****

**Title of project:**

Descriptive statistics and Correlation calculator

**Team members’ names:**

آلاء سيد محمد محمد ( سكشن 4 )

رضوى سمير عبد الحميد موسى ( سكشن 7 )

مارلين مجدي سعد ( سكشن 12 )

مريم أبو الحمد عبد الباري هاشم ( سكشن 16 )

مي عماد الدين محمد ( سكشن 18 )

**Class:**

Class of 2017-2021

**Name of Doctor:**

Dr. Mahmoud Mounir

**Year:**

**2019**

**Introduction:**

**Topic:**

Analysis of descriptive statistics and prediction using correlation

**Project Goals:**

* Calculation of measures of central tendency (mean - median - mode).
* Calculation of measures of dispersion (range - IQR - variance – standard deviation )
* Drawing graphs and figures to represent samples of data.
* Calculation of correlation between two sets of data.

**Project direction:**

The project applies the rules we have studied in the statistics course this semester using python programming language.

It analyzes one dimensional data graphically and numerically.

It also performs correlation analysis on two dimensional data.

* Why python?

Programming languages which handle statistics like SAS or SPSS have drawbacks when it comes to the size of the data.  If one has to deal with these huge datasets, he or she should be proficient with Python or R. A given task can be handled with either of the languages because of the versatility of libraries available in both. Both these languages are domination the analysis and artificial intelligence domain of the tech industry. However, based on the surveys conducted, it is possible that Python will beat R in every aspect and take over.

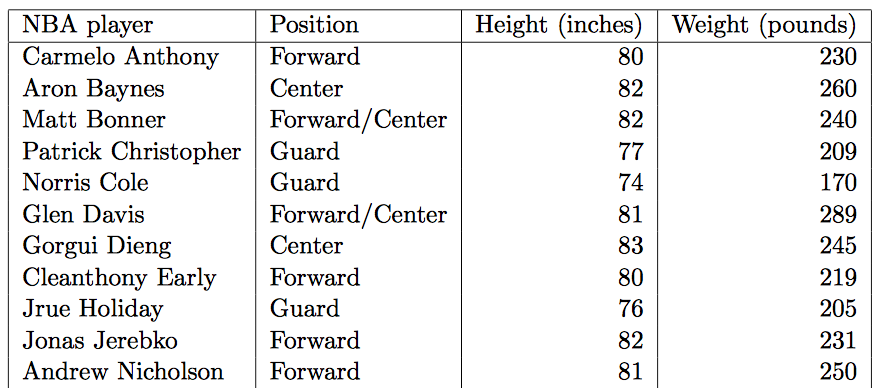
**Summary of research:**

**Research Question:**

With this evolving world where data is the key, there are two sides when it comes to handling the data — one where the data is used to make better decisions and the other where data is used to predict results which in turn help in making better decisions. This is where an analyst or a data scientist comes into the picture.

In our project, we are focused on exploring and analyzing the data itself, representing it graphically and calculating the relationship between it and other datasets.

For example suppose we have a sample of 11 basketball players with their heights and weights given in the following table and we need to study this sample (its central tendency and its variance) using our project.

****

**Exploratory data analysis:**

* **Calculating measures of central tendency :**

First, we are going to focus on the first dimension of this sample which is basketball players’ heights. We want to describe the center of the distribution. We can do so by calculating the following.

1. **Mean: In this case, it’s the average age for a basketball player.**
2. **Median: The middle value of data when arranged from the smallest to largest.**
3. **Mode: The most frequently occurring value of the data.**

Now let’s calculate those using our projects.

1. First, the project was made by pyCharm IDE but for it to run properly on another device, make sure you have installed all the packages used in the project or you will get “a module not found” error.

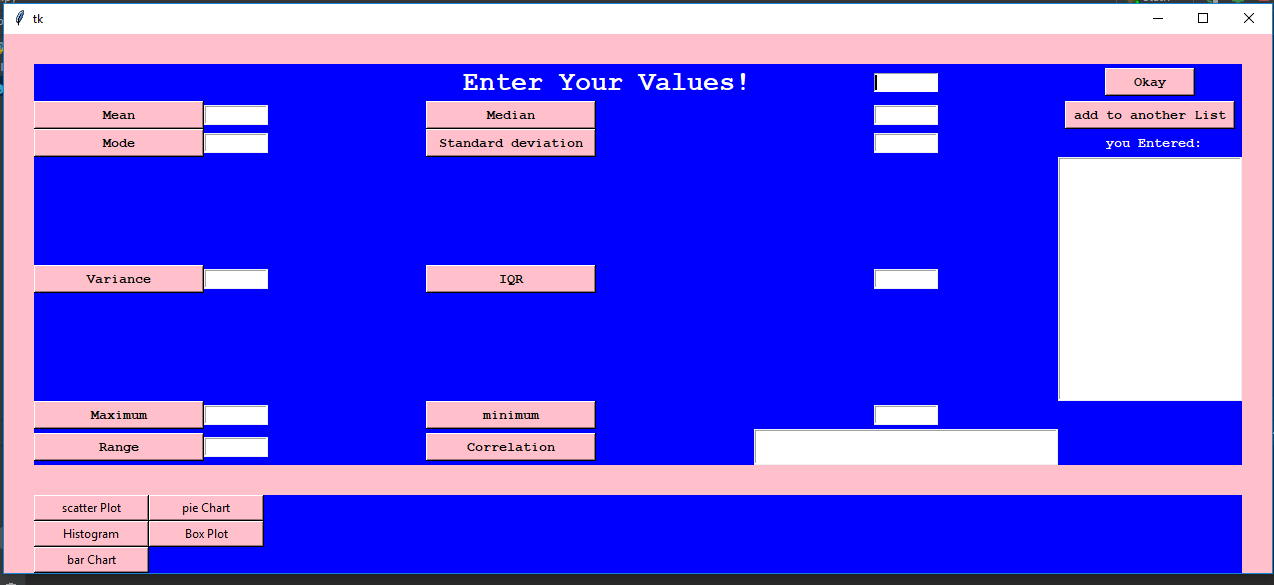
Most important packages in the project:

numpy: the fundamental package for array computing with python.

pandas: powerful data structures for data analysis, time series and statistics.

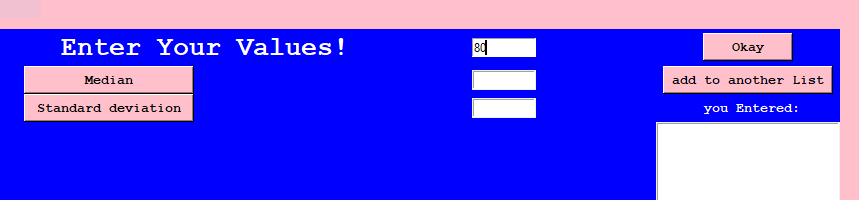
matplotlib: python plotting package.

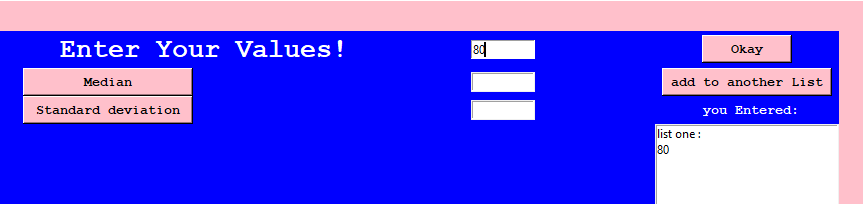
1. Now, this is the frame that appears when we first run the project.
2. On the right side of the frame there are a textbox where the user enters their list of data and another textbox where this list of entered data appears so that the user can keep track of the data they entered. On the left side, there are the buttons that calculate the descriptive statistics analysis and there are textboxes next to each one of them where the result appears. In the bottom left, there are the buttons that draws the graphs and figures that represent data. When the user clicks on any of them another frame that contains the drawn graph appears.

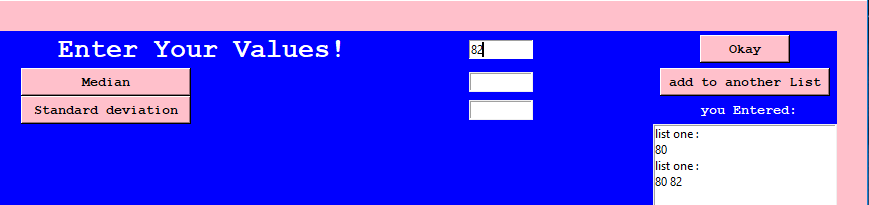


Now, let’s return to calculating the mean, median and mode for those basketball players using this interface of our project.

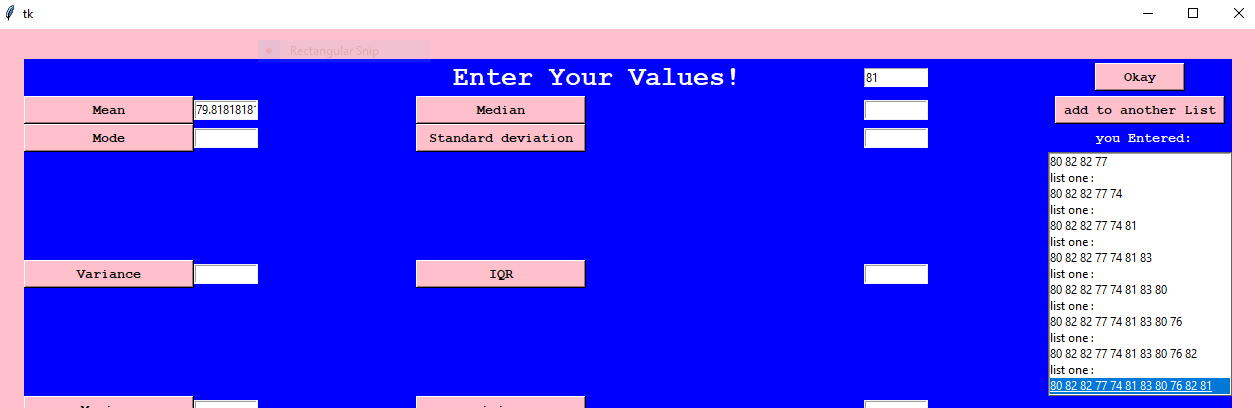
Enter the first value of your sample in this textbox then press the okay button.

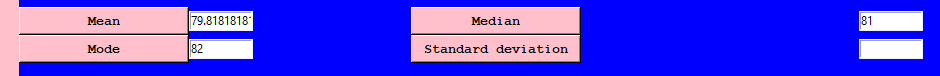




The data you entered appears on the textbox in the right updated with each new element you add in the list:

Now that you have entered the data you can use the buttons in the left to calculate the mean, median and mode.

Let’s start by the mean:

Then the median and the mode:

To do so, we used the built in functions: **mean(data), median(data) and mode(data)** from statistics module.

From the results we notice that the mean < median < mode which implies that this sample has left skewness.

Note that: The larger the sample is, the better values for mean, median and mode.

* **Calculating measures of variance:**

Yet to adequately describe a distribution we need more information. We also need information about the variability or dispersion of the data. We need, in other words, measures of dispersion.

1. **Range : Maximum value – minimum value**

**It’s easy to understand and simple to compute but it doesn’t give a good impression of the variability as it only takes into account the extreme values**

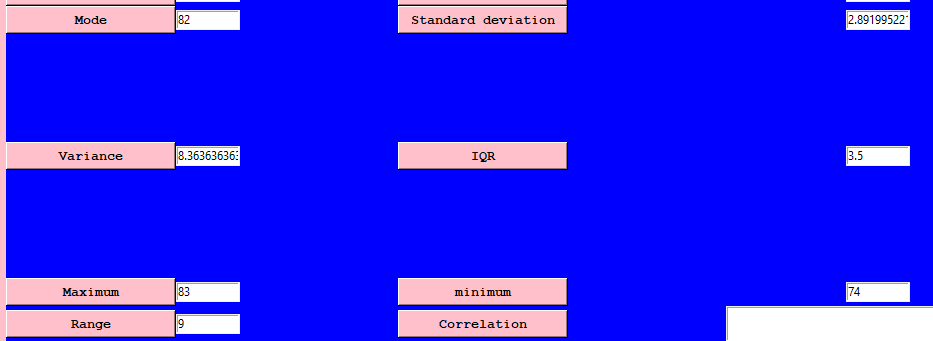
1. **Interquartile range : The third quartile – the first quartile**

**Distribution in 4 equal parts and it leaves out the extreme values.**

1. **Variance: Takes into account all the values.**
2. **Standard deviation: Variance’s square root.**

Now let’s calculate these using our python interface.

Just simply click on the buttons:



To calculate range, we used the built in functions: **min(data), max(data)** to get the largest and smallest number then subtracted them.

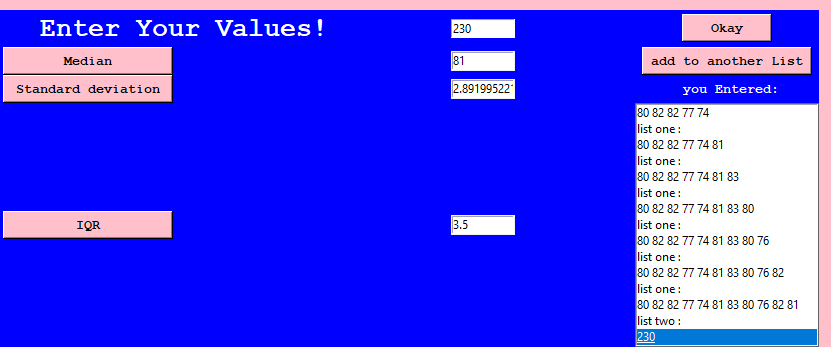
As for IQR, we used **numpy.percentile** function to get q1 and q3 then subtracted them.

Variance and standard deviation were calculated using **statistics.variance(data)**.

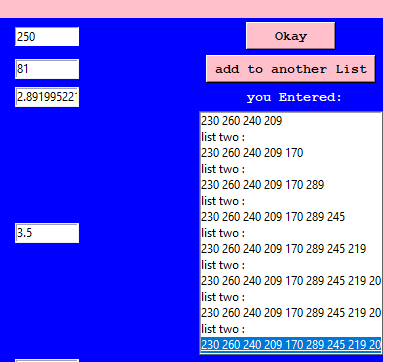
* **Calculating correlation :**

Correlation represents the relationship between two datasets.

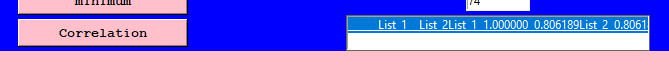
So now is the time to enter the players’ weights and see if there’s a relationship between a player’s height and their weight.

To add another list to the program, enter the data in the same textbox but instead of clicking okay click add to another list every time you want to add an element to list two.

Keep adding elements to the second list and press add to another list each time you want to add an element. List two appears on the textbox on the right. List is updated with each new element you add.



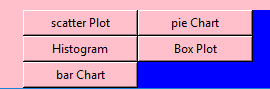
After entering the players’ weights, press the correlation button to find out if there is a relationship between player’s heights and their weights.



The correlation result between the two lists is 0.806189 which actually means that there is a positive strong correlation. There is a strong positive relationship between a player’s height and his weight.

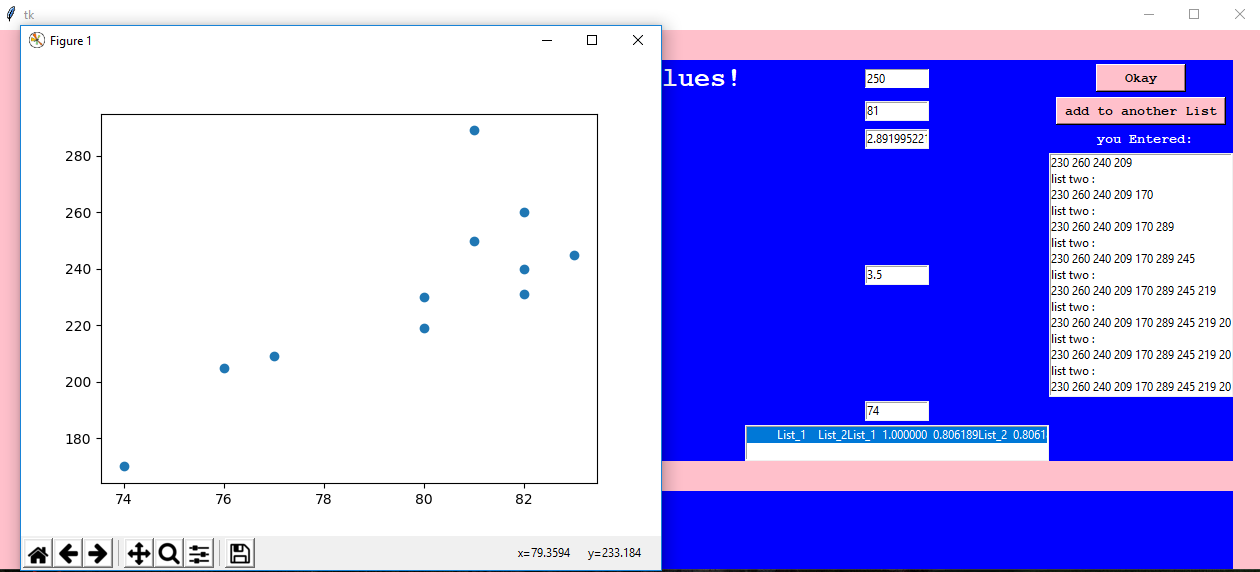
To calculate correlation, we use **DataFrame** from **pandas** module to add the two lists together in a data frame so that we can use them to call **dataframe**.**corr()** function.

* Data visualization and representation:



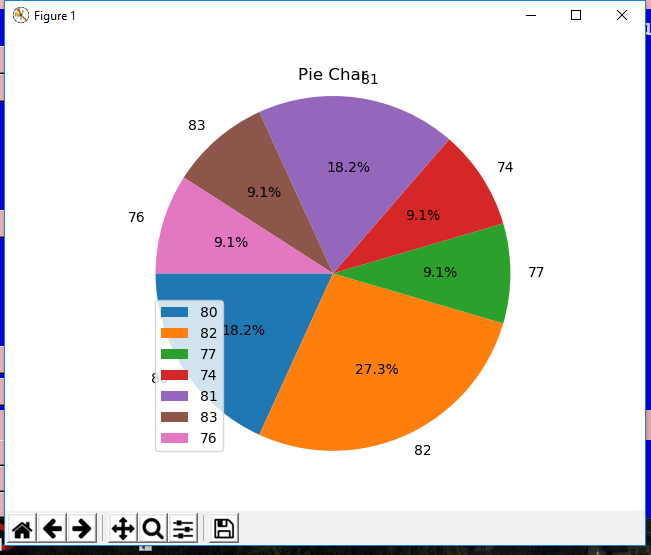
1. Scatter plots :

* Used to represent correlation between two data sets so only try to draw it if you have added two lists. Otherwise, it will show an error message box.



The function used to draw this shape is matplotlib.pyplot.scatter(data1, data2).

1. Pie Chart:

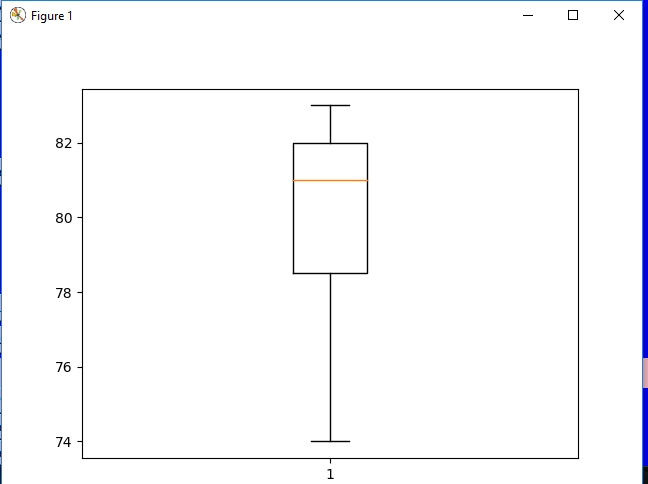
It represents the percentage of the first list. You can easily recognize the mode by looking at a pie chart as it has the biggest part of the pie.

**matplotlib.pyplot.pie()**is the built in function used to draw this pie chart.

# Histogram : The histogram used here represents the data for two data sets using matplotlib.pyplot.hist2d(data1,data2).

1. Box plot : The box plot is the most popular way to represent the variation of data.

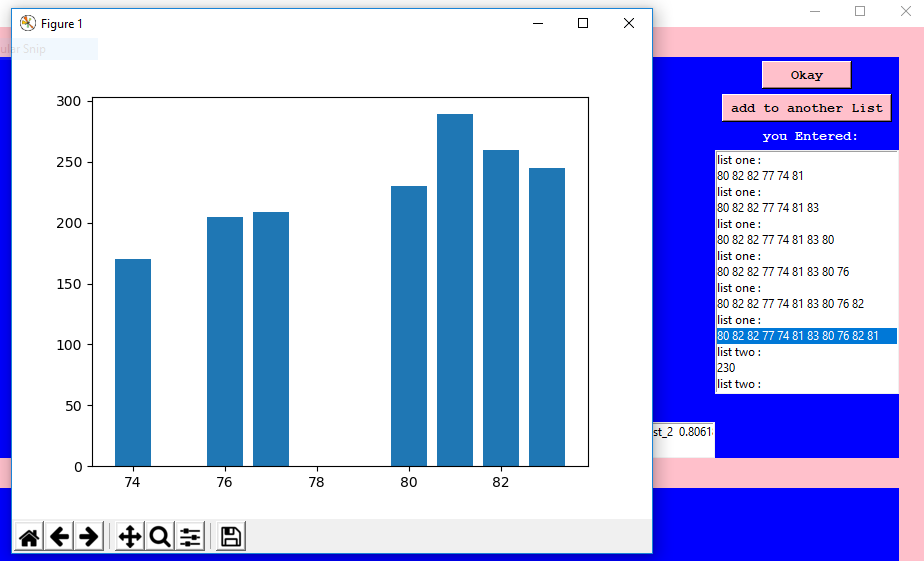
Here, we used[**matplotlib.pyplot.boxplot(data)**](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.boxplot.html?highlight=boxplot#matplotlib.pyplot.boxplot)to draw the box plot to represent the variation in player’s heights.



1. Bar chart:

The bar chart used in this project represents two dimensional data. Therefore, make sure you entered two data sets before clicking this button.

[**matplotlib.pyplot.bar**](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.bar.html?highlight=bar#matplotlib.pyplot.bar)**(data1, data2)** is the built in function that we used to draw this figure.

****