**\*\* represents any addition or discussion need to made**

**Title:** Stock Analysis and Visualization Package

**Authors:** Henry Fox, Syed Hani Haider,

**Summary:**

The purpose of this package is to provide the user with basic equity performance statistics and a graphical representation of a user selected publicly traded company’s share price over time. Once the user downloads the selected company historical share price data from Yahoo finance our program will check the time series data for any abnormalities. The program will inspect the share price data for dates or missing data and then perform the analysis.

The analysis will consist of statistical data displayed in a table format with: the highest high and the lowest low for the time period, the biggest percentage move up or down in a single day and the dates associated with these data points. The graph will display the company's stock performance over the selected time frame in addition to the 50 and 200 day moving average. We believe this package will provide fast data analysis of a user selected publicly traded company to better inform their investment decision.

\*\* comparing stocks is also important as it gives investor more knowledge that how is one stock doing in regards with other, this will have the complete summary and also some graphs for both the current and stock which we are looking and the stocks we want to comparing it with.

**Proposal Design:**

Class Stocks:

Attributes:

stocks\_ filename: this will have all the data for the stock which we want the data to be presented

major\_index :

This will have the file \*\*(please add info for it i don't know have what will the data have )

Methods :

1. Summary
2. Rasing error
3. Graphs
4. Moving average
5. Comparing stocks summary
6. Comparing stocks graph

Def \_ \_init\_ \_(self, stocks\_filename:str , major\_index:str):

Constructor: creates an new instance for stock

parameter :

Self -- the current object

stocks \_filename:string -- the data for the stocks

major \_index:string -- the data for major index

Def rasing\_error(self):

This will raise any error, this method will be called in all the method or that we can raise error

Def summary(self):

Method:summary: gets the complete summary of stocks

Parameter:

Self-- the current object:

Returns the complete summary which will have stock price and date, where it was highest stock, lowest stock, largest moving up in a single day % and largest moving downward in a single day by %

Def graph(self,start\_date=self.start date, end\_date=self.end\_date):

Method -- graph

Parameters

Self -- the current object

Start\_date: the date which the stock price we want to see

end \_date: the date which the stock price we want to end

Returns a line graph of the stock prices

Def moving\_average(self,start\_date=self.start date, end\_date=self.end\_date)

Method -- moving average

Parameters

Self -- the current object

Start\_date: the date which the stock price we want to see

end \_date: the date which the stock price we want to end

Returns the moving average of the stocks

Def comparing\_stocks(self ,comparing\_stock\_filename):

Method -- comparing\_stocks

parameters :

Comparing\_stock\_filename-- file we are going to compare with

Returns \*\*\* need to discuss what should be returning

Def comparing\_stocks\_graph((self,start\_date=self.start date, end\_date=self.end\_date):

Method -- comparing\_stocks\_graphs

Parameters

Self -- the current object

Start\_date: the date which the stock price we want to see

end \_date: the date which the stock price we want to end

Returns a line graph of the stock prices

Below the line need to be removed before the submission

----------------------------------------------------------------------------

Class Stocks

\_\_init\_\_(filename\_stocks, filename\_major\_index ):

#uploade the data

(3/29 -1(from the current date ))not equal to the date :

Print(file outdates)

Def summary(self):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date range2012 - 2020 | Highest High | Lowest Low | Largest Move Up in a single day % | Largest Move Downward in a single day% |
| Price | $250 | $25 | 15% | -10% |
| Date | March 20, 2019 | February 15, 2018 | March 20, 2019 | November 8, 2018 |
|  |  |  |  |  |

Def Rasing errors(self):

#rasise error if the data is not correct

Def moving\_avgrage(self,start\_date, end\_date):

#raise an error if lesss than certain number

#gets the average of certain days

Def graphs(self):

#calling the moving average method

#add major index

Def change\_of\_data(seld, first month, second\_month:)

#returns the graph with the comparison

Def comparing\_stock(self, stock\_name1,stock\_name2)

Def capering\_stocks\_graphs

TEST CASES:

Every mthod should have atleast 10 test case

Worst case

Best case