0
5	2.403951	49.757016	0
6	1.522753	94.655631	0
7	8.795585	89.352235	1
8	6.410035	96.987995	1
9	7.372653	83.543171	1

```
# This returns the number of rows and column inside the dataframe
examdf.shape
```

```
(500, 3)
```

```
examdf.describe()
```

	Study Hours	Previous Exam Score	Pass/Fail

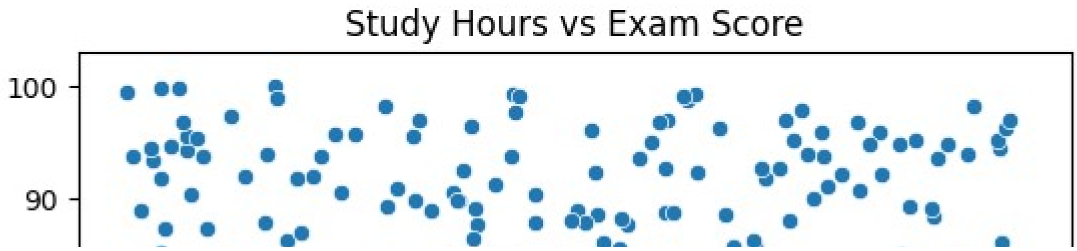
	Study Hours	Previous Exam Score	Pass/Fail
count	500.000000	500.000000	500.000000
mean	5.487055	68.917084	0.368000
std	2.688196	17.129607	0.482744
min	1.045554	40.277921	0.000000
25%	3.171517	53.745955	0.000000
50%	5.618474	68.309294	0.000000
75%	7.805124	83.580209	1.000000
max	9.936683	99.983060	1.000000

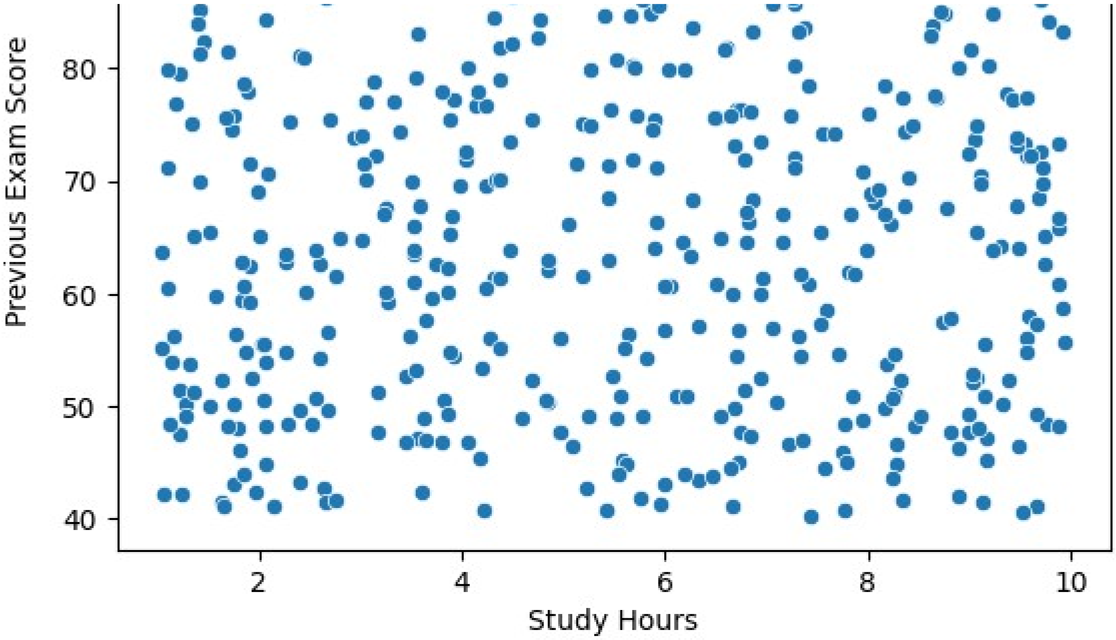
```
examdf.corr(method='pearson')
```

	Study Hours	Previous Exam Score	Pass/Fail
Study Hours	1.000000	0.010354	0.583505
Previous Exam Score	0.010354	1.000000	0.443706
Pass/Fail	0.583505	0.443706	1.000000

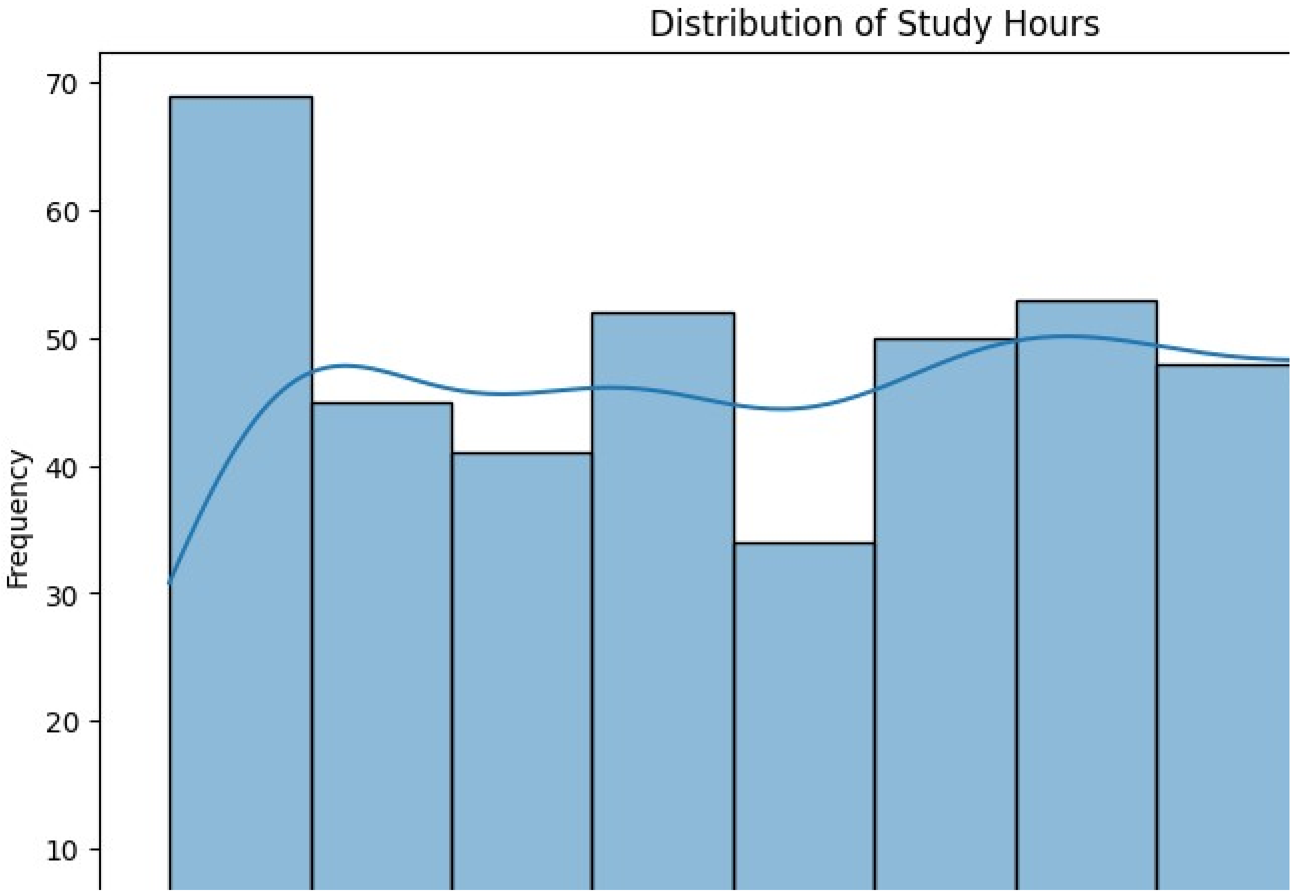
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt


# Correlation between science study hours and exam score
studyhourscorr = examdf['Study Hours'].corr(examdf['Previous Exam Score'])
print("Correlation between study hours and exam score:", studyhourscorr)

# Scatter plot
sns.scatterplot(x='Study Hours', y='Previous Exam Score', data=examdf)
plt.title('Study Hours vs Exam Score')
plt.xlabel('Study Hours')
plt.ylabel('Previous Exam Score')
plt.show()
```

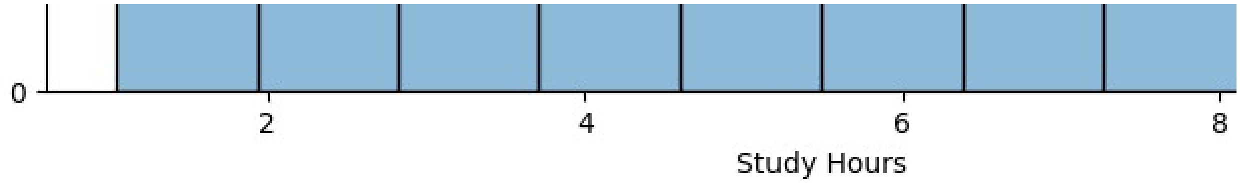
```
Correlation between study hours and exam score: 0.010354204028283442
```



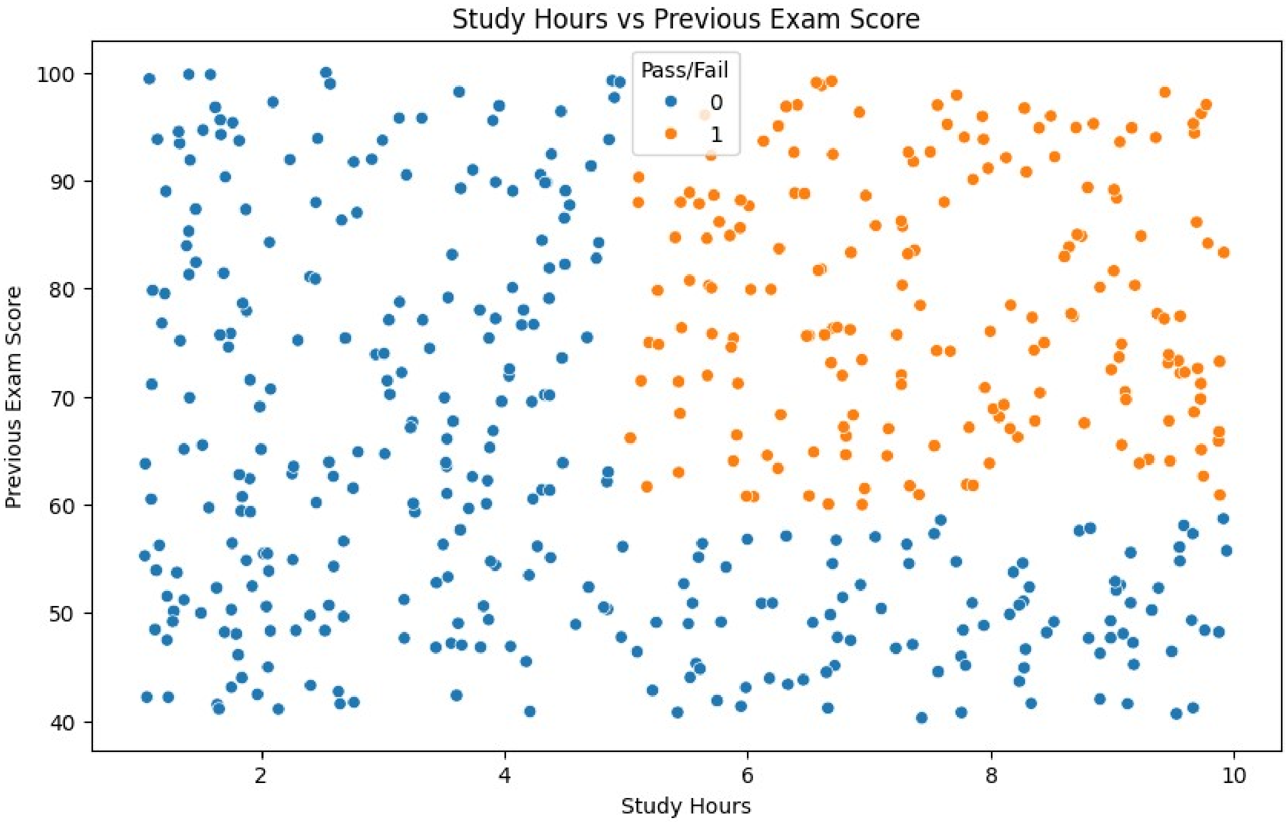


```
# Histogram of Study Hours
plt.figure(figsize=(10, 6))
sns.histplot(examdf['Study Hours'], bins=10, kde=True)
plt.title('Distribution of Study Hours')
plt.xlabel('Study Hours')
plt.ylabel('Frequency')
plt.show()
```

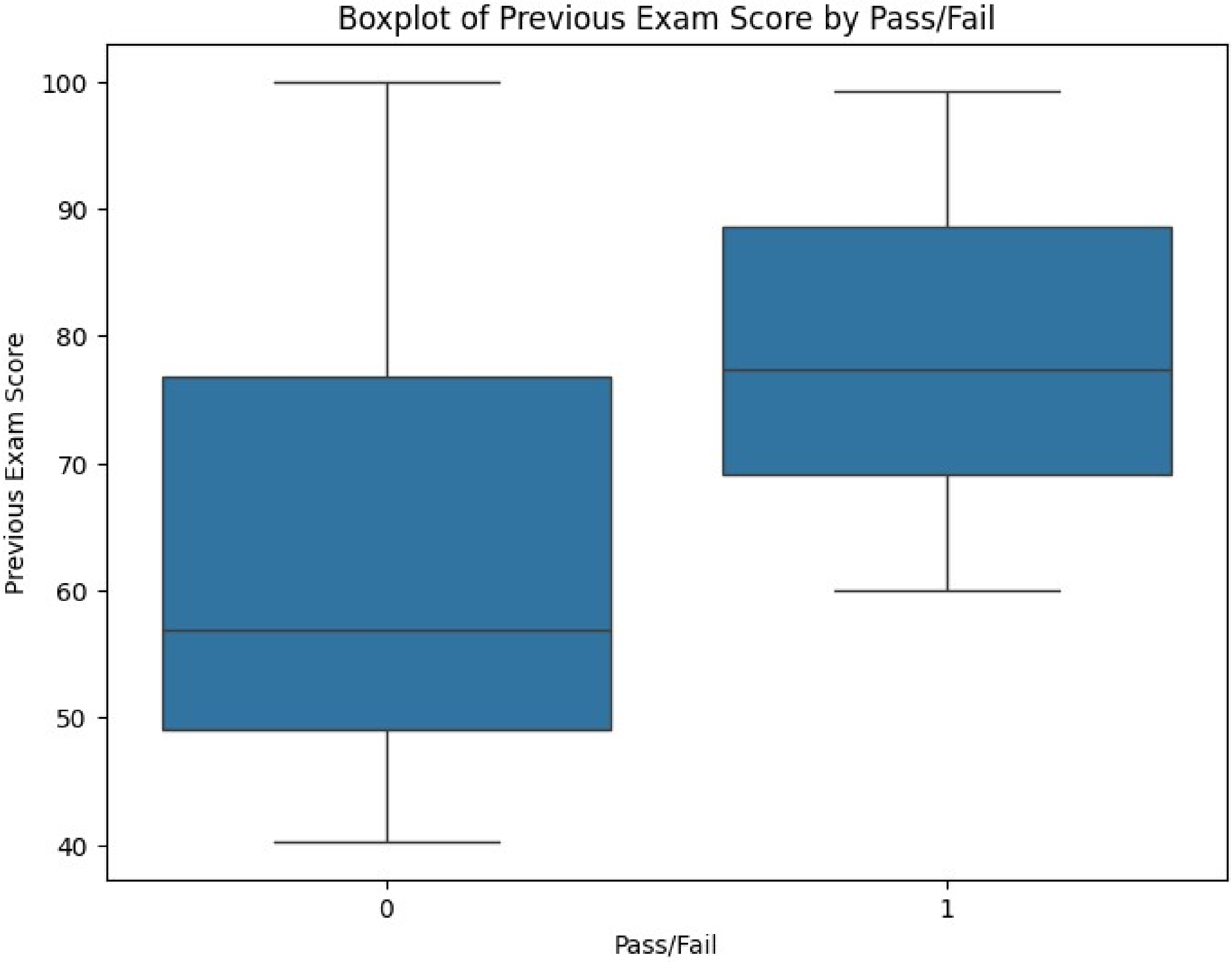




```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Study Hours', y='Previous Exam Score', hue='Pass/Fail', data=examdf)
plt.title('Study Hours vs Previous Exam Score')
plt.xlabel('Study Hours')
plt.ylabel('Previous Exam Score')
plt.legend(title='Pass/Fail')
plt.show()
```

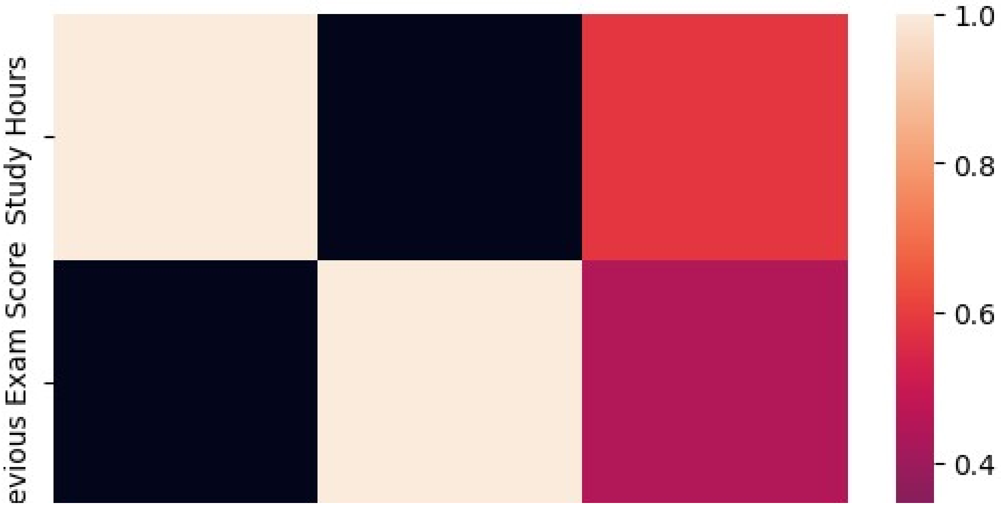


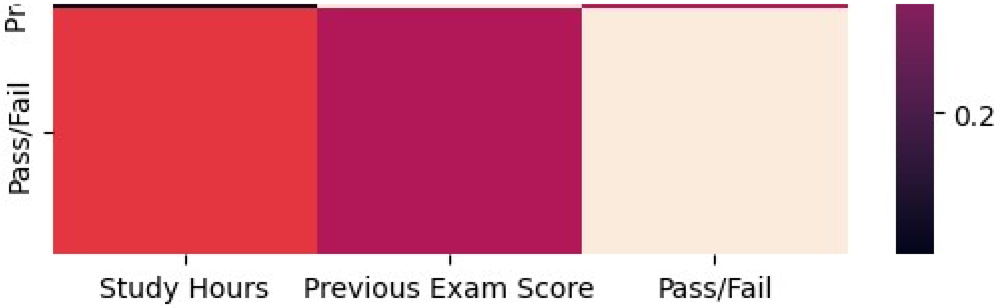
```
# Boxplot of Previous Exam Score by Pass/Fail
plt.figure(figsize=(8, 6))
sns.boxplot(x='Pass/Fail', y='Previous Exam Score', data=examdf)
plt.title('Boxplot of Previous Exam Score by Pass/Fail')
plt.xlabel('Pass/Fail')
plt.ylabel('Previous Exam Score')
plt.ylabel('Previous Exam Score')
plt.show()
```



```
score = examdf.corr()
sns.heatmap(score)
#plt.savefig('attribute_correlations.png', tight_layout=True)
```

```
<Axes: >
```





Conclusion: This activity helped think and make analysis about data sets and how can I manipulate it to generate correlations and results.