Carlos Cespedes
FIN611 Fall 2020
Professor Taylor
Project 1: Python and Statistics for Financial Analysis

1) High level problem description and background of the quantitative finance/Fintech area that you are interested in studying

When it comes to trying to understand if your stock investment is profitable or not, there are countless tools to use in order to analyze and find trends. With tools such as Excel or Minitab, the ability to import downloadable data from financial websites has allowed the analysis of data to become more effective. Although they are powerful tools, the ability to shorten the amount of recursive work with several lines of code and the ability to handle larger amounts of data has made Python the preferred tool for analysis. For someone who is specifically looking to analyze large amounts of historical data over long periods of time, it's preferable to use Python who shares similar tools as Excel or Minitab, but I also have access to packages which would expand my toolkit. Not only that but after extensive research comparing both Excel and Python, I was better able to understand why Python is becoming a larger preference in the workforce. Although Excel is easier to learn and understand, along with rich visuals such as graphs, Python has attributes such as connectivity (Internet), automation and scalability; something that Excel lacks. The problem I am hoping to solve with this project is to build models that would be able to inform me if my stock is profitable or not using models like Moving Average without having to limit myself to the amount of data I am importing, as well as automating my code to become more efficient.

Using the Youtube Playlist, although it didn’t go over any financial topics, I was able to easily cover the tutorial on how to use Python and how to write your code correctly. After understanding the materials in the Youtube videos, I was able to move on to the article provided. In this article it had more in depth explanations as to how one would proceed to learn and use Python to better understand their data. The article explained lightly some of the methods analysts use when going over their data and how they would clean their data. The topics in the article were covered more in depth in the next section of the module that included the course on Coursera. In the first week of the course we went over how to clean data, switch data types, the simple statistics you can run on your data, and methods to create your models. The project will mostly reflect what was learned during this period excluding several lines of code that I learned through my research that were not part of the module.

2) Description of an associated dataset that you would like build/analyze

The dataset I decided to use would come from my own portfolio, choosing stocks to analyze over the period of 1 year. The reason I am only using 1 year for this project mostly has to do with the fact that Uber IPO was recent and there isn't enough historical data to do a larger subset. In order to keep the model and project consistent I decided to make the one year the common period between both stocks of Uber and Moderna. Overall with this project I hope to find trends within the data, better understand when exactly should I be holding or when I should sell, how profitable is my portfolio, and try to predict future trends. Using Nasdaq's historical data, I am able to download information from both Uber and Moderna going back to September 22, 2019. This historical data includes Date, Closing Price, Volume, Open, High, Low. Most of the data I am importing has to do with the prices of the stock throughout the day as well as the volume of the stocks themselves. For the most part they are all numerical excluding the date, so they should all yield a result when put through the different descriptive methods.

3) Summary stats/descriptive information/data-mining that you would like to perform on this dataset

For this specific data set I would like to make sure that when imported I am using a DataFrame in order to use the basic statics that are involved in the .describe() method. This will include measurements such as count, mode, mean, standard deviation, max and min and finally quartitles. After that is completed I would like to create several different variables to the DataFrame that would be able to provide more stats about my stock later on in the project. Some of the variables I hope to add include Price of Tomorrow, Price Difference, and daily return. Once these variables are added I would like to implement my first model using Moving Average for short term and long term. These terms will be split into sub groups because I will also be using two different periods for the short term as well as the long term. The short terms being used for this model will include periods of 5 and 20 and for the long term it will have periods of 40 and 60. Finally to complete the first portion of this project I would like to include several graphs that would accompany the data in order to better visualize the trends of the Closing price in relations to the Moving Averages.

4) Models that you would like to fit to the data (or more generally models that you are interested in learning about, e.g. look into scikit-learn for specific types of models)

The first part of this project will mostly focus on the collection, modification and displaying of the data. But when it comes to the analysis portion of the project, the model I decided to use was Moving Average. When I was researching further into what Moving average was and the implications it has for my specific data, it allowed me to see how versatile the model is and how many different results we can get depending on the periods in questions. To go more in depth with the model I used specifically in this project, I researched what are the most effective periods most analysts use for their reports. The two types of Moving Averages I researched and will be using are Short term periods and Long term Periods. Short periods would include periods of 5 through 20, while the long term periods would range from 20-60. That is no to say that Moving Averages cannot take more periods such as 100, but relatively speaking the most commonly used would be from 5-20 and 20-60. The reason they are split up into these groups too is because they provide signals that would alert me if something were to happen on the stock, both Fast signal (short period) and Slow signal (long term). The purpose of Moving Average is to create a succession of average values of different subsets that come from the original full dataset and smooth out the constant fluctuation of stock prices.

The way I specifically implemented the Moving Average in this project is by using both Short Term and Long Term periods on both and having 2 models for each period. In other words I plan on conducting a short moving average model for a 5 day period, and a 20 day period. In addition to the short term models, the long term models will include both a 40 day period and finally a 60 day period. With this I'm hoping to learn when exactly my specific stock will be worth holding onto or when it should be fluctuating in price. This will help me better plan for future investments and allow me to minimize my risk when it comes to trying to interpret the market at face value.

5) Describe how you plan on getting started building your dataset, and implementing/applying your models for this project

My overall plan for his project consists of 3 parts which include, the collection of my data, cleaning and modifying the data and finally implementing the models discussed above. The plan for the project is also to follow a similar structure to that of the Coursera online course, the financial article in the module and other relevant research that I found either on Youtube or Stack Overflow.

For the first part of the project I plan on collecting my data from a reliable source either MarketWatch or Nasdaq. Both of these sources provide different measurements that I can include in my analysis as well as allow me to export their data into a CSV file. I have to make sure that when I collect the data, that I am collecting it from the same period in time. This is not to compare the stocks being that they are in different industries and wouldn't have any correlation with each other but rather to keep the project constant. After I find the data needed is collected I will make sure to download the right format in order to export it to Python where I will begin the next step of the plan, cleaning and modifying the data.

Once the data has been collected the next step would be to import this data into Python. Along with importing the CSV into Python we need to make sure that we import all the packages we plan on using for this project. For the project I will be using the package named Pandas and using all the functions that it offers such as DataFrame.

```
#importing packages and importing CSV files
import pandas as pd
ub = pd.read_csv('Uber.csv')
md= pd.read_csv('Moderna.csv')
```

Once I have imported pandas and the csv file I will be to display what the data looks like, check the data types and finally reassign any data type that is incorrect.

```
#Displays the first 5 rows of the data and the last 5 rows
print (ub.head())
print (md.head())
print (ub.tail())
print (md.tail())
```

For this specific data set the two main problems I will run into will have to deal with the Date and the “$” in front of all the prices in the data. Using the .replace() function I will be able to replace all the “$” in the file with just an empty space which will facilitate the data type change. The next step would be to use the .astype() function in order to reassign float to all the prices. The reason I plan on doing this is because when I run the basic statics using the .describe() function, it will not take any object including strings. In order to properly get the basic statics of each column I would need to change the data types before moving on.

```
#Takes out the $ from the data
ub=ub.replace({'\$':"}, regex = True)
md=md.replace({'\$':"}, regex = True)
```

```
#Switch data types
ub.columns = ['Date','Close','Volume', 'Open', 'High', 'Low']
ub = ub.astype(({"Close": float, "Volume": int, "Open": float, "High": float, "Low": float}))

md.columns = ['Date','Close','Volume', 'Open', 'High', 'Low']
md = md.astype(({"Close": float, "Volume": int, "Open": float, "High": float, "Low": float}))
```

In order to verify the changes I have recently made instead of checking the head and tail of the data, I prefer to check the types of the columns to see if they have changed to the correct type.

```
#Lets me know the types in each column
print(ub.dtypes)
print(md.dtypes)
```

The final step of this plan is where the main analysis takes place and where different methods will be used to get my reports. The first part of this step would be to finally use the

.describe () function for both stocks. This will allow me to get basic statics of each column in the CSV file including Closing Price, Volume, Open, High, Low.

```
#Stats on CSV
print(ub.describe())
print(md.describe())
```

After getting that the next step would be to create several different variables that would give insight to how the stock behaves everyday but also help build models down the road. The new variables being used would be Closing Price of tomorrow, Price Difference, and Daily return. The way Closing price is found is by using the .shift() function to move all the numbers in that column up 1 and this would allow for the closing price of the next day to be displayed on the same line as the closing price of today. This specific variable will help in finding the next variable which is Price difference because it allows us to subtract the closing prices of both days and then later display it as another column in the file. Just as before the Price difference column will also help me find the Daily return because I would be able to divide the Price Difference and the Closing Price.

```
#Finding out the Closing Price of tomorrow
ub['Price1']= ub['Close'].shift(-1)
md['Price1']= md['Close'].shift(-1)

#Finding the price difference
ub['PriceDiff']= ub['Price1']- ub['Close']
md['PriceDiff']= md['Price1']- md['Close']

#Daily Return
ub['Return']= ub['PriceDiff']/ub['Close']
md['Return']= md['PriceDiff']/md['Close']
```

Now that the basic statistics have been done on all the columns, the next step would be to use a model in order to have a better idea of the type of investments I have made and if they are profitable or not. Using the Moving Average model would yield better results compared to the basic statics above, and would allow me to better predict my actions with the help of the visuals that Moving average helps create. During this section I plan on creating the line of code but only changing the period. I do plan in a future project to try and automate this and instead have the code ask the user how many periods they desire rather than writing the full line of code recursively.

```
#Short Term 5 Moving Avg Fast Signal
```

```
ub['Average5'] = ub['Close'].rolling(window=5).mean()
md['Average5'] = md['Close'].rolling(window=5).mean()

#Short Term 20 Moving Average Fast Signal
ub['Average20'] = ub['Close'].rolling(window=20).mean()
md['Average20'] = md['Close'].rolling(window=20).mean()

#Long Term 40 Moving Avg Slow Signal
ub['Average40'] = ub['Close'].rolling(window=40).mean()
md['Average40'] = md['Close'].rolling(window=40).mean()

#Long Term 60 Moving Avg Slow Signal
ub['Average60'] = ub['Close'].rolling(window=60).mean()
md['Average60'] = md['Close'].rolling(window=60).mean()
```

Once this has been completed I would be able to use these new variables and measurements to create visuals that would help me better understand the information. Besides using .head() and .tail() to see the actual data, I decided to use .plot() instead because it would allow me to see the trend of the stocks and would actually provide information that will affect my investments unlike the other two. So for the final part of the project I plan on plotting the closing prices of the stocks alongside the different moving averages. This would be the final step that would allow me to interpret the data and report my findings on the stocks in question.

```
#Plot moving average
ub['Close'].plot(legend=True)
ub['Average5'].plot(legend=True)
ub['Average20'].plot(legend=True)
ub['Average40'].plot(legend=True)
ub['Average60'].plot(legend=True)

md['Close'].plot(legend=True)
md['Average5'].plot(legend=True)
md['Average20'].plot(legend=True)
md['Average40'].plot(legend=True)
md['Average60'].plot(legend=True)
```