Name: Buenafe, Lorenz Angelo N.
Course & Section: CPE 019 - CPE32S9

Part 1: The Dataset

Step 1: Loading the Dataset From a File. Before the dataset can be used, it must be loaded onto memory. In the code below, The first line imports the pandas modules and defines pd as a descriptor that refers tothe module. The second line loads the dataset CSV file into a variable called brainFile . The third line uses read_csv() , a pandas method, to convert the CSV dataset stored in brainFile into adataframe. The dataframe is then stored in the brainFrame variable. Run the cell below to execute the described functions.

```
import pandas as pd
brainFile = '/content/brainsize.txt'
brainFrame = pd.read_csv(brainFile, sep = "\t")
```

Step 2: Verifying the dataframe. To make sure the dataframe has been correctly loaded and created, use the head() method. AnotherPandas method, head() displays the first five entries of a 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

Step 1: The pandas describe() method. The pandas module includes the describe() method which performs same common calculationsagainst a given dataset. In addition to provide common results including count, mean, standarddeviation, minimum, and maximum, describe() is also a great way to quickly test the validity of thevalues 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

Step 2: Scatterplot graphs Scatterplot graphs are important when working with correlations as they allow for a quick visualverification of the nature of the relationship between the variables. This lab uses the Pearson correlationcoefficient, which is sensitive only to a linear relationship between two variables. Other more robustcorrelation methods exist but are out of the scope of this lab.

a. Load the required modules. Before graphs can be plotted, it is necessary to import a few modules, namely numpy and matplotlib .Run the cell below to load these modules.

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

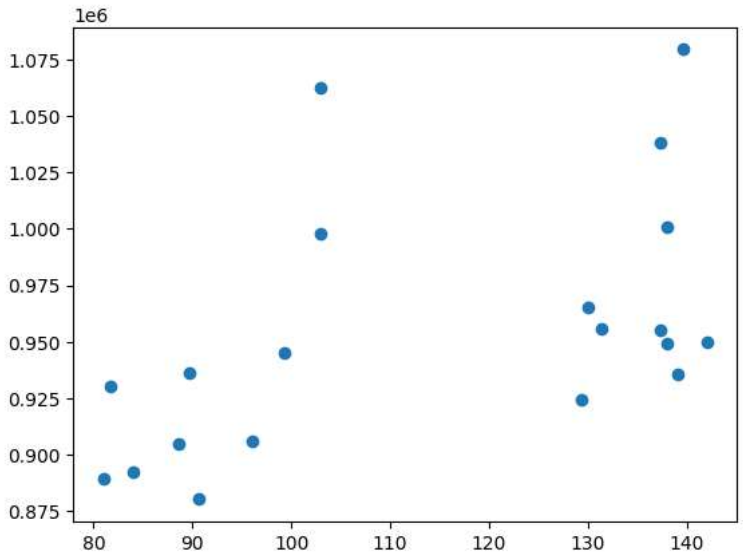
b. Separate the data. To ensure the results do not get skewed because of the differences in male and female bodies, thedateframe is split into two dataframes: one containing all male entries and another with only femaleinstances.

Running the cell below creates the two new dataframes, menDf and womenDf, each one containing therespective entries.

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

c. Plot the graphs. Because the dataset includes three different measures of intelligence (PIQ, FSIQ, and VIQ), the first linebelow uses Pandas mean() method to calculate the mean value between the three and store the result inthe menMeanSmarts variable. Notice that the first line also refers to the menDf, the filtered dataframecontaining only male entries. The second line uses the matplotlib method scatter() to create a scatterplot graph between the menMeanSmarts variable and the MRI_Count attribute. The MRI_Count in this dataset can be thought as ofa measure of the physical size of the subjects' brains. The third line simply displays the graph. The fourth line is used to ensure the graph will be displayed in this notebook.

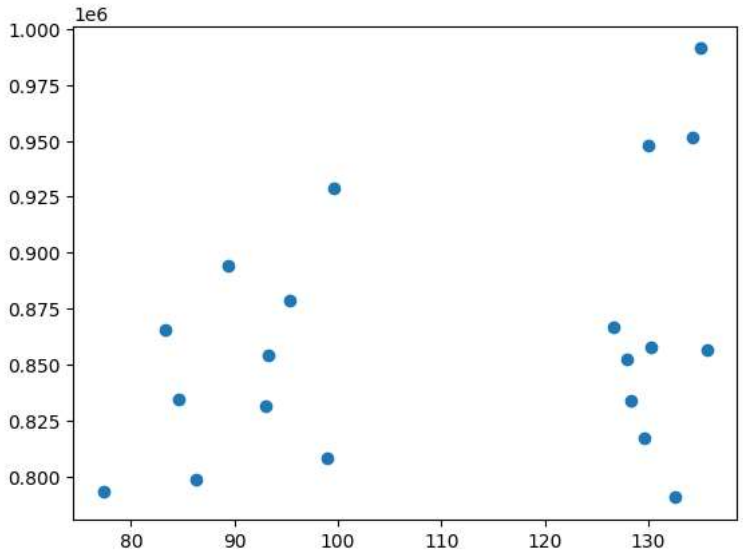
```
menMeanSmarts = menDf[["PIQ", "FSIQ", "VIQ"]].mean(axis=1)
plt.scatter(menMeanSmarts, menDf["MRI_Count"])
plt.show()
%matplotlib inline
```



Similarly, the code below creates a scatterplot graph for the women-only filtered dataframe.

```
#Graph the women-only filtered dataframe
#womenMeanSmarts = ?
#plt.scatter(?, ?)

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



Part 3: Calculating Correlation with Python

Step 1: Calculate correlation against brainFrame. The pandas corr() method provides an easy way to calculate correlation against a dataframe. By simplycalling the method against a dataframe, one can get the correlation between all variables at the sametime.

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

```
<ipython-input-24-4d3089cc6357>: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 filledwith 1s? Is that a coincidence? Explain.

- Because we used the pearson method

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.

- It is not a coincidence because they are formatted in the same way

Using the same corr() method, it is easy to calculate the correlation of the variables contained in thefemale-only dataframe:

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

```
<ipython-input-25-01fad84dd5db>: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

And the same can be done for the male-only dataframe:

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

```
<ipython-input-26-4396b7a1db7e>:1: FutureWarning: The default value of numeric_only in
  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

Part 4: Visualizing

Step 1: Install Seaborn. To make it easier to visualize the data correlations, heatmap graphs can be used. Based on coloredsquares, heatmap graphs can help identify correlations in a glance. The Python module named seaborn makes it very easy to plot heatmap graphs. First, run the cell below to download and install the seaborn module.

```
!pip install seaborn
```

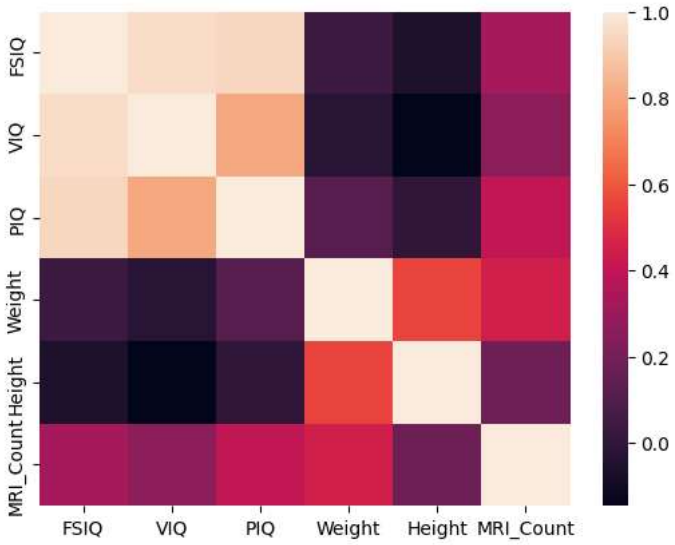
```
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packages (0.13.1)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in /usr/local/lib/python3.10/dist-packages (from seaborn) (1.23.5)
Requirement already satisfied: pandas>=1.2 in /usr/local/lib/python3.10/dist-packages (from seaborn) (1.5.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in /usr/local/lib/python3.10/dist-packages (from seaborn) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.2
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (23.2
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.1
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.2->seaborn) (2023.4)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4-
```

Step 2: Plot the correlation heatmap. Now that the dataframes are ready, the heatmaps can be plotted. Below is a breakdown of the code in thecell below: Line 1: Generates a correlation table based on the womenNoGenderDf dataframe and stores it on wcorr . Line 2: Uses the seaborn heatmap() method to generate and plot the heatmap. Notice that heatmap() takes wcorr as a parameter. Line 3: Use to export and save the generated heatmap as a PNG image. While the line 3 is not active (ithas the comment # character preceding it, forcing the interpreter to ignore it), it was kept forinformational purposes.

```
import seaborn as sns

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

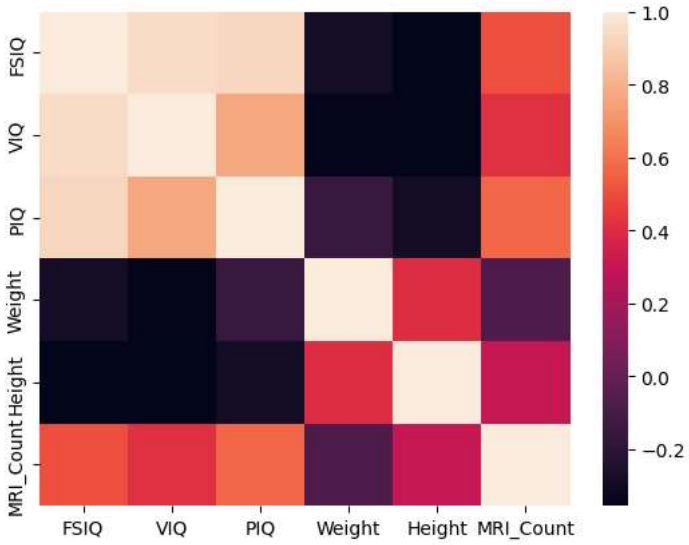
```
<ipython-input-28-307f963f6b56>:3: FutureWarning: The default value of numeric_only in
  wcorr = womenDf.corr()
<Axes: >
```



Similarly, the code below creates and plots a heatmap for the male-only dataframe.

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

```
<ipython-input-29-7befc568629f>:1: FutureWarning: The default value of numeric_only in
  mcorr = menDf.corr()
<Axes: >
```



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