Name: Albo, Russel Zen D.
Course and Section: CPE32S9
Date of Submission: February 9, 2024
Instructor: Engr. Roman Richard

Lab - Correlation Analysis in Python

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 arelinearly 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 Part3, you will use Python to calculate the correlation between two sets of variable.

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

You will use a dataset that contains a sample of 40 right-handed Anglo Introductory Psychologystudents at a large Southwestern university. Subjects took four subtests (Vocabulary, Similarities, BlockDesign, and Picture Completion) of the Wechsler (1981) Adult Intelligence Scale-Revised. Theresearchers 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 withheldthe weights of two subjects and the height of one subject for reasons of confidentiality. Two simplemodifications were applied to the dataset:

1. Replace the quesion 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 markscorrectly.
2. Replace all tab characters with commas, converting the dataset into a CSV dataset.

The prepared dataset is saved as brainsize.txt .

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(), apandas 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.

```
# Code cell 1
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.

```
# Code cell 2
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

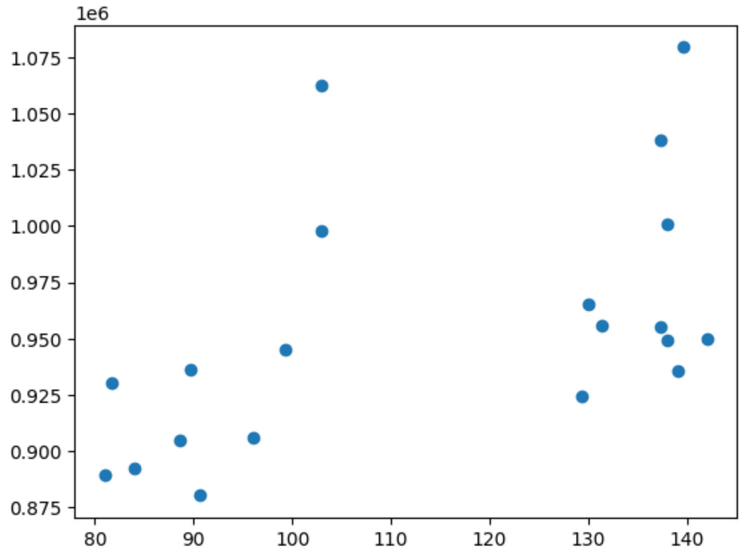
```
# Code cell 3
brainFrame.describe()
```

	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

```
# Code cell 4
import numpy as np
import matplotlib.pyplot as plt
```

```
# Code cell 5
menDf = brainFrame[(brainFrame.Gender == 'Male')]
womenDf = brainFrame[(brainFrame.Gender == 'Female' )]
```

```
# Code cell 6
menMeanSmarts = menDf[["PIQ", "FSIQ", "VIQ" ]].mean(axis=1)
plt.scatter(menMeanSmarts, menDf["MRI_Count"])
plt.show()
%matplotlib inline
```




```
# Code cell 7
# Graph the women-only filtered dataframe
#womenMeanSmarts = ?
#plt.scatter(?, ?)

plt.show()
%matplotlib inline
```

```
# Code cell 8
brainFrame.corr(method='pearson')
```

```
<ipython-input-32-cab48f3abe05>:2: 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

```
# Code cell 9
womenDf.corr(method='pearson')
```

```
<ipython-input-33-a6271751808a>:2: 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

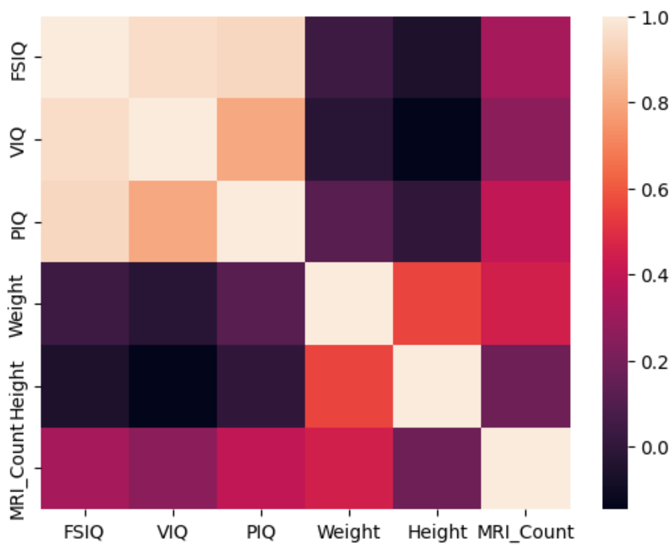
```
# Code cell 10
# Use corr() for the male-only dataframe with the pearson method
#?.corr(?)
```

```
# Code cell 11
!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-
```

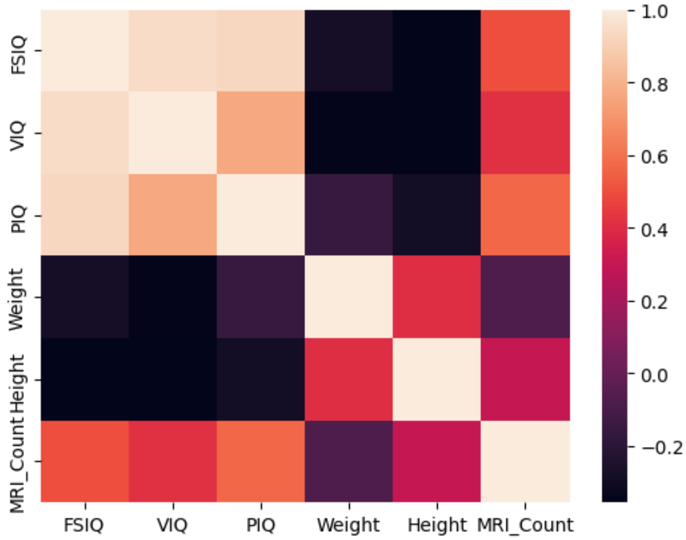
```
# Code cell 12
import seaborn as sns

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

```
<ipython-input-36-4bc71e77167c>:4: FutureWarning: The default value of numeric_only in
  wcorr = womenDf.corr()
<Axes: >
```




```
# Code cell 14
mcorr = menDf.corr()
sns.heatmap(mcorr)
#plt.savefig('attribute_correlations.png', tight_layout=True)
```

```
<ipython-input-37-ff3e250059fc>:2: FutureWarning: The default value of numeric_only in
  mcorr = menDf.corr()
<Axes: >
```



Many variable pairs present correlation close to zero. What does that mean? Why separate the genders?

- The reason why many variables present correlation to zero because the two genders which is male and female have no relationship or signifies a negative relationship. And th reason on seperating genders because of finding out the difference between them with different result.

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

The variables are FSIQ, VIQ, and PIQ. These variables have a stronger correlation due to their value which is closer to 1.

Supplementary Activity

```
# Code cell 1
import pandas as pd
raisin = '/content/Raisin_Dataset - Raisin_Grains_Dataset.csv'
raisinFrame = pd.read_csv(raisin)
```

```
# Code cell 2
raisinFrame.head()
```

	Area	MajorAxisLength	MinorAxisLength	Eccentricity	ConvexArea	Extent	Perimete
0	87524	442.246011	253.291155	0.819738	90546	0.758651	1184.04
1	75166	406.690687	243.032436	0.801805	78789	0.684130	1121.78
2	90856	442.267048	266.328318	0.798354	93717	0.637613	1208.57
3	45928	286.540559	208.760042	0.684989	47336	0.699599	844.16
4	79408	352.190770	290.827533	0.564011	81463	0.792772	1073.25

```
# Code cell 2
raisinFrame.tail()
```

	Area	MajorAxisLength	MinorAxisLength	Eccentricity	ConvexArea	Extent	Perime
895	83248	430.077308	247.838695	0.817263	85839	0.668793	1129
896	87350	440.735698	259.293149	0.808629	90899	0.636476	1214
897	99657	431.706981	298.837323	0.721684	106264	0.741099	1292
898	93523	476.344094	254.176054	0.845739	97653	0.658798	1258
899	85609	512.081774	215.271976	0.907345	89197	0.632020	1272

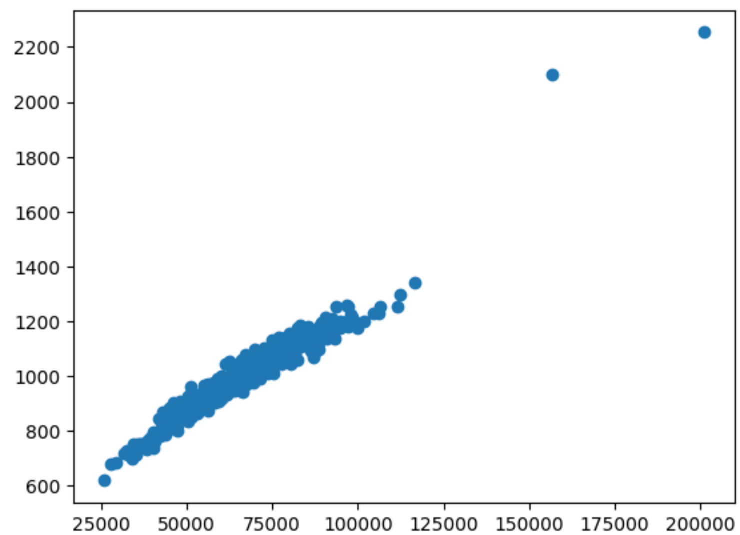
```
# Code cell 2
raisinFrame.describe()
```

	Area	MajorAxisLength	MinorAxisLength	Eccentricity	ConvexArea	
count	900.000000	900.000000	900.000000	900.000000	900.000000	900
mean	87804.127778	430.929950	254.488133	0.781542	91186.090000	0
std	39002.111390	116.035121	49.988902	0.090318	40769.290132	0
min	25387.000000	225.629541	143.710872	0.348730	26139.000000	0
25%	59348.000000	345.442898	219.111126	0.741766	61513.250000	0
50%	78902.000000	407.803951	247.848409	0.798846	81651.000000	0
75%	105028.250000	494.187014	279.888575	0.842571	108375.750000	0
max	235047.000000	997.291941	492.275279	0.962124	278217.000000	0

```
# Code cell 4
import numpy as np
import matplotlib.pyplot as plt
```

```
# Code cell 5
KecimenDf = raisinFrame[(raisinFrame.Class == 'Kecimen')]
BesniDf = raisinFrame[(raisinFrame.Class == 'Besni' )]
```

```
# Code cell 6
KecimenMeanSmarts = KecimenDf[["Area", "ConvexArea"]].mean(axis=1)
plt.scatter(KecimenMeanSmarts, KecimenDf["Perimeter"])
plt.show()
%matplotlib inline
```




```
# Code cell 7

plt.show()
%matplotlib inline
```

```
# Code cell 8
raisinFrame.corr(method='pearson')
```

```
<ipython-input-90-bea6b8b61e1a>:2: FutureWarning: The default value of numeric_only in
  raisinFrame.corr(method='pearson')
```

	Area	MajorAxisLength	MinorAxisLength	Eccentricity	ConvexArea
Area	1.000000	0.932774	0.906650	0.336107	0.995920
MajorAxisLength	0.932774	1.000000	0.728030	0.583608	0.945031
MinorAxisLength	0.906650	0.728030	1.000000	-0.027683	0.895651
Eccentricity	0.336107	0.583608	-0.027683	1.000000	0.348210
ConvexArea	0.995920	0.945031	0.895651	0.348210	1.000000
Extent	-0.013499	-0.203866	0.145322	-0.361061	-0.054802
Perimeter	0.961352	0.977978	0.827417	0.447845	0.976612

```
# Code cell 9
BesniDf.corr(method='pearson')
```

```
<ipython-input-91-383a77451919>:2: FutureWarning: The default value of numeric_only in
  BesniDf.corr(method='pearson')
```

	Area	MajorAxisLength	MinorAxisLength	Eccentricity	ConvexArea
Area	1.000000	0.888452	0.895563	0.116798	0.993685
MajorAxisLength	0.888452	1.000000	0.621551	0.489437	0.909429 -
MinorAxisLength	0.895563	0.621551	1.000000	-0.299362	0.880913
Eccentricity	0.116798	0.489437	-0.299362	1.000000	0.135419 -
ConvexArea	0.993685	0.909429	0.880913	0.135419	1.000000
Extent	0.146613	-0.111707	0.288752	-0.324046	0.093412
Perimeter	0.939498	0.963025	0.768241	0.283181	0.965320 -

```
# Code cell 12
import seaborn as sns

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

```
<ipython-input-93-17ea27171542>:4: FutureWarning: The default value of numeric_only in
  Bcorr = BesniDf.corr()
<Axes: >
```

```
# Code cell 14
Kcorr = KecimenDf.corr()
sns.heatmap(Kcorr)
#plt.savefig('attribute_correlations.png', tight_layout=True)
```

```
<ipython-input-94-cbd4eec7b77f>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future versio
  Kcorr = KecimenDf.corr()
<Axes: >
```