Project Report

Probability and Statistics

Dr. Maha Amin

Stock Guide

Team Members		
Name	Section	BN
Abdelrahman Hamdy Ahmed	1	36
Abdelrahman Noaman Loqman	2	2
Abdelaziz Salah Mohammed	2	3
Kirollos Samy Hakim	2	12
Khaled Hesham Sayed	1	22
Ahmed Mostafa Mohammed	1	10

12/30/2021

Abstract

Our main topic is about investigating the stability of stock exchange in 5 different companies every month starting from the beginning of the month and the price it closed on. We collected a sample of data from each of these companies and applied descriptive statistics to determine some measures of spread including the sample mean and the standard deviation where this sample mean describes the profit earned and the standard deviation represents the risk to be measured. Proceeding to the main hypothesis of our project, our most important aspect is the monthly return on the closing stock price which is measured by observing the difference between the current month and the previous one. This is our main random variable to know how stable the company is; therefore, our main question is the following: Is it safe to buy stocks from this company or not? This is determined by applying forecasting to predict the future monthly return on stock price.

Background

A stock is a financial instrument that represents a proportionate claim on a company's assets (what it owns) and earnings (what it generates in profits). Stocks are also known as shares or equity in a firm. Stock ownership entails owning a piece of the company equal to the number of shares held as a percentage of the total number of outstanding shares. An individual or entity who holds 100,000 shares of a business with one million outstanding shares, for example, owns 10% of the company. The majority of firms have outstanding shares in the millions or billions of dollars. Stocks are also known as shares or equity in a firm. Existing shareholders can deal with potential buyers on stock exchanges, which are secondary markets. It's vital to remember that companies that trade on stock exchanges don't buy and sell their own stock on a regular basis. Companies may buy back stock or issue new shares, but these are not day-to-day operations and frequently take place outside of an exchange's structure.

Problem Definition

The problem we are trying to solve is mainly encouraging shareholders to determine whether their business is moving in the right direction and earning reasonable profit from the closing stock prices which will help them in the decision-making process and predict the company’s future situation. On the other hand, people who buy these stocks will make sure that the company they are dealing with is in a stable state in terms of the monthly return, our random variable. Such confusion in whether to buy stocks or not describes our problem making it very important for both sides (investors and owners) and interesting for them at the same time. Analyzing the results claims that calculating the monthly return is not as easy as it looks and challenging to solve since it requires prior knowledge of statistical analysis including the hypothesis we are aiming to conduct which will be described later and the risk we need to calculate.

Methods of Solution

Our analysis is purely implemented in python with help of excel sheets, because we found out that it is easy to analyze the data of different samples at the same time and represent it in an efficient way using its libraries.

✓ **The main approach was establishing a hybrid between python and excel to analyze the data and perform our concepts mentioned below**

Models used:

- NumPy: For certain calculations in statistics
- Panda: To read the csv files and store them in a data frame which represents our sample of each company. We used it to find our random variable (the daily return) from the .pct_change() function. Also, it is used in graphing the percentage change of the daily return each day and finding the mean and standard deviation of each sample to be used for our hypothesis test.
- Matplot: We used this specific library to help in plotting the graphs
- Datetime: Show the date and time in a certain format
- Seaborn: For plotting the normal distribution curve specifying the bins and style of every graph

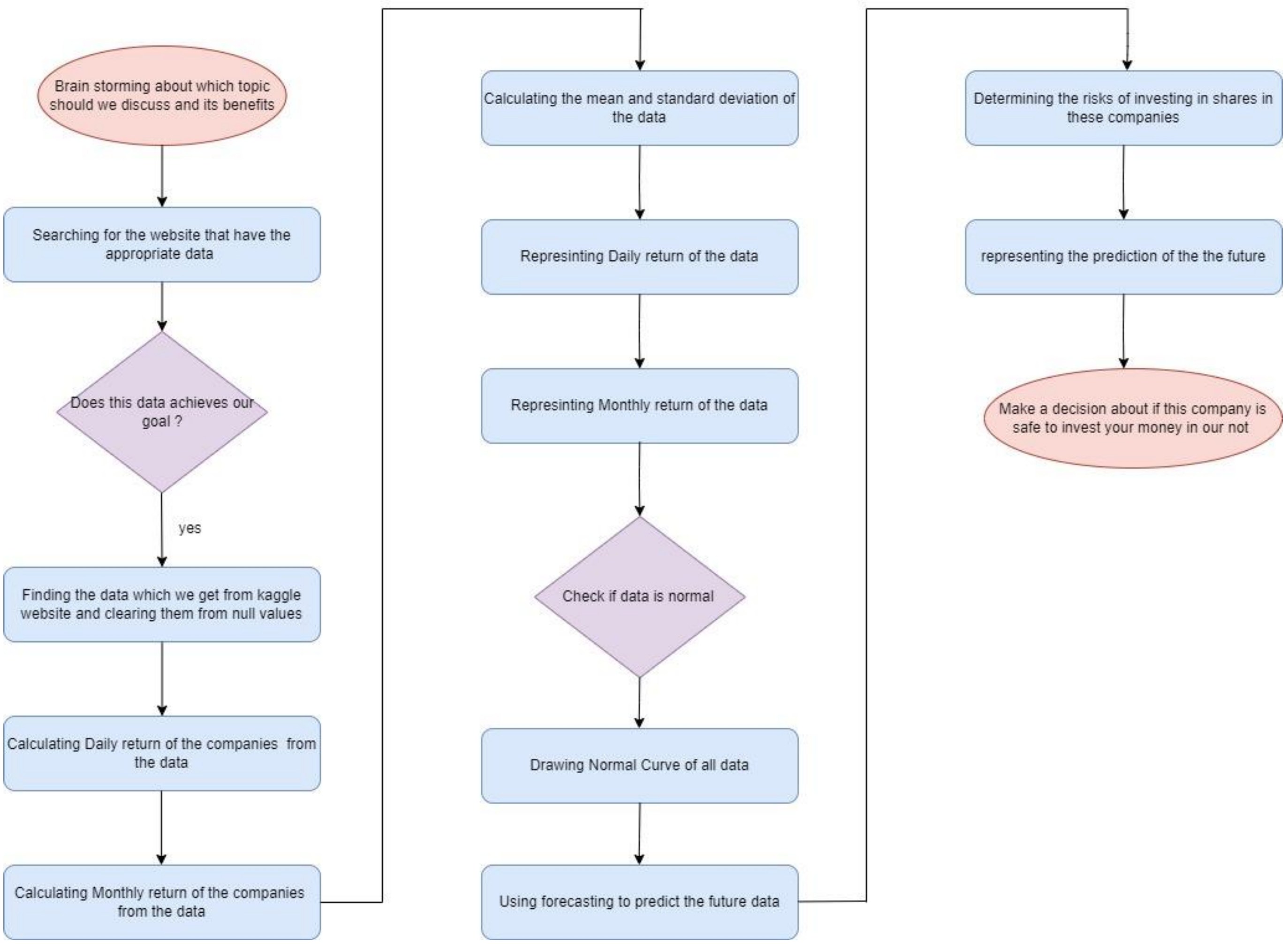
Excel:

We used Microsoft Excel to view our csv files which contains the historical data of the 5 companies along a duration of 5 years as a representation of each set of data containing the opening/closing stock price as well as the newly added columns (monthly and daily return).

Techniques done:

- Evaluating the mean and variance.
- Forecasting the future monthly return by using linear regression.
- Finding the standard deviation to predict the risk of each company.

Algorithm Flowchart


Brain storming about which topic should we discuss and its benefits
Searching for the website that have the appropriate data
Does this data achieves our goal ?
yes
Finding the data which we get from kaggle website and clearing them from null values
Calculating Daily return of the companies from the data
Calculating Monthly return of the companies from the data
Calculating the mean and standard deviation of the data
Represinting Daily return of the data
Represinting Monthly return of the data
Check if data is normal
Drawing Normal Curve of all data
Using forecasting to predict the future data
Determining the risks of investing in shares in these companies
representing the prediction of the the future
Make a decision about if this company is safe to invest your money in our not

Data Description

We have collected our data for the most 5 famous companies:

- Apple
- Facebook
- Google
- Amazon
- Tesla

The sample data consists of:

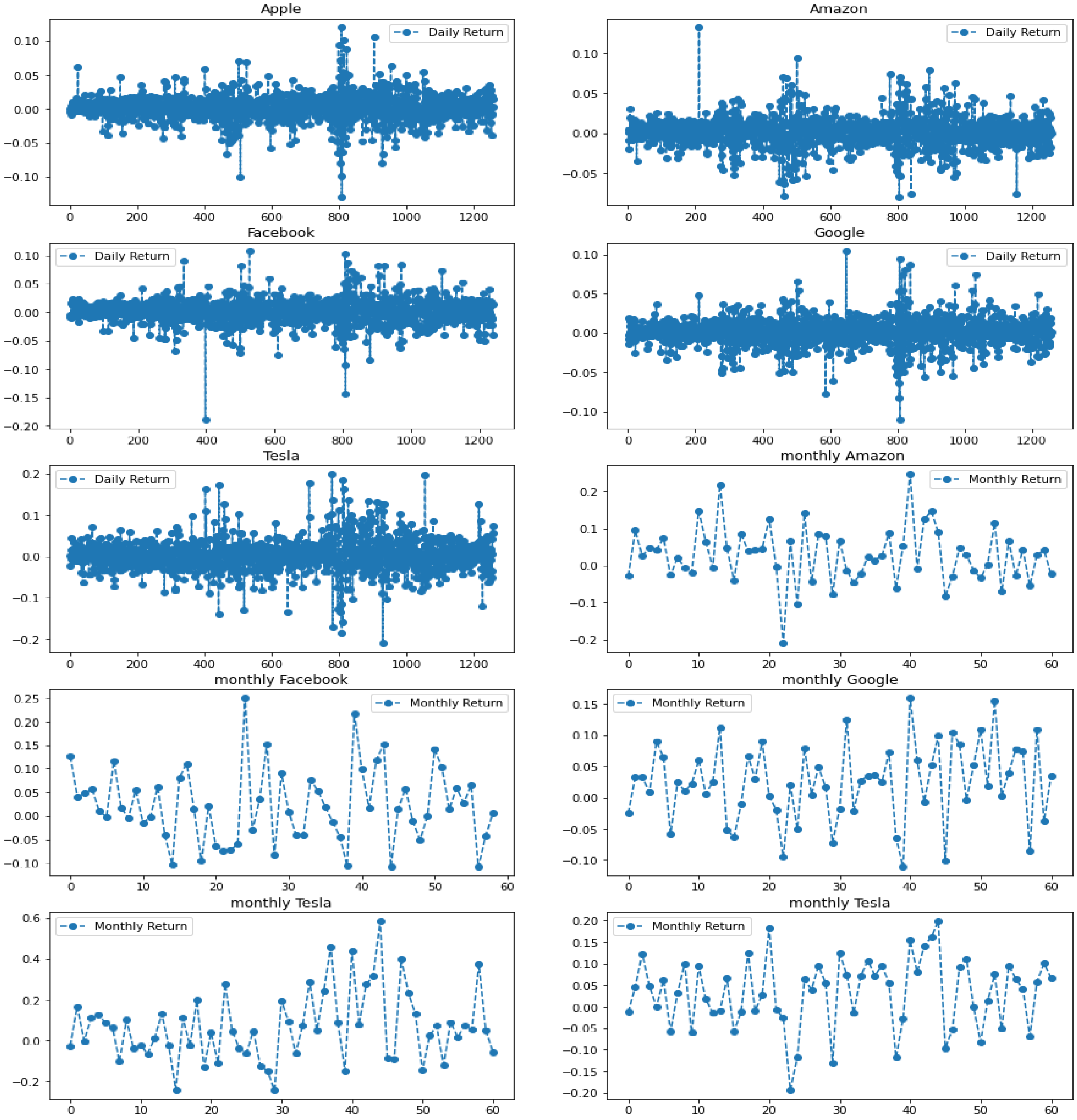
- The date in which we calculate the stock price starting from 2016 - 2021
- Opening stock price of the day
- Closing stock price of the day

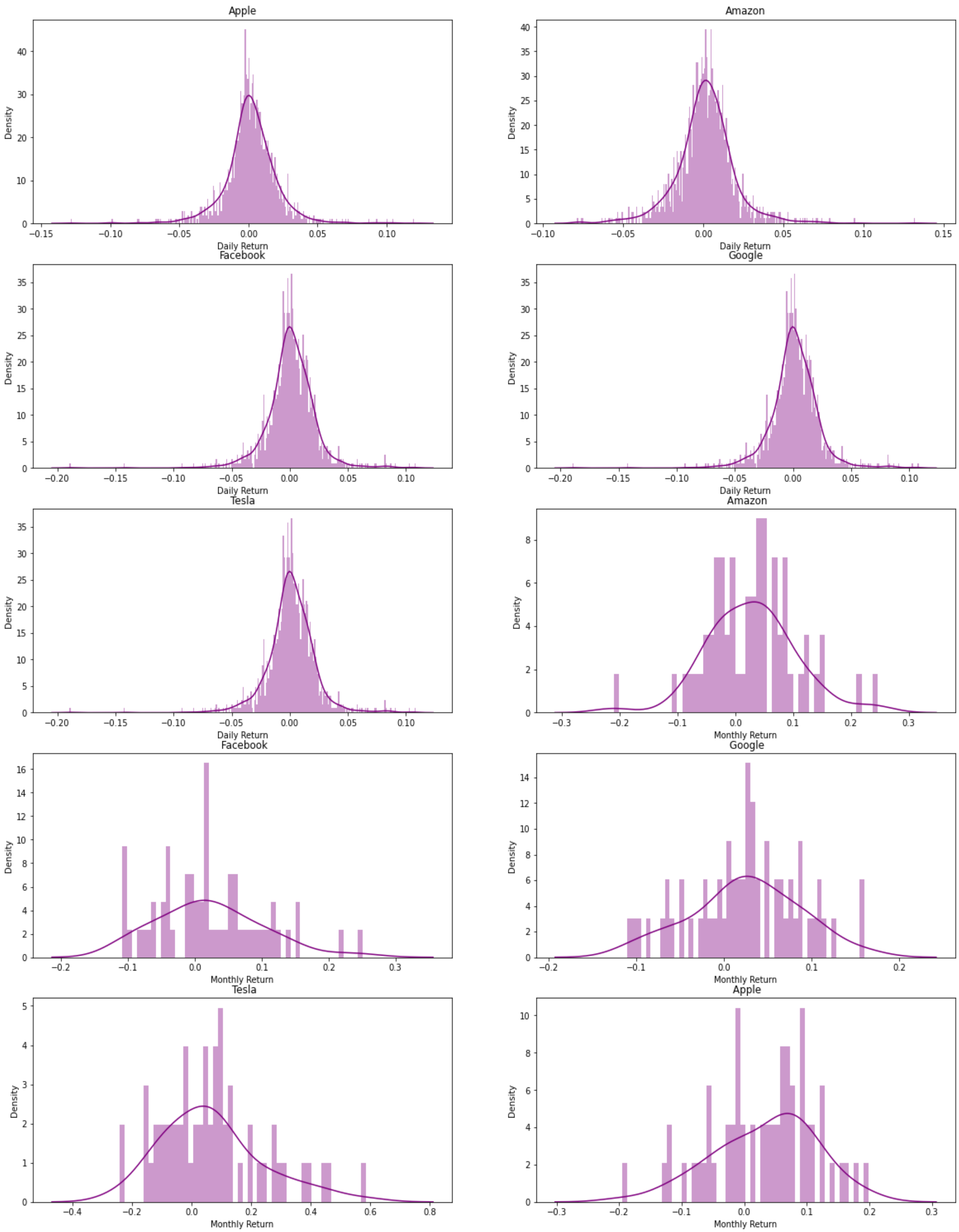
Our reference for this sample data was https://www.kaggle.com/

The following 2 pages graphically describe the data in detail:

- The daily return vs Frequency density
- Normal distribution curve

Using our methodology, we calculated the monthly return of each company and inserted it in our csv files as a new column to be used in the code for several reasons such as graphical representations. Numerically, the data obtained is randomized on the total population (increasing and decreasing).




Apple
Density
0
10
20
30
40
−0.15
−0.10
−0.05
0.00
0.05
0.10
Daily Return
Amazon
Density
0
5
10
15
20
25
30
35
40
−0.10
−0.05
0.00
0.05
0.10
0.15
Daily Return
Facebook
Density
0
5
10
15
20
25
30
35
−0.20
−0.15
−0.10
−0.05
0.00
0.05
0.10
Daily Return
Google
Density
0
5
10
15
20
25
30
35
−0.20
−0.15
−0.10
−0.05
0.00
0.05
0.10
Daily Return
Tesla
Density
0
5
10
15
20
25
30
35
−0.20
−0.15
−0.10
−0.05
0.00
0.05
0.10
Daily Return
Amazon
Density
0
2
4
6
8
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
Monthly Return
Facebook
Density
0
2
4
6
8
10
12
14
16
−0.2
−0.1
0.0
0.1
0.2
0.3
Monthly Return
Google
Density
0
2
4
6
8
10
12
14
−0.2
−0.1
0.0
0.1
0.2
Monthly Return
Tesla
Density
0
1
2
3
4
5
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
Monthly Return
Apple
Density
0
2
4
6
8
10
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
Monthly Return

Analysis of the results

To analyze the data of each company, we need to:

- Calculate the sample mean of every monthly change between the closing price this month and the previous one.
- Get the standard deviation of our sample data (s).
- Determining the degree of confidence.
- Set our null hypothesis (To be compared with the result later)
- Find the confidence interval which states an upper bound and lower bound of the sample mean
- Try to prove the alternative hypothesis of a certain daily return.
- Make a decision on whether we accept the claim or reject the alternative hypothesis

According to the above analysis, we can achieve the original goal of our statistical research which is deciding the safety of buying a stock from this company with a certain price and an estimate daily return. Therefore, we were able to assess the performance of our algorithm.

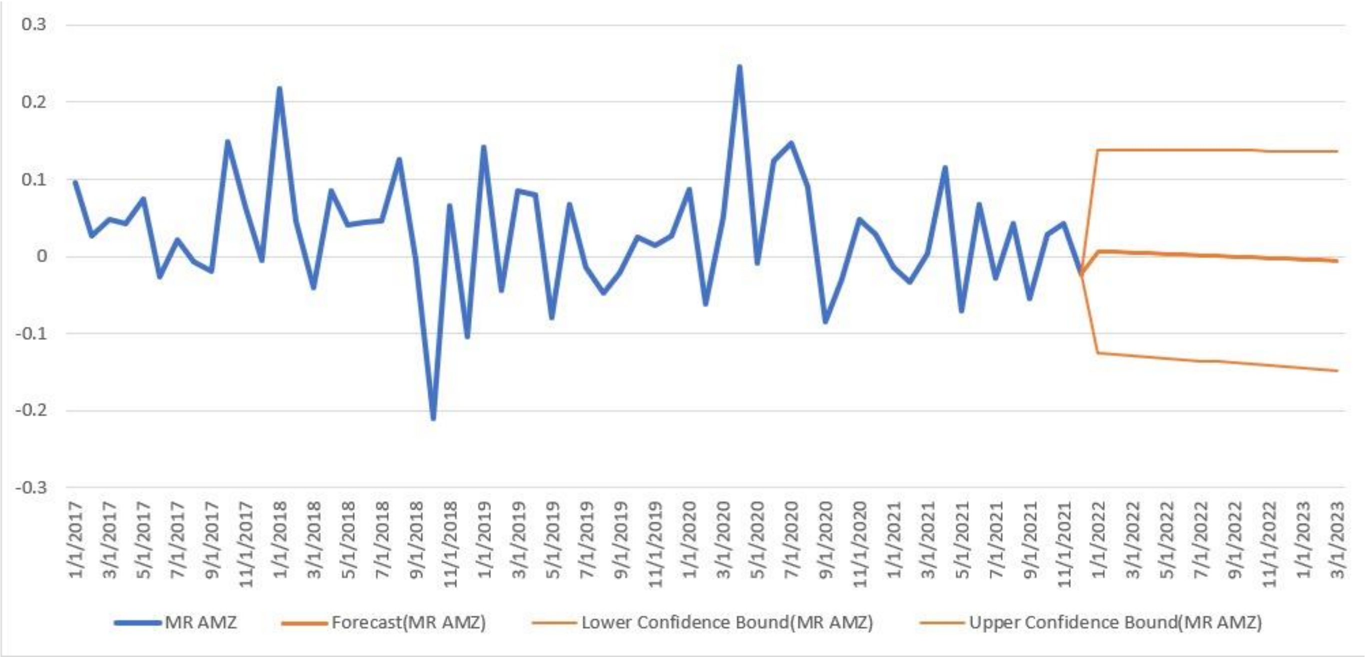
Forecasting and calculating the risk

- ✓ Forecasting is a strategy that uses previous data as inputs to make well-informed predictions about the direction of future trends. Forecasting is used by businesses to determine how to allocate their budgets or plan for anticipated expenses in the future. This is usually determined by the expected demand for the goods and services provided. Forecasting starts with management's knowledge and expertise. Organizations must understand the more complex elements of the various forecasting methodologies in order to gain the most numerous benefits from forecasts. Also, be aware of what an appropriate forecasting method type can and cannot achieve, as well as which forecast type is most suited to a certain requirement. We aimed to use forecasting by linear regression to predict the future monthly return for the 5 companies over a period of 6 months. As a result of applying linear regression on scatter plotted points generated from the given data to obtain a linear equation in order to substitute in it with any value so that we can predict the future points.

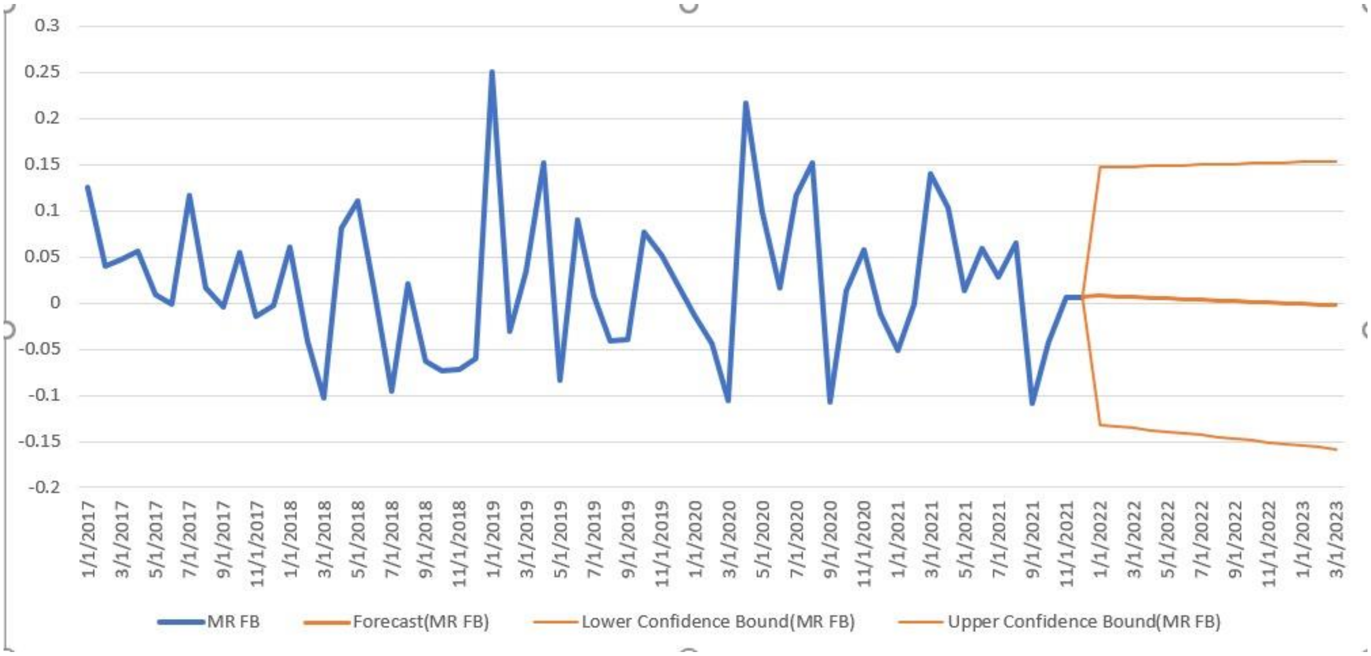
- ✓ The risk bound of every company is obtained from calculating the standard deviation over the 5 years which acts as a limit of safety, represented as a number. If we go below it, then the company is less likely to lose and vice versa if it goes above the risk

Forecasting graphs:

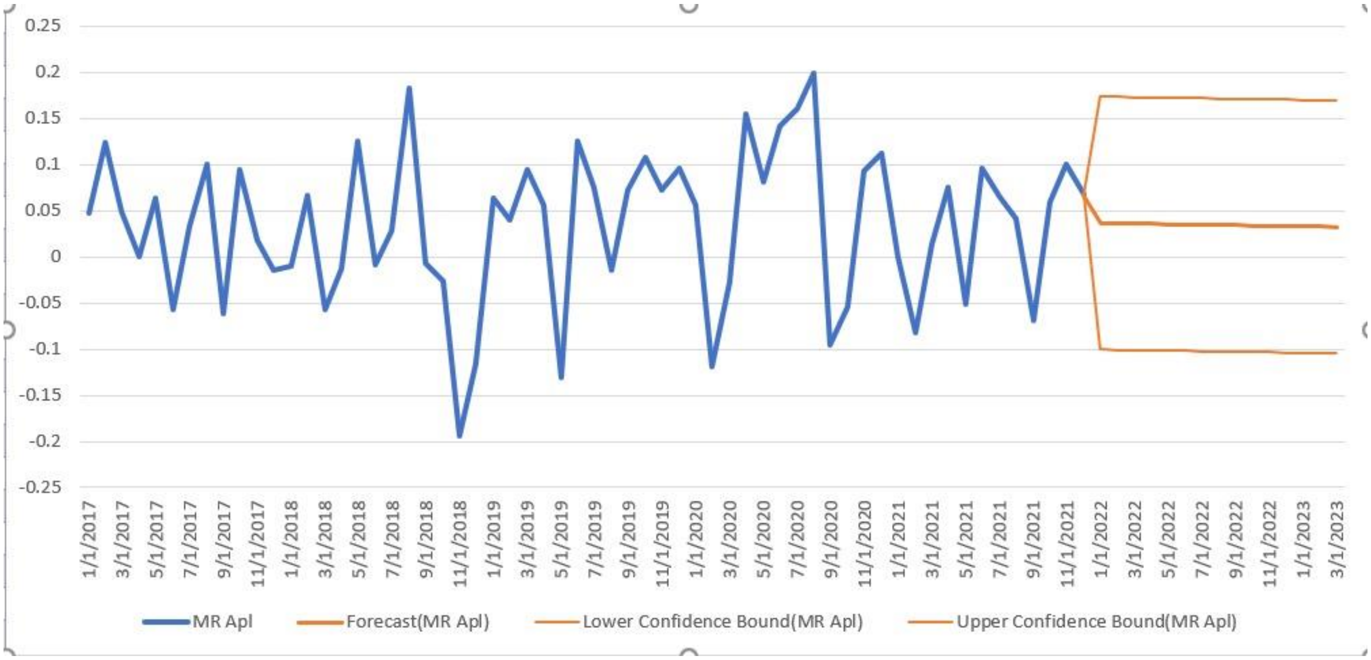
AMAZON



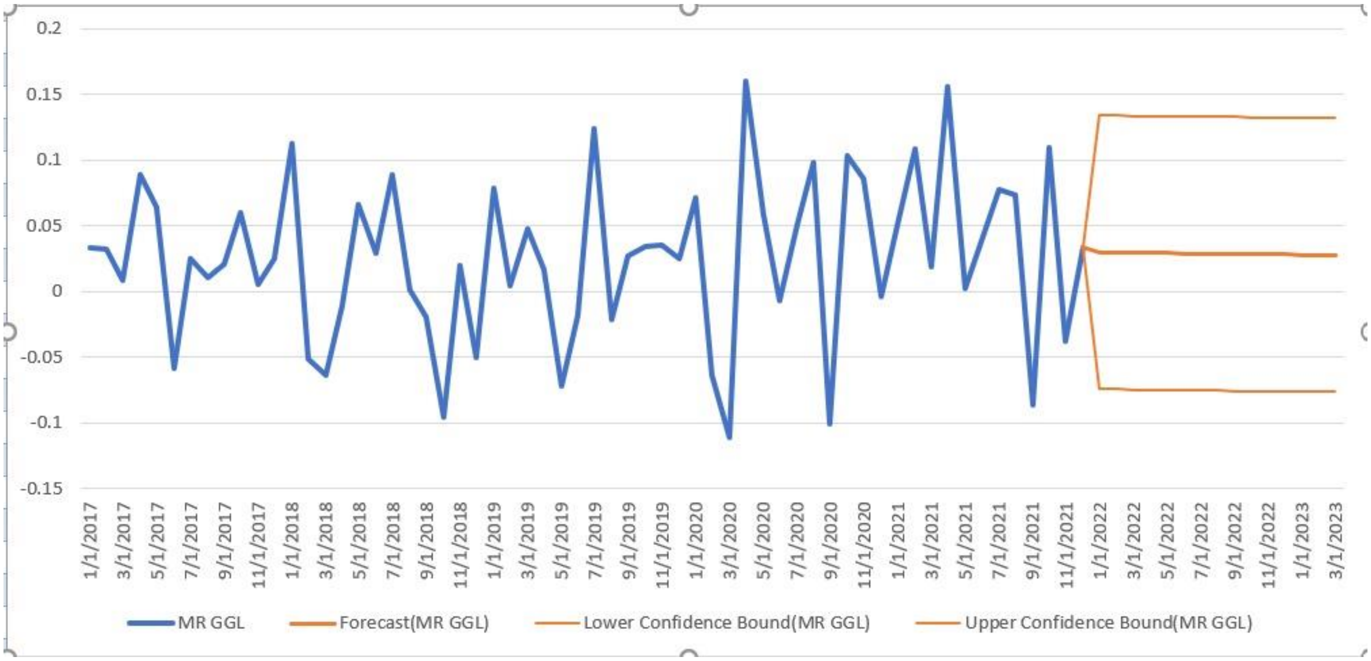
FACEBOOK



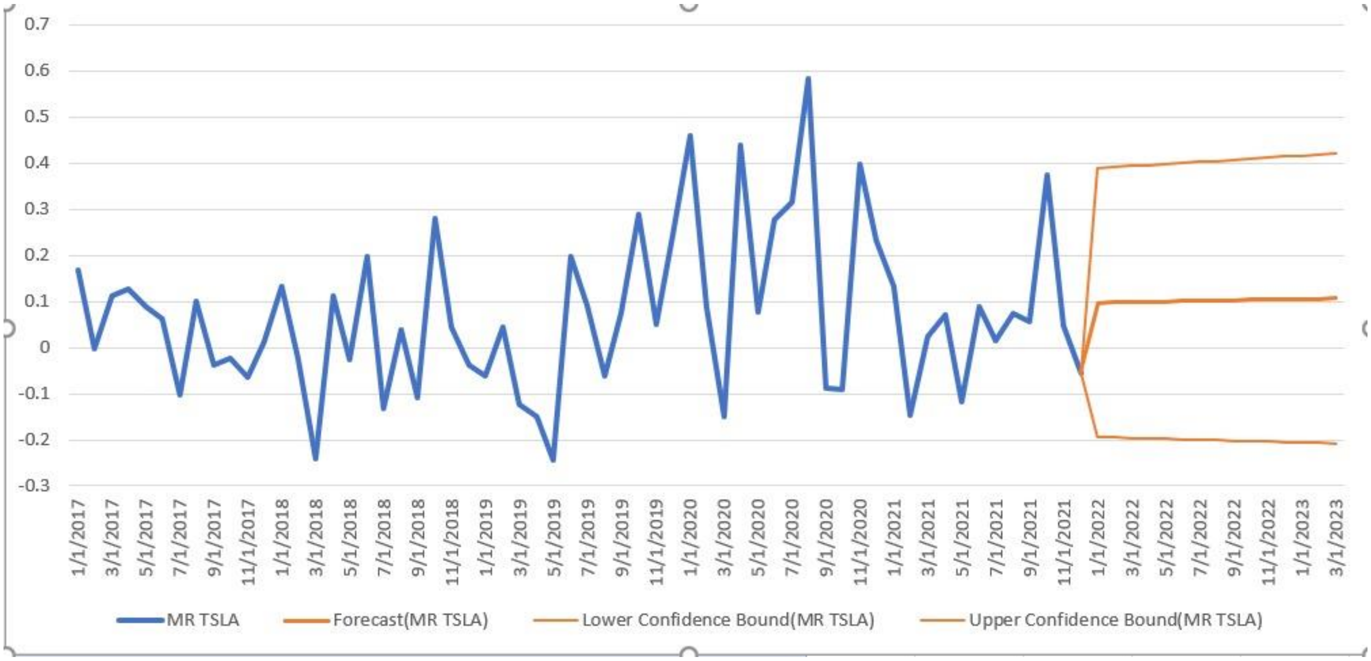
APPLE



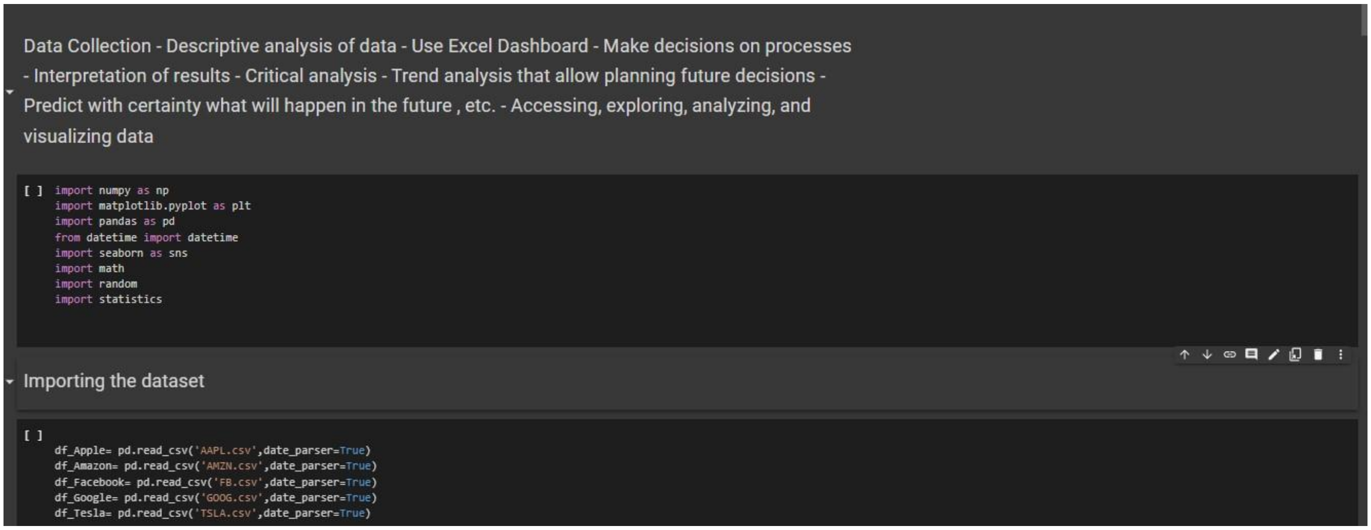
GOOGLE



TESLA



Appendix

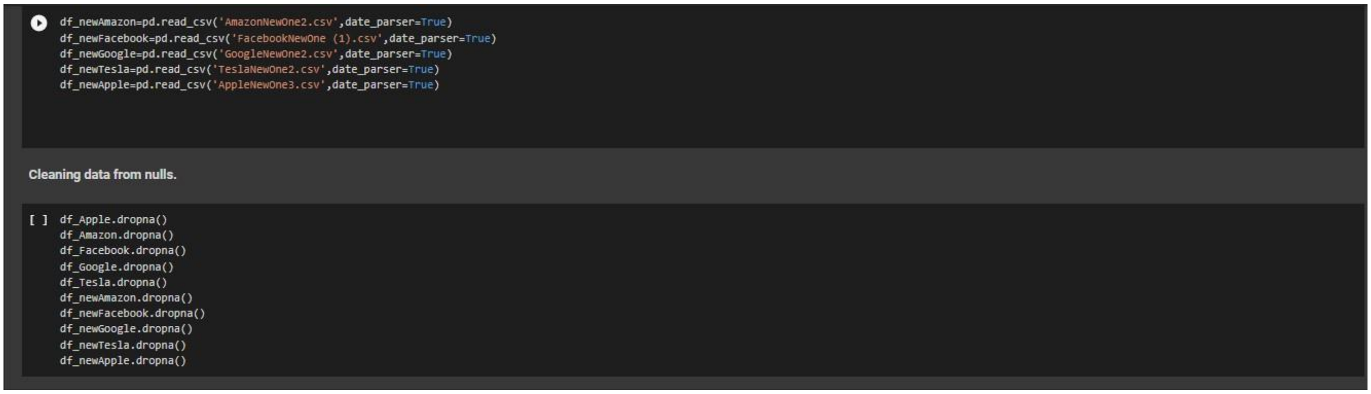


Data Collection - Descriptive analysis of data - Use Excel Dashboard - Make decisions on processes - Interpretation of results - Critical analysis - Trend analysis that allow planning future decisions - Predict with certainty what will happen in the future , etc. - Accessing, exploring, analyzing, and visualizing data

```
[ ] import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd
    from datetime import datetime
    import seaborn as sns
    import math
    import random
    import statistics
```

Importing the dataset

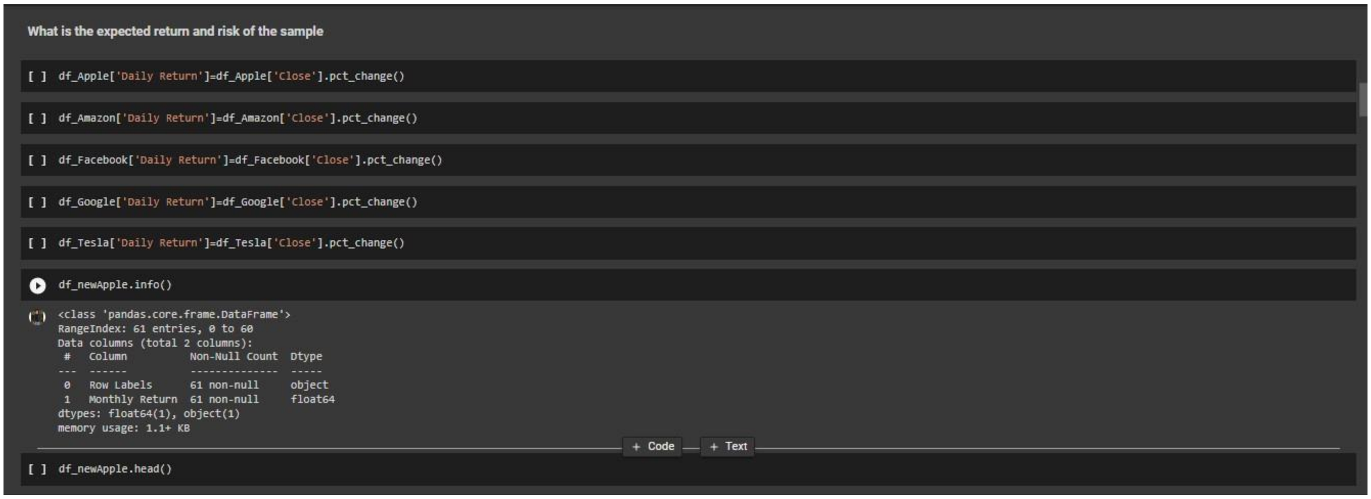
```
[ ]
    df_Apple= pd.read_csv('AAPL.csv',date_parser=True)
    df_Amazon= pd.read_csv('AMZN.csv',date_parser=True)
    df_Facebook= pd.read_csv('FB.csv',date_parser=True)
    df_Google= pd.read_csv('GOOG.csv',date_parser=True)
    df_Tesla= pd.read_csv('TSLA.csv',date_parser=True)
```



```
df_newAmazon=pd.read_csv('AmazonNewOne2.csv',date_parser=True)
df_newFacebook=pd.read_csv('FacebookNewOne (1).csv',date_parser=True)
df_newGoogle=pd.read_csv('GoogleNewOne2.csv',date_parser=True)
df_newTesla=pd.read_csv('TeslaNewOne2.csv',date_parser=True)
df_newApple=pd.read_csv('AppleNewOne3.csv',date_parser=True)
```

Cleaning data from nulls.

```
[ ] df_Apple.dropna()
    df_Amazon.dropna()
    df_Facebook.dropna()
    df_Google.dropna()
    df_Tesla.dropna()
    df_newAmazon.dropna()
    df_newFacebook.dropna()
    df_newGoogle.dropna()
    df_newTesla.dropna()
    df_newApple.dropna()
```



What is the expected return and risk of the sample

```
[ ] df_Apple['Daily Return']=df_Apple['Close'].pct_change()
```

```
[ ] df_Amazon['Daily Return']=df_Amazon['Close'].pct_change()
```

```
[ ] df_Facebook['Daily Return']=df_Facebook['Close'].pct_change()
```

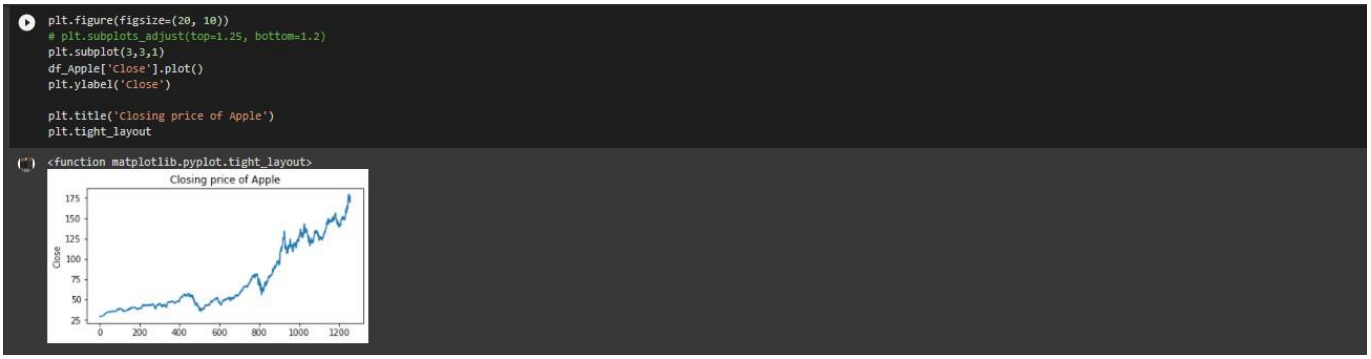
```
[ ] df_Google['Daily Return']=df_Google['Close'].pct_change()
```

```
[ ] df_Tesla['Daily Return']=df_Tesla['Close'].pct_change()
```

```
df_newApple.info()
```

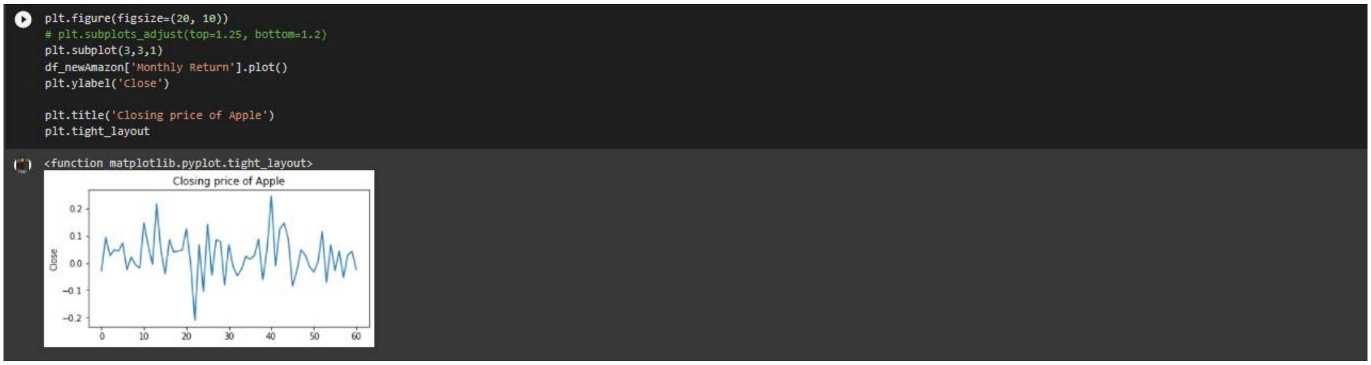
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 61 entries, 0 to 60
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Row Labels      61 non-null     object
 1   Monthly Return  61 non-null     float64
dtypes: float64(1), object(1)
memory usage: 1.1+ KB
```

```
[ ] df_newApple.head()
```



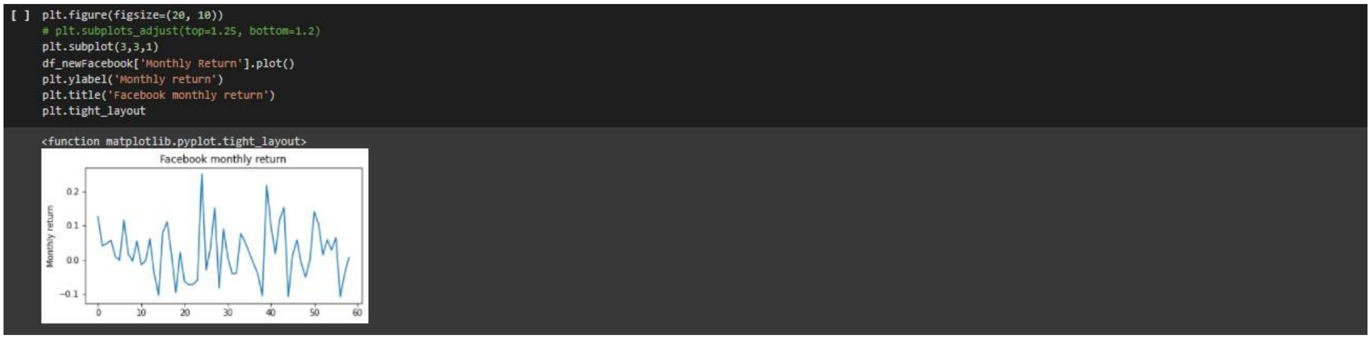
```
plt.figure(figsize=(20, 10))
# plt.subplots_adjust(top=1.25, bottom=1.2)
plt.subplot(3,3,1)
df_Apple['Close'].plot()
plt.ylabel('Close')

plt.title('Closing price of Apple')
plt.tight_layout
```

```
<function matplotlib.pyplot.tight_layout>
```



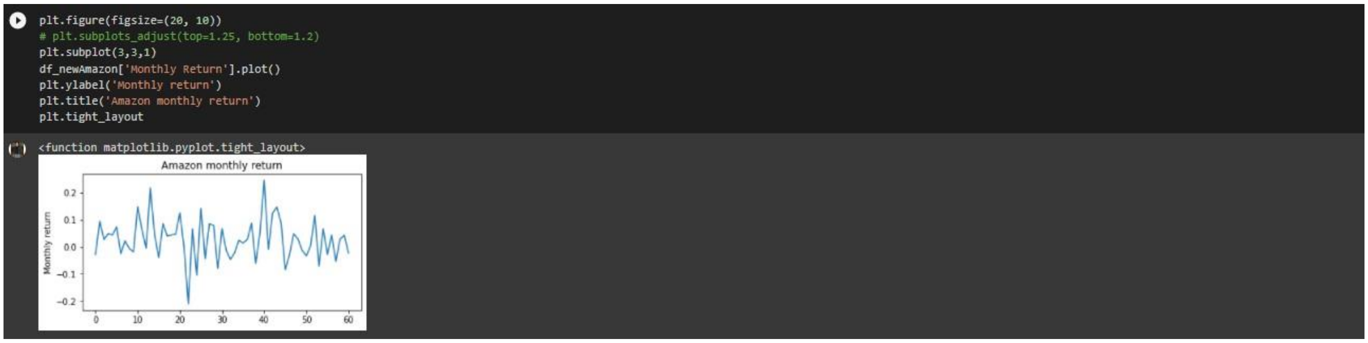
```
plt.figure(figsize=(20, 10))
# plt.subplots_adjust(top=1.25, bottom=1.2)
plt.subplot(3,3,1)
df_newAmazon['Monthly Return'].plot()
plt.ylabel('Close')

plt.title('Closing price of Apple')
plt.tight_layout
```

```
<function matplotlib.pyplot.tight_layout>
```



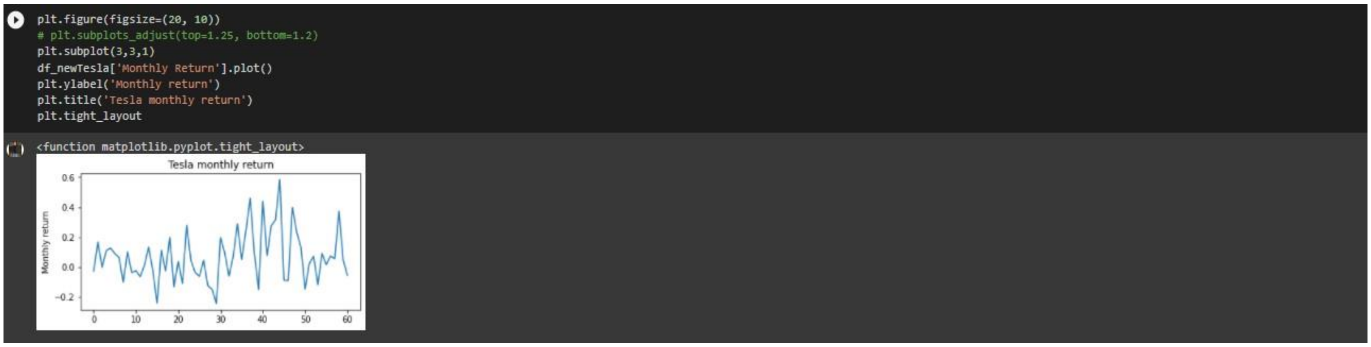
```
[ ] plt.figure(figsize=(20, 10))
    # plt.subplots_adjust(top=1.25, bottom=1.2)
    plt.subplot(3,3,1)
    df_newFacebook['Monthly Return'].plot()
    plt.ylabel('Monthly return')
    plt.title('Facebook monthly return')
    plt.tight_layout
```

```
<function matplotlib.pyplot.tight_layout>
```



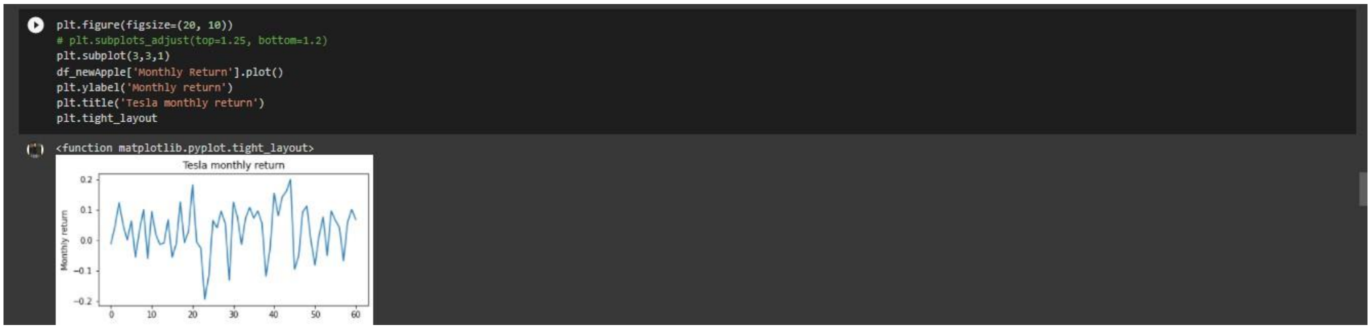
```
plt.figure(figsize=(20, 10))
# plt.subplots_adjust(top=1.25, bottom=1.2)
plt.subplot(3,3,1)
df_newAmazon['Monthly Return'].plot()
plt.ylabel('Monthly return')
plt.title('Amazon monthly return')
plt.tight_layout
```

```
<function matplotlib.pyplot.tight_layout>
```



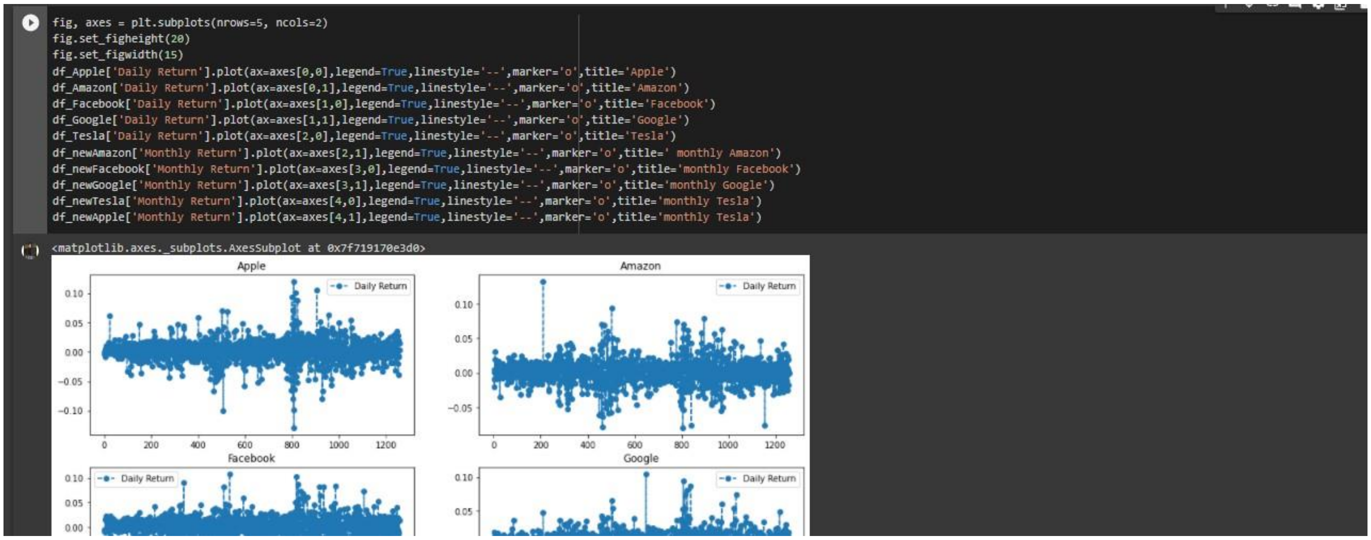
```
plt.figure(figsize=(20, 10))
# plt.subplots_adjust(top=1.25, bottom=1.2)
plt.subplot(3,3,1)
df_newTesla['Monthly Return'].plot()
plt.ylabel('Monthly return')
plt.title('Tesla monthly return')
plt.tight_layout
```

```
<function matplotlib.pyplot.tight_layout>
```



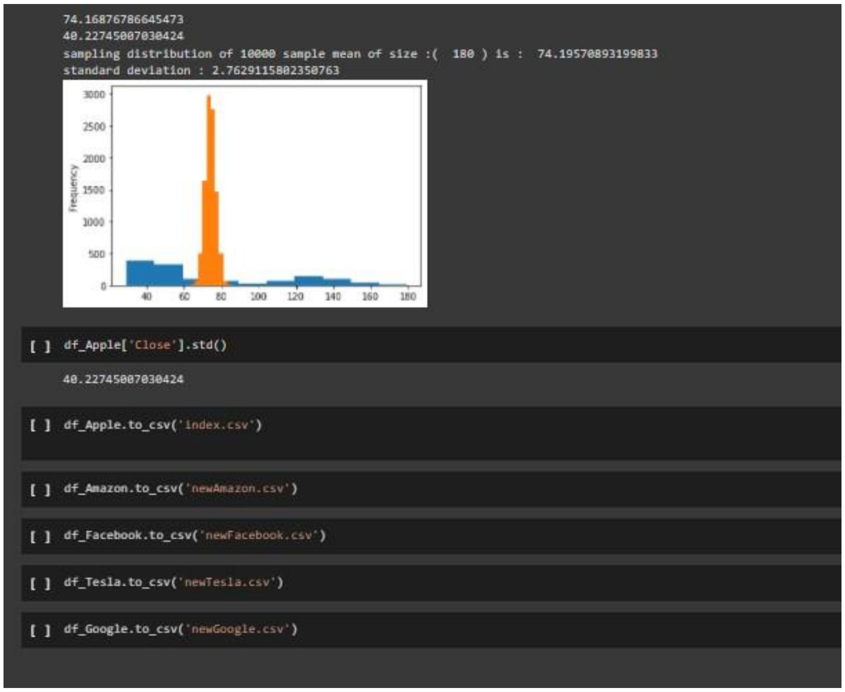
```
plt.figure(figsize=(20, 10))
# plt.subplots_adjust(top=1.25, bottom=1.2)
plt.subplot(3,3,1)
df_newApple['Monthly Return'].plot()
plt.ylabel('Monthly return')
plt.title('Tesla monthly return')
plt.tight_layout
```

```
<function matplotlib.pyplot.tight_layout>
```



```
fig, axes = plt.subplots(nrows=5, ncols=2)
fig.set_figheight(20)
fig.set_figwidth(15)
df_Apple['Daily Return'].plot(ax=axes[0,0],legend=True,linestyle='--',marker='o',title='Apple')
df_Amazon['Daily Return'].plot(ax=axes[0,1],legend=True,linestyle='--',marker='o',title='Amazon')
df_Facebook['Daily Return'].plot(ax=axes[1,0],legend=True,linestyle='--',marker='o',title='Facebook')
df_Google['Daily Return'].plot(ax=axes[1,1],legend=True,linestyle='--',marker='o',title='Google')
df_Tesla['Daily Return'].plot(ax=axes[2,0],legend=True,linestyle='--',marker='o',title='Tesla')
df_newAmazon['Monthly Return'].plot(ax=axes[2,1],legend=True,linestyle='--',marker='o',title=' monthly Amazon')
df_newFacebook['Monthly Return'].plot(ax=axes[3,0],legend=True,linestyle='--',marker='o',title='monthly Facebook')
df_newGoogle['Monthly Return'].plot(ax=axes[3,1],legend=True,linestyle='--',marker='o',title='monthly Google')
df_newTesla['Monthly Return'].plot(ax=axes[4,0],legend=True,linestyle='--',marker='o',title='monthly Tesla')
df_newApple['Monthly Return'].plot(ax=axes[4,1],legend=True,linestyle='--',marker='o',title='monthly Tesla')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f719170e3d0>
```



```
74.16876786645473
40.22745007030424
sampling distribution of 10000 sample mean of size :(  180 ) is :  74.19570893199833
standard deviation : 2.7629115802350763
```

```
[ ] df_Apple['Close'].std()
```

```
40.22745007030424
```

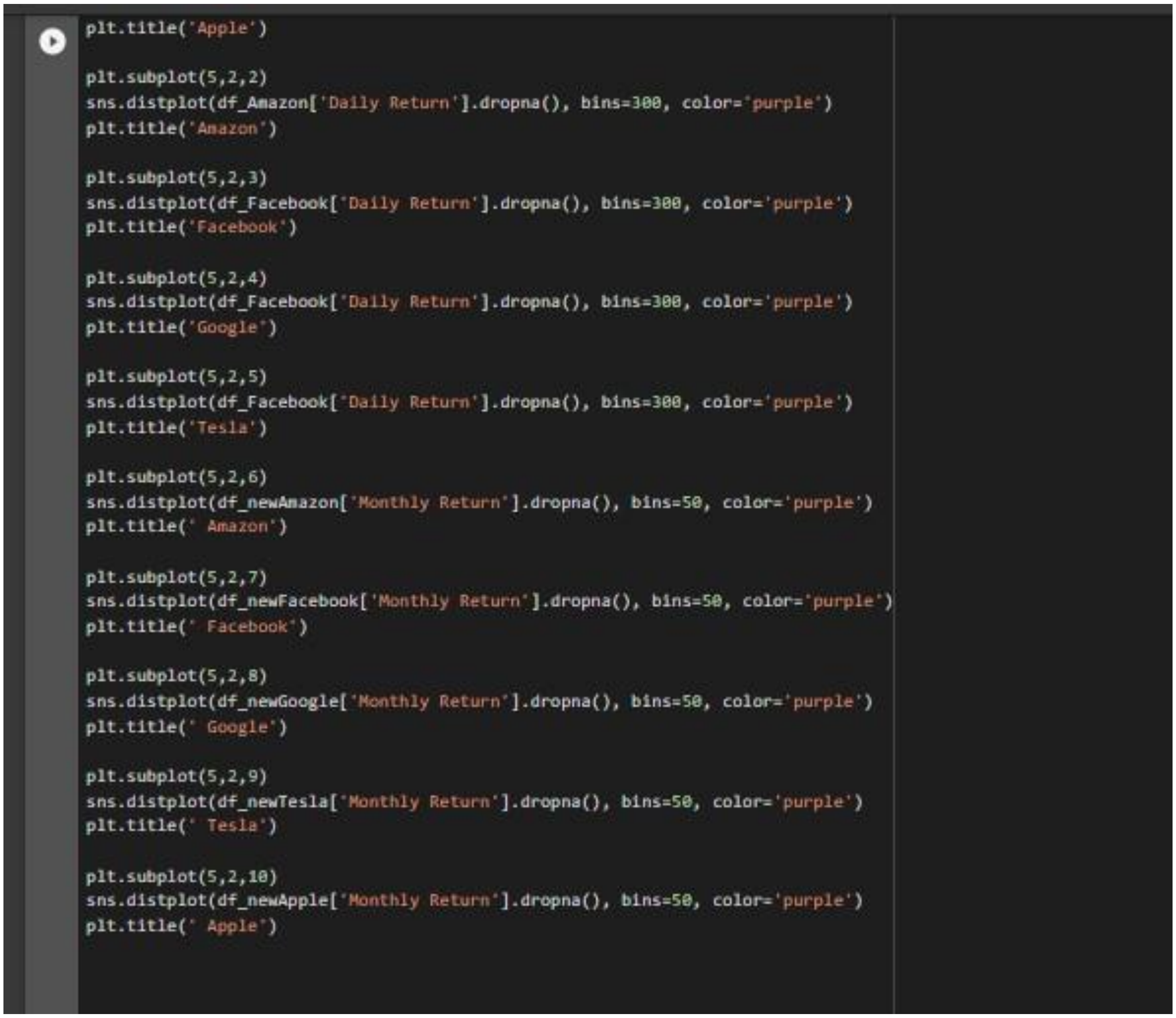
```
[ ] df_Apple.to_csv('index.csv')
```

```
[ ] df_Amazon.to_csv('newAmazon.csv')
```

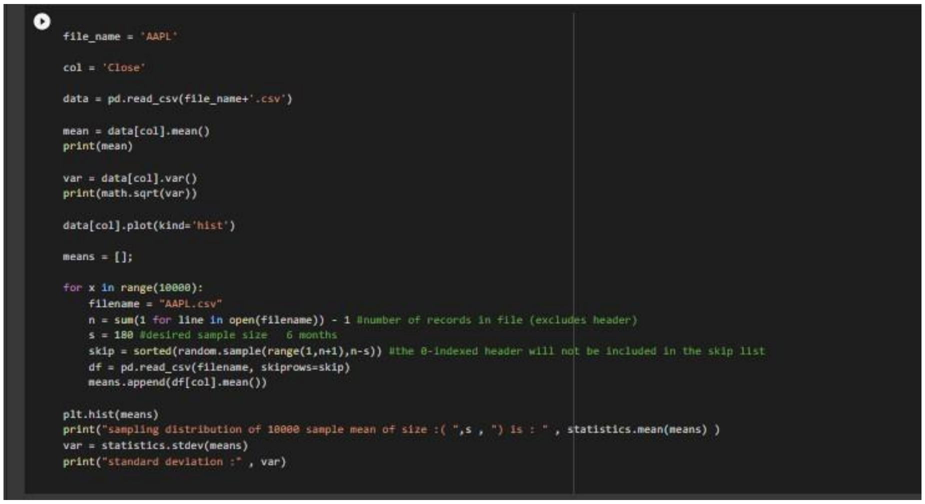
```
[ ] df_Facebook.to_csv('newFacebook.csv')
```

```
[ ] df_Tesla.to_csv('newTesla.csv')
```

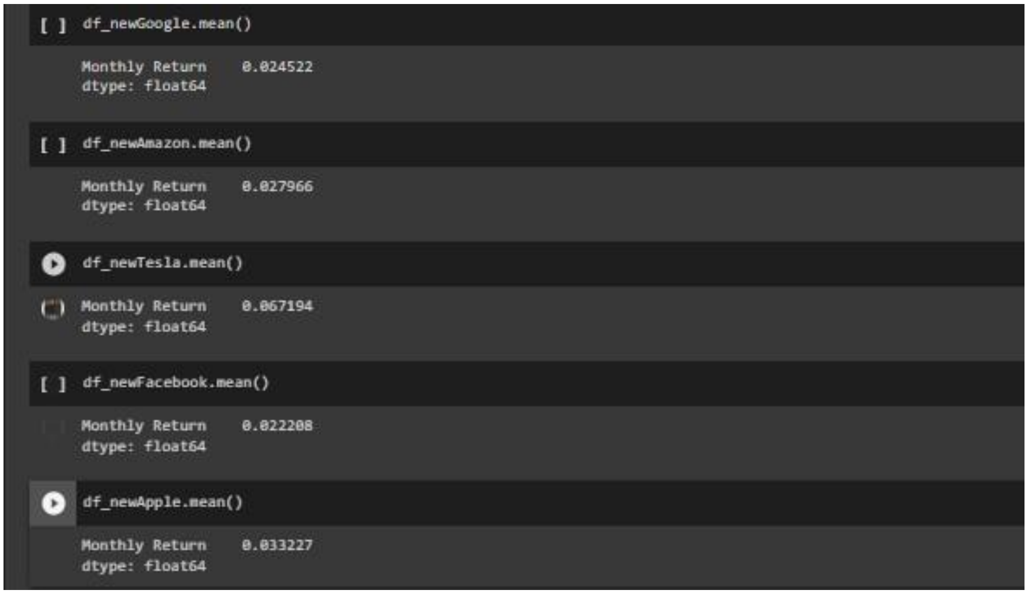
```
[ ] df_Google.to_csv('newGoogle.csv')
```



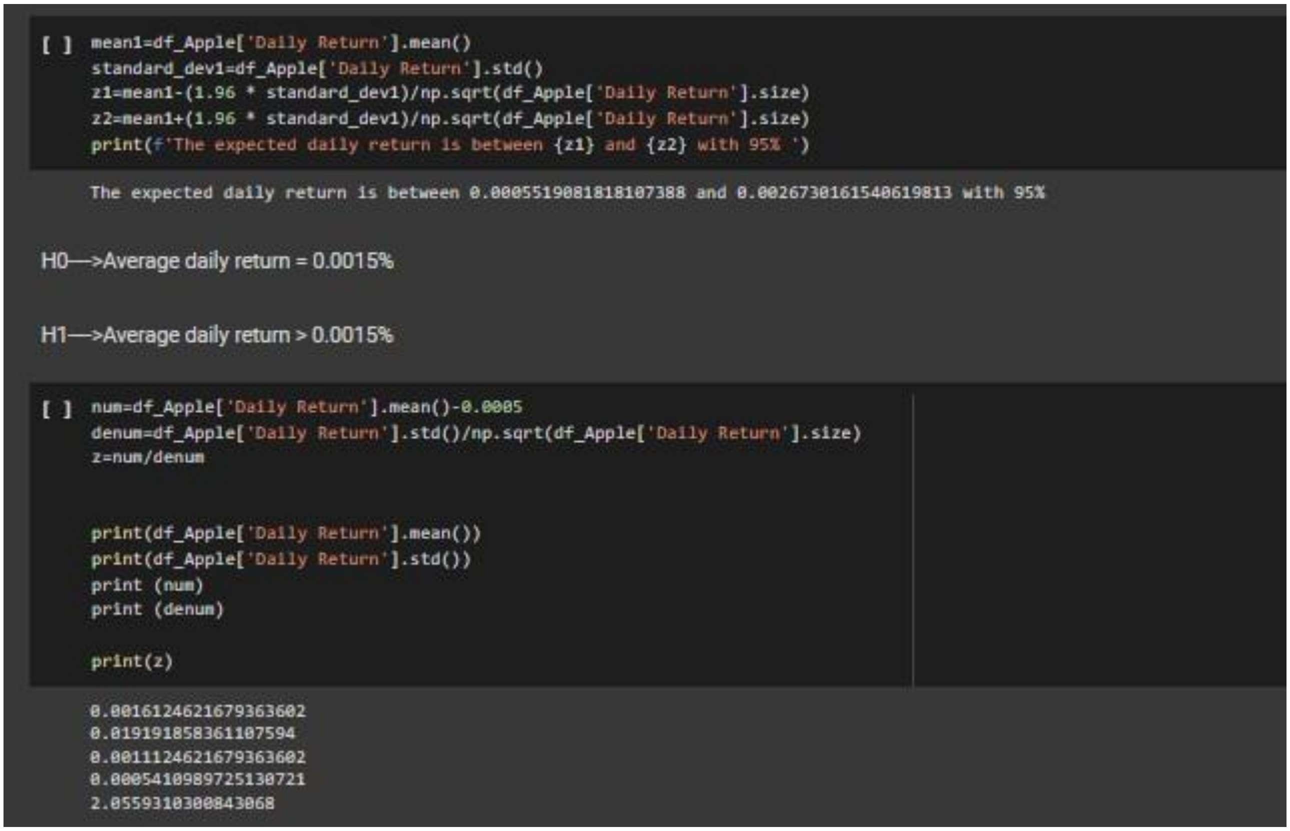
```
plt.title('Apple')

plt.subplot(5,2,2)
sns.distplot(df_Amazon['Daily Return'].dropna(), bins=300, color='purple')
plt.title('Amazon')

plt.subplot(5,2,3)
sns.distplot(df_Facebook['Daily Return'].dropna(), bins=300, color='purple')
plt.title('Facebook')

plt.subplot(5,2,4)
sns.distplot(df_Facebook['Daily Return'].dropna(), bins=300, color='purple')
plt.title('Google')

plt.subplot(5,2,5)
sns.distplot(df_Facebook['Daily Return'].dropna(), bins=300, color='purple')
plt.title('Tesla')

plt.subplot(5,2,6)
sns.distplot(df_newAmazon['Monthly Return'].dropna(), bins=50, color='purple')
plt.title(' Amazon')

plt.subplot(5,2,7)
sns.distplot(df_newFacebook['Monthly Return'].dropna(), bins=50, color='purple')
plt.title(' Facebook')

plt.subplot(5,2,8)
sns.distplot(df_newGoogle['Monthly Return'].dropna(), bins=50, color='purple')
plt.title(' Google')

plt.subplot(5,2,9)
sns.distplot(df_newTesla['Monthly Return'].dropna(), bins=50, color='purple')
plt.title(' Tesla')

plt.subplot(5,2,10)
sns.distplot(df_newApple['Monthly Return'].dropna(), bins=50, color='purple')
plt.title(' Apple')
```



```
file_name = 'AAPL'

col = 'Close'

data = pd.read_csv(file_name+'.csv')

mean = data[col].mean()
print(mean)

var = data[col].var()
print(math.sqrt(var))

data[col].plot(kind='hist')

means = [];

for x in range(10000):
    filename = "AAPL.csv"
    n = sum(1 for line in open(filename)) - 1 #number of records in file (excludes header)
    s = 180 #desired sample size   6 months
    skip = sorted(random.sample(range(1,n+1),n-s)) #the 0-indexed header will not be included in the skip list
    df = pd.read_csv(filename, skiprows=skip)
    means.append(df[col].mean())

plt.hist(means)
print("sampling distribution of 10000 sample mean of size :( ",s , ") is : " , statistics.mean(means) )
var = statistics.stdev(means)
print("standard deviation :" , var)
```



```
[ ] df_newGoogle.mean()

    Monthly Return    0.024522
    dtype: float64

[ ] df_newAmazon.mean()

    Monthly Return    0.027966
    dtype: float64

    df_newTesla.mean()

    Monthly Return    0.067194
    dtype: float64

[ ] df_newFacebook.mean()

    Monthly Return    0.022208
    dtype: float64

    df_newApple.mean()

    Monthly Return    0.033227
    dtype: float64
```



```
[ ] mean1=df_Apple['Daily Return'].mean()
    standard_dev1=df_Apple['Daily Return'].std()
    z1=mean1-(1.96 * standard_dev1)/np.sqrt(df_Apple['Daily Return'].size)
    z2=mean1+(1.96 * standard_dev1)/np.sqrt(df_Apple['Daily Return'].size)
    print(f'The expected daily return is between {z1} and {z2} with 95% ')

    The expected daily return is between 0.0005519081818107388 and 0.0026730161540619813 with 95%
```

H0—>Average daily return = 0.0015%

H1—>Average daily return > 0.0015%

```
[ ] num=df_Apple['Daily Return'].mean()-0.0005
    denum=df_Apple['Daily Return'].std()/np.sqrt(df_Apple['Daily Return'].size)
    z=num/denum


    print(df_Apple['Daily Return'].mean())
    print(df_Apple['Daily Return'].std())
    print (num)
    print (denum)

    print(z)

    0.0016124621679363602
    0.019191858361107594
    0.0011124621679363602
    0.0005410989725130721
    2.0559310300843068
```

References

- Miller-Amp-Freund39s-Probability-and-Statistics-for-engineers
- Kaggle.com
- R.B. Ash and C.A. Dol´eans-Dade. (2000). Probability and Measure Theory, 2nd Ed. Academic Press.
- Chow, Y. S. and H. Teicher. (1997). Probability Theory: Independence, Interchangeability, Martingales, 3rd Ed. Springer-Verlag, New York.
- Chung, K. L. (2001). A Course in Probability Theory, 3rd Ed. Academic Press, San Diego
- Python Essential Reference - David M. Beazley
- Python for Data Analysis by Wes McKinney
- Data Analytics Made Accessible by Dr. Anil Maheshwari
- Python for Everybody: Exploring Data in Python 3 by Dr. Charles Russell Severance