Carnegie Mellon University Africa

DATA, INFERENCE AND APPLIED MACHINE LEARNING

ASSIGNMENT 1

Course Code: 18-785

Name : Ange Izabayo

Email: aizabayo@andrew.cmu.edu

Submission Date on 04th September 2023

Report

Introduction in this assignment I used Jupyter notebook IDE to write the code and run them, as it is data related assignment, I imported different library to help in the process of executing the code. In the first cell of Jupyter notebook it is where libraries are declared, which include pandas for reading and analysing data file, matplotlib plotting graph and related task, math for mathematical functions, SciPy as interpolate to deal with estimation and finding the missing values. Beside libraries there are built in functional used and are explained as there are needed.

Q1. I began by declaring variable to store the thickness of the paper, height of mount Everest, and setting initial variable of number of times to zero. Initially, with the first fold, the thickness doubles. With each subsequent fold, it doubles again and become four times thicker on second time, and so on. I used a while loop to check if the thickness of the folded paper is greater than the height of the Mount Everest. if the condition I set is not met, the loop continues to fold paper. The loop continues to fold until the condition become true and the result is, it takes 24 folds for the thickness of the paper to exceed the height of the Mount Everest if thickness of paper is 1mm.

Q2. I began by importing the math library, which I will use in the mathematical function. Next, I initialized the variables as given. Using the provided formula, I calculated the time it will take for the volume to decrease to less than half of the initial volume. when I ran the code, the output indicate that it will take is 6.9 second for volume to decrease a half of it is initial.

Q3. Given data amount deposited; interest rate offered annually. Asked the amount of money you will be given in first year to five years. I declared these variable as they were given create the number of years, we need to generate income in range from of five years. By using dong for loop, it will continue to iterate and finding the earn by the result I printed the five number of years to

Corresponding amount in each year. After data I created the sum of that will be earned after five eyes.

Q4. I began this question by declaring variables to store given data and initializing a list to represent the loan payment periods in years. Then I used a for loop to iterate through each year, displaying the payment details for each year. Within the loop, I calculated the monthly payment with interest and used built-in round function to round to the nearest dollar as it was asked. Then I ran the code to present the amount to be paid in each year. Then to make it clear I added line to show the whole amount that will be paid in three years.

Q5. I began this question by importing matplotlib a library in python that used in data plotting and visualization [1], as I am using Jupyter notebook, this library is embedded in it so it does not require me to install it. I imported interpolate from SciPy. I declared variables to store given data of the scenario which include Investment, Customers, customer grow rate, and profit of each customer.

Used while loop to check if the condition is true or not, the condition is to check if the profit generated exceed the initial investment if it does display the day it will happen In that way I initially declared different variable to cope with the conditions number of day by increment by one at every condition to keep track the days, where days is list that work in every iteration. And I performed some mathematical calculation of determining profit, rate at which customers are increasing, total profit which goes with the days.

I used interpolation to estimate the day when the accumulated profit has reached to 100,000 $ by using SciPy as interpolate to create function that will calculate the approximate day profit will be fully accumulated. Then after that I created code to plot graph of accumulated profit against number of days as shown in figure1. In the graph I declared code to mark Initial Investment with green dash and marking Breakeven day with red dashed line.

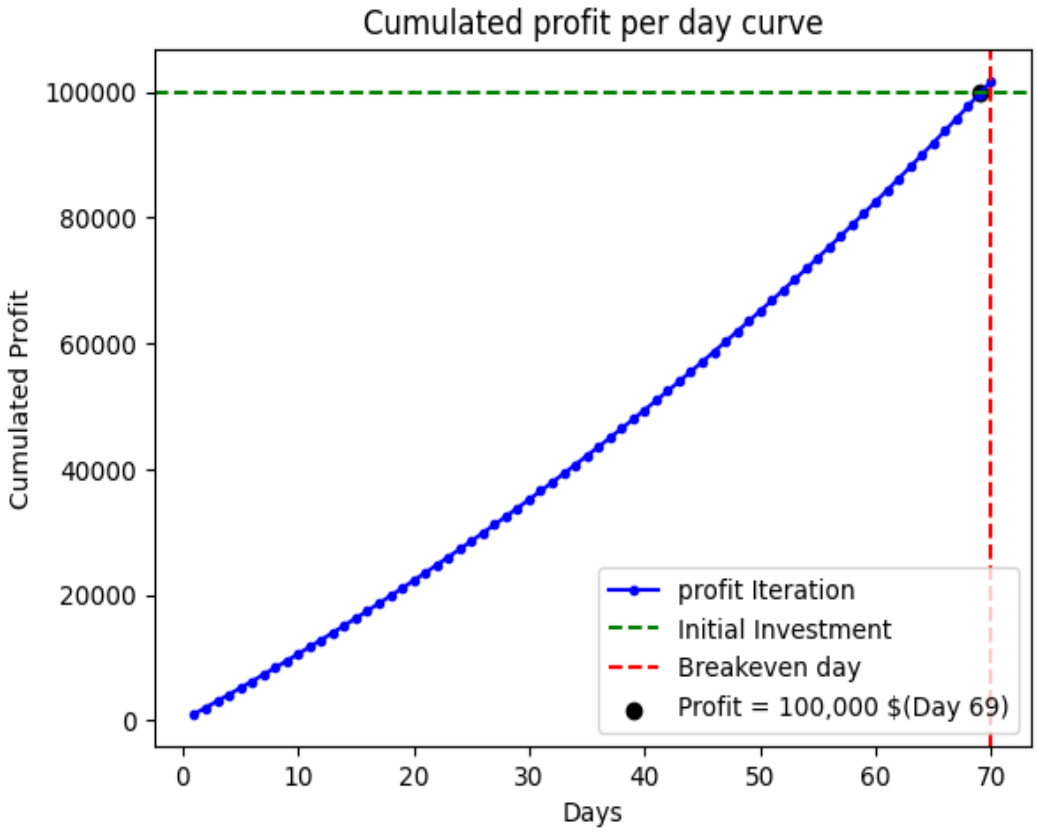


Figure 1Cumulated profit against Days

In summary this case of setting business and calculating the number days it will return the initial investment has acquired me with skills of knowing how to calculate and learning different module which include interpolation for estimating or making approximation, according to this case it will turn the profit on day 70, and Visualization of the data on the graph

Q6. Given Ebola data as case study where it contains data of Date, Cases, Death, Diff, Noofdays. I began this question by reading it and importing the library I will use first I imported pandas to help me to load data file and to use in various functions [2]. and used matplotlib to plot graph because I have used it in previous question, and I am using the same notebook there is no need to import it again just using it. I used the syntax used to load the datafile into Jupyter notebook to analyse them and it has 80 rows and 5 columns.

The first task was to extract data of Cases, deaths, and Dates from data file and then print it which shows the data of those three only excluding Diff and Noofdays data. There are dates which was not included in the file and the task was to

- By using built in date range function I managed to track the missing dates between starting date to ending date and then use date_range function which is used to create fixed frequency date time index [3] and then print out the date which were missing in dataset. I created variable

Missing Date to contain all date which were missing between starting date 2014-03-22, to end date 2014-11-12. Created dataframe of the missing date to help me to merge it with the current data frame. I Merged the Missing dataframe with the original dataframe given using concatenation and then sort the date from first to the last date respectively, as it is shown in figure 2 the rows increased to 236 and columns remain 5. When the process of Merging and sorting completed, I updated the dataset to and name it new_ebola_download shown in the image. Through merging Date, Cases and Deaths have no value as well as Diff and Nooffdays. By using interpolation, I created program to fill the missing values linearly in Cases and Deaths slot, shown in the figure 2 before with missing value in Cases and Death, and afterward in figure 3.

```
the Original DataFrame is Merged with Missing Date Dataframe

          Date    Cases   Death     Diff  Noofdays
0   2014-03-22     49.0    29.0      2.0       1.0
1   2014-03-23      NaN     NaN      NaN       NaN
2   2014-03-24     86.0    59.0      1.0       3.0
3   2014-03-25     86.0    60.0      1.0       4.0
4   2014-03-26     86.0    62.0      1.0       5.0
..         ...     ...     ...      ...       ...
231 2014-11-08      NaN     NaN      NaN       NaN
232 2014-11-09  14068.0  5496.0      2.0     233.0
233 2014-11-10      NaN     NaN      NaN       NaN
234 2014-11-11  14383.0  5492.0      1.0     235.0
235 2014-11-12  14413.0  5498.0 -41955.0     236.0

[236 rows x 5 columns]
updated data file
```

Figure 2 With missing values

```
Filling the missing values linearly of 'Cases','Death'

          Date    Cases   Death     Diff  Noofdays
0   2014-03-22     49.0    29.0      2.0       1.0
1   2014-03-23     67.5    44.0      NaN       NaN
2   2014-03-24     86.0    59.0      1.0       3.0
3   2014-03-25     86.0    60.0      1.0       4.0
4   2014-03-26     86.0    62.0      1.0       5.0
..         ...     ...     ...      ...       ...
231 2014-11-08  13894.4  5451.8      NaN       NaN
232 2014-11-09  14068.0  5496.0      2.0     233.0
233 2014-11-10  14225.5  5494.0      NaN       NaN
234 2014-11-11  14383.0  5492.0      1.0     235.0
235 2014-11-12  14413.0  5498.0 -41955.0     236.0

[236 rows x 5 columns]
```

Figure 3 After fill using interpolation

- Another task I made code to extract indices where case values and death value exceeding list of numbers provided, to approach this task I first declared the list and create empty variable to cases indices and other for deaths indices, then I used loop to check the condition of the limit value which it must exceed to be printing and after I Plotted the graph of number of cases and Deaths against Dates by using matplotlib library to plot Data. I used scatter plot function from matplot to visual representation of the number of death and Cases and I have printed them so with date and index they are on as shown in figure 4. In the graph I created code to point out where Cases or Death has exceeded the condition given with big dot on the line as shown in figure5.

```
                Exceedance Cases

For Cases > 100,the first exceedance is on 5 index, the corresponding date is 2014-03-27 00:00:00 with 111.0 cases
For Cases > 500,the first exceedance is on 83 index, the corresponding date is 2014-06-13 00:00:00 with 502.8 cases
For Cases > 1000,the first exceedance is on 115 index, the corresponding date is 2014-07-15 00:00:00 with 1004.0 cases
For Cases > 2000,the first exceedance is on 143 index, the corresponding date is 2014-08-12 00:00:00 with 2051.0 cases
For Cases > 5000,the first exceedance is on 174 index, the corresponding date is 2014-09-12 00:00:00 with 5092.5 cases

                Exceedance Deaths

For Deaths > 100,the first exceedance is on 15 index, the corresponding date is 2014-04-06 00:00:00 with 102.16666666666667 num
bers
For Deaths > 500,the first exceedance is on 105 index, the corresponding date is 2014-07-05 00:00:00 with 508.75 numbers
For Deaths > 1000,the first exceedance is on 140 index, the corresponding date is 2014-08-09 00:00:00 with 1013.0 numbers
For Deaths > 2000,the first exceedance is on 165 index, the corresponding date is 2014-09-03 00:00:00 with 2089.0 numbers
For Deaths > 5000,the first exceedance is on 216 index, the corresponding date is 2014-10-24 00:00:00 with 5026.0 numbers
```

Figure 4 Cases& Death exceed certain value

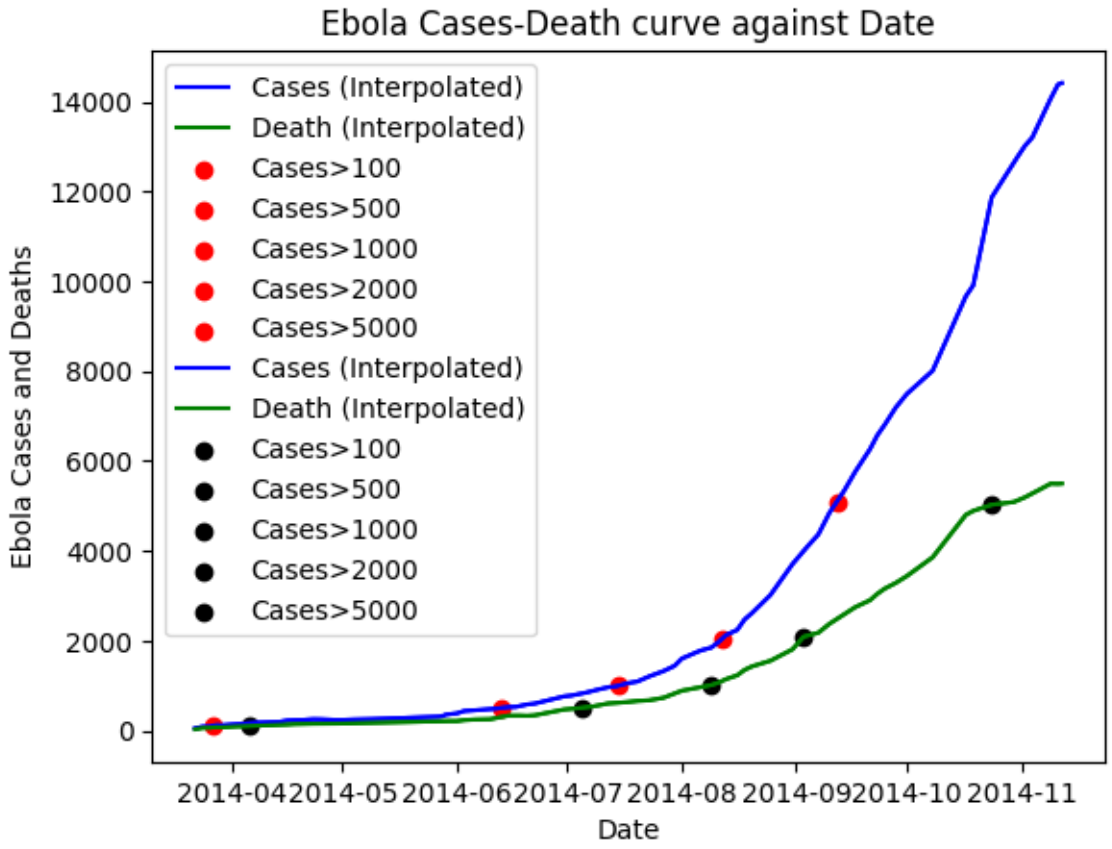


Figure 5 Ebola Cases- Death curve Against Date

To wrap up this question of dealing with Ebola data and using date- range to find the missing dates, approach of interpolation to estimate the missing, and plotting code has improved my knowledge in data visualization and eager to produce the required result. As shown in different figure and in code all code run and I showed with comment and tables every step I reach on, which means that this is what run only but in code you can print every case to know what it displays.

Q7. Question number seven uses data from question number 6, means is to continue from with data I used, as I used Jupyter notebook it does not require to read the data file again but what I did I called the dataframe which contain data to check if are ready to be used and as I was tasked to find the average growth rate per day of Ebola number of cases and deaths. By approaching I used build function “pct change” to calculate the percentage growth rate on Cases and Deaths, then after I added two columns that show Cases and Deaths rate in dataframe. As task was to find the average growth rate, I used the percentage growth rate of cases and other of deaths to find Average for each using built in function mean to find the mean of cases and deaths, as shown in figure 6.

Representing Percentage Growth rate for Death and Cases

	Date	Cases	Death	Diff	Noofdays	Cases_rate	Death_rate
0	2014-03-22	49.0	29.0	2.0	1.0	NaN	NaN
1	2014-03-23	67.5	44.0	NaN	NaN	37.755102	51.724138
2	2014-03-24	86.0	59.0	1.0	3.0	27.407407	34.090909
3	2014-03-25	86.0	60.0	1.0	4.0	0.000000	1.694915
4	2014-03-26	86.0	62.0	1.0	5.0	0.000000	3.333333
..	...	...	...	...	...	...	...
231	2014-11-08	13894.4	5451.8	NaN	NaN	1.265232	0.817368
232	2014-11-09	14068.0	5496.0	2.0	233.0	1.249424	0.810741
233	2014-11-10	14225.5	5494.0	NaN	NaN	1.119562	-0.036390
234	2014-11-11	14383.0	5492.0	1.0	235.0	1.107167	-0.036403
235	2014-11-12	14413.0	5498.0	-41955.0	236.0	0.208580	0.109250

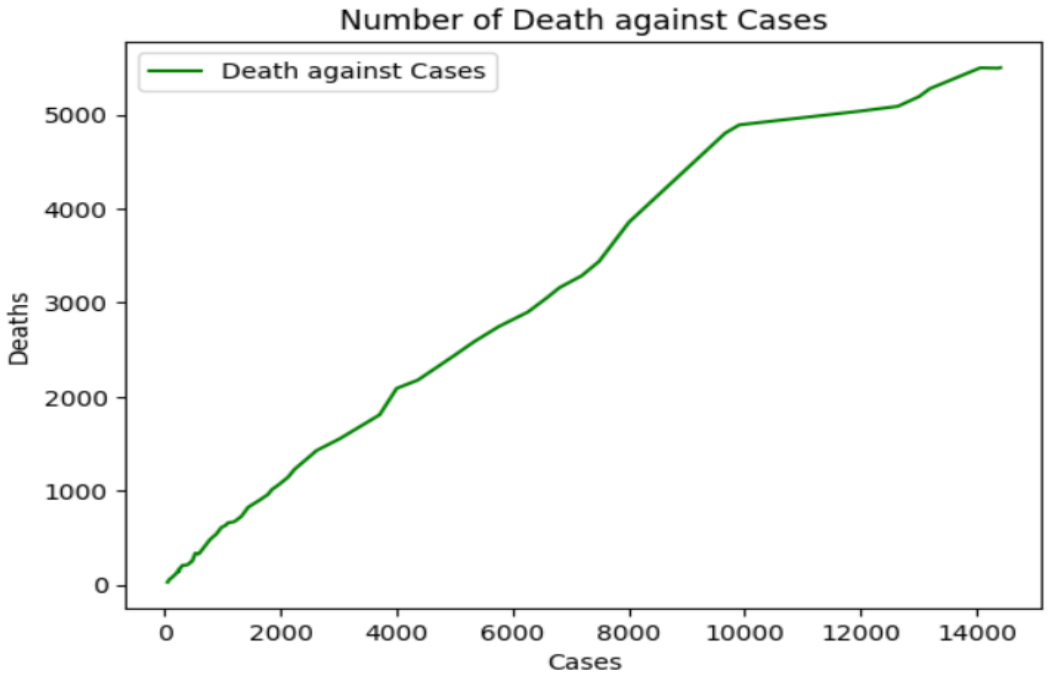
[236 rows x 7 columns]
The mean value of Cases Growth rate is 2.506521891649996

The mean value of Death Grow rate is 2.330608083367968

Figure 6 Mean {Cases, Deaths} Growth rate

Q8. By continuing with Question 6 data I was tasked to plot the graph of number of deaths versus the number of Cases. Using matplotlib I plotted the graph where Cases are on x axis and on y axis is Deaths the result shown in figure shows the relationship between number of cases to the deaths of Ebola. Next, I used the dataset of Deaths and Cases I calculated the ratio of death to Cases. Moreover, I used built in function mean to determine the average ratio of Ebola deaths to Cases and the result is shown in Jupyter notebook and in the figure 7 below the graph.

```
Using Data from Question 6
```




```
Computing Average Ratio

The Average ratio of Ebola Deaths to Cases is 0.5577992908998353
```

Figure 7 Death vs Cases and Average ratio below

Q9. Given two data set CSV file for ETF called 'SPY' and 'TLT' and asked to plot two time series. By using pandas, I read the two files in Jupyter notebook and named the variable that store them different like df_0 for SPY data and df_1 for TLT data, then I extracted the Dates and adjusted closing value from the dataframe and printed as instructed for both data file SPY and for TLT. Additionally, I created function to normalize the adjusted closing stock by 100 where I assigned to normalize starting from index zero of Adjusted close for both TLT and SPY data to set them to common standard format to help in improving the accuracy of the data analysed. After setting common factor I plotted both stock on the same graph as shown in figure below. Due to date are many and cannot fit on the sheet I used matplotlib module plt. xticks to adjust the dates in the range it can be visible and readable. As shown in figure 8 the effect of normalizing by 100 is that it provides the common point to start on.

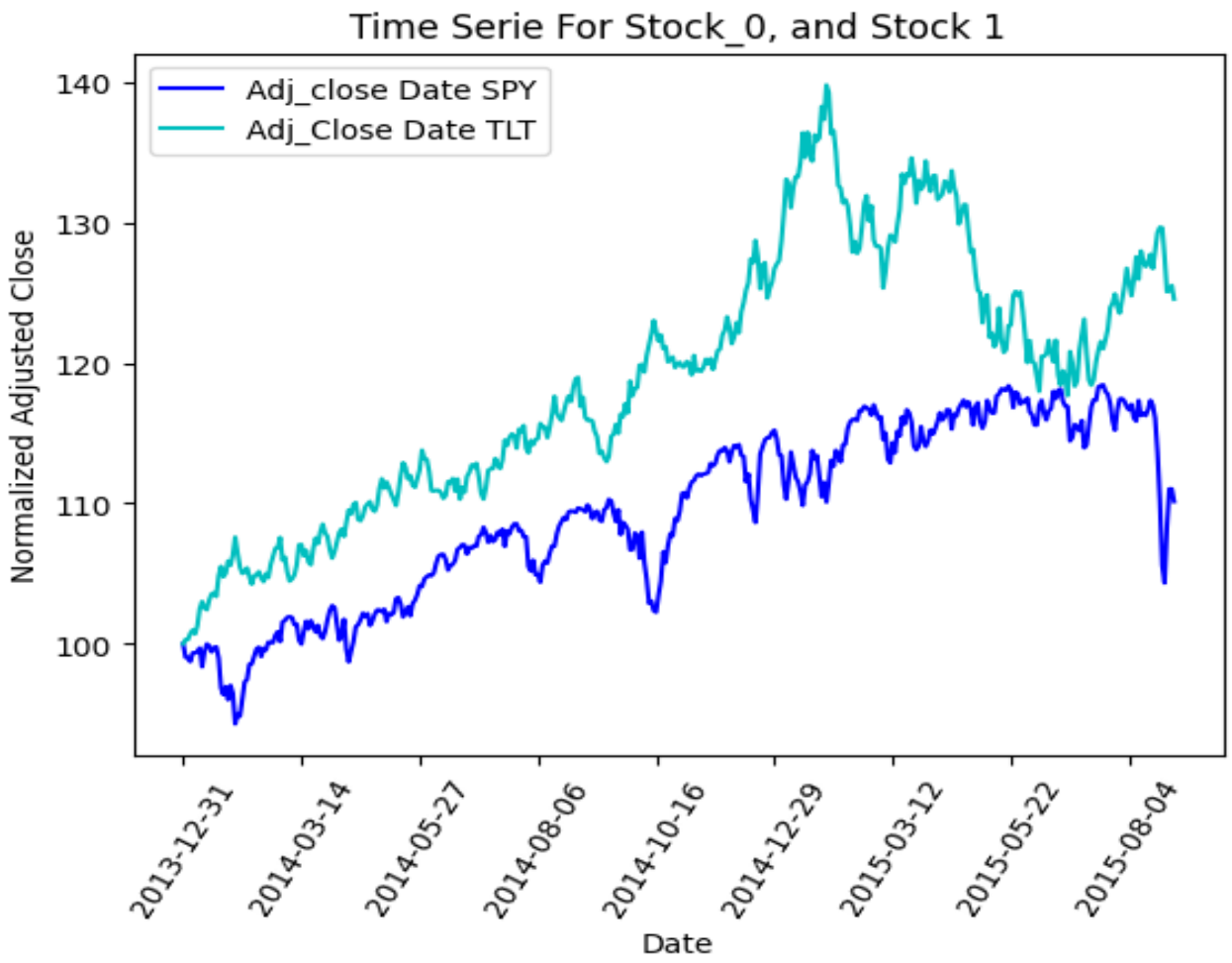


Figure 8 Time series for TLT and SPY stocks

Q10. Continuing with Number 9's data I was asked to determine daily return for each trading days and calculating average, min, and max daily return for both dataframe (SPY and TLT) expressed in percentage. I started by calling dataframe of SPY df_0 and print them to check if data are still valid, then I calculated daily percentage return using built in function that calculate daily return and multiply by hundred percent, the result is displayed in the data set of SPY. And I did the same for TLT data by first return dataframe given in question number 9 and use built in function to determine the daily percentage return. After that I created the code to find the Average, minimum and maximum ofc data set. As Python a very rich interpreters it has built in function for Average(mean), Min, and Max. I used the function to find the summary statistic of each dataset and the result is shown in the figure below.

```
The summary statistic of SPY dataSet

The minimum percentage of SPY Adj Close:-4.210696226671873%
The maximum percentage of SPY Adj Close:3.839356573503383 %
The Average percentage of SPY Adj Close: 0.026256688949696263%
```

Figure 9 SPY Summary Statistic Result

```
The summary statistic of TLT dataSet

The minimum percentage of SPY Adj Close:-2.4324720378358644%
The maximum percentage of SPY Adj Close:2.6468578136499055 %
The Average percentage of SPY Adj Close: 0.05597260696112029%
```

Figure 10 TLT Summary statistic Result

In conclusion this assignment wakes up call in data analysis which indeed has acquired me with different skills of approaching questions and analysing them using programming language. I answered all the questions and provided with clear evidence for easy to read, where I used Jupyter notebook.

Reference

[1] “Matplotlib — Visualization with Python.” https://matplotlib.org/ (accessed Sep. 02, 2023).

[2] Zach, “The Easiest Way to Use Pandas in Python: import pandas as pd,” *Statology*, May 31, 2021. https://www.statology.org/import-pandas-as-pd/ (accessed Sep. 02, 2023).

[3] “Pandas date_range() function - w3resource.” https://www.w3resource.com/pandas/date_range.php (accessed Sep. 02, 2023).

[4] Zach, “The Easiest Way to Use Pandas in Python: import pandas as pd,” *Statology*, May 31, 2021. https://www.statology.org/import-pandas-as-pd/ (accessed Sep. 02, 2023).