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 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.

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 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 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 marks correctly.**
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 to the 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 a dataframe. The dataframe is then stored in the `brainFrame` variable.

Run the cell below to execute the described functions.

In [ ]:
```
# Code cell 1
import pandas as pd
brainFile = './Data/brainsize.txt'
brainFrame = pd.read_csv(brainFile)
```

Step 2: Verifying the dataframe.

To make sure the dataframe has been correctly loaded and created, use the `head()` method. Another Pandas method, `head()` displays the first five entries of a dataframe.

In [ ]:
```
# Code cell 2
brainFrame.head()
```

Part 2: Scatterplot Graphs and Correlatable Variables

Step 1: The pandas `describe()` method.

The pandas module includes the `describe()` method which performs same common calculations against a given dataset. In addition to provide common results including count, mean, standard deviation, minimum, and maximum, `describe()` is also a great way to quickly test the validity of the values in the dataframe.

Run the cell below to output the results computed by `describe()` against the `brainFrame` dataframe.

In [ ]:
```
# Code cell 3
brainFrame.describe()
```

Step 2: Scatterplot graphs

Scatterplot graphs are important when working with correlations as they allow for a quick visual verification of the nature of the relationship between the variables. This lab uses the Pearson correlation coefficient, which is sensitive only to a linear relationship between two variables. Other more robust correlation 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.

In [ ]:
```
# Code cell 4
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, the dateframe is split into two dataframes: one containing all male entries and another with only female instances.

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

In [ ]:
```
# Code cell 5
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 line below uses Pandas `mean()` method to calculate the mean value between the three and store the result in the `menMeanSmarts` variable. Notice that the first line also refers to the menDf, the filtered dataframe containing 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 of a 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.

In [ ]:
```
# Code cell 6
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.

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

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 simply calling the method against a dataframe, one can get the correlation between all variables at the same time.

In [ ]:
```
# Code cell 8
brainFrame.corr(method='pearson')
```

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.

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.

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

In [ ]:
```
# Code cell 9
womenDf.corr(method='pearson')
```

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

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

Part 4: Visualizing

Step 1: Install Seaborn.

To make it easier to visualize the data correlations, heatmap graphs can be used. Based on colored squares, 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.

In [ ]:
```
# Code cell 11
!pip install seaborn
```

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 the cell 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 (it has the comment # character preceding it, forcing the interpreter to ignore it), it was kept for informational purposes.

In [ ]:
```
# Code cell 12
import seaborn as sns

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

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

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

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

Why separate the genders?

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