

12/13/2020

Cahir Geraghty

CS-110 A

Final Project Report

The purpose of this program is to allow the user to pick from 4 different market sectors and compare two of them. For this purpose 2 classes have been created to store and analyze the data from the sectors. The user is prompted twice to input an integer between 1 and 4 each one corresponding to a different sector. The files are then read and analyzed to create a graph which compares the percentage change per day of the two different sectors.

```
4
5  class Day:
6
7      def __init__(self, stock_name, row):
8          self.stock_ticker = stock_name
9          self.date = row[0]
10         self.opening = float(row[2].replace(',',''))
11         self.high = float(row[3].replace(',',''))
12         self.low = float(row[4].replace(',',''))
13         self.closing = float(row[1].replace(',',''))
14         self.daily_average = self.get_daily_average()
15         self.daily_change = float(row[6].replace('%',''))
16
```

This image shows the Day class that I created to contain the data for a given day in the historical data. This class has 8 variables: stock_ticker, which is the name of the stock; date, which is the date that the information corresponds to; opening, which is the

opening price of the stock; high, which is the highest the price reached on that day; low, which is the lowest price reached on that day; closing, which is the closing price of the stock; daily_average, which is the average price between opening and closing; and daily_change, which is a percentage value of how much the price of the stock has changed that day. The day class has two methods: get_daily_average, which computes the daily_average; and a str method that modifies the print statement.

```
23 class Sector():
24     def __init__(self, file):
25         self.file = file
26         self.filename = self.getfilename()
27         self.daylist = self.getdaylist()
28
29     def getfilename(self):
30         return self.file.split(" ")[-1][:-4]
31
32     def getdaylist(self):
33         daylist = []
34         with open(self.file) as stockfile:
35             rowlist = csv.reader(stockfile)
36             for rowindex, row in enumerate(rowlist):
37                 if rowindex != 0:
38                     daylist.append(Day(self.filename, row))
39         return daylist
40
41     def graph(self):
42         datelist = [day.date for day in self.daylist]
43         dailychangelist = [day.daily_change for day in self.daylist]
44         plt.plot(datelist, dailychangelist, label=self.filename)
45
```

This image shows the Sector class which takes a csv file and reads it into a list of Days.

The Sector class has 3 variables: file, which is the file path for the csv reader to use; filename, which is the name of the sector chosen; and daylist, which is a list of Days

containing the information from the file. The Sector class has three methods:

getfilename, which finds the Sector's name; getdaylist, which reads the file and creates the list of Days that holds the information; and graph, which plots the data with the date as the x-axis and the daily_change as the y-axis.

```
47 def main():
48     validchoice = 0
49     first_file_choice = eval(input('Enter: \n 1 for Health Care \n 2 for Technology \n 3 for Financials \n 4 for Industrials \n'))
50     if first_file_choice == 1:
51         first_file_path = 'C:\\Users\\cgera\\OneDrive\\Desktop\\Stevens Classes 2020\\CS 110-A\\2020.01.01 - 2020.05.01 DJHealthcare.csv'
52         first_sector = Sector(first_file_path)
53         first_sector.graph()
54     elif first_file_choice == 2:
55         first_file_path = 'C:\\Users\\cgera\\OneDrive\\Desktop\\Stevens Classes 2020\\CS 110-A\\2020.01.01 - 2020.05.01 DJTech.csv'
56         first_sector = Sector(first_file_path)
57         first_sector.graph()
58     elif first_file_choice == 3:
59         first_file_path = 'C:\\Users\\cgera\\OneDrive\\Desktop\\Stevens Classes 2020\\CS 110-A\\2020.01.01 - 2020.05.01 DJFinancials.csv'
60         first_sector = Sector(first_file_path)
61         first_sector.graph()
62     elif first_file_choice == 4:
63         first_file_path = 'C:\\Users\\cgera\\OneDrive\\Desktop\\Stevens Classes 2020\\CS 110-A\\2020.01.01 - 2020.05.01 DJIndustrials.csv'
64         first_sector = Sector(first_file_path)
65         first_sector.graph()
66     else:
67         print("Invalid Choice")
68         validchoice = 1
69
```

This image shows the first third of the main function which prompts the user to input which market sector they would like to choose first. The options are “1 for Healthcare, 2 for Technology, 3 for Financials, 4 for Industrials”. If, elif, and else statements are used to assess which option is chosen, if 1, 2, 3, or 4 are imputed the program instantiates the sector using the associated file and plots the graph. If anything else is imputed then the program prints “Invalid Choice” and sets the variable validchoice to 1 which then closes the program. If a valid option is selected then the program repeats the process for the second market sector. Be advised that the program allows you to select the same sector twice which is an intentional feature to allow for further expansion of the program. If a valid option is chosen again the program continues to the last portion of the code.

```
91     if validchoice == 0:
92         plt.xlabel('Date')
93         plt.ylabel('Daily Change')
94         plt.title('Sectors Comparison Graph')
95         plt.xticks(rotation=90)
96         plt.legend()
97         plt.grid()
98         plt.show()
99
```

This image shows the last portion of the main function which assigns labels to the graph, creates a legend, adds a grid, and shows the graph. Note that this code uses an if statement to ensure that two valid options have been selected before this code runs.

The program then runs the main() method. This program uses Dow Jones Sector indexes as data sets for each sector. Due to the fact that Yahoo finance doesn't have the historical data for these indexes they were downloaded from Investing.com and the program was adjusted to read the input from their files.