

# Equities Analysis in Various Sectors

Tyson Werner

## Section 1:

For my project, I choose to study fifteen different stocks from five sectors represented in the stock market: energy, financial, healthcare, industrial, health. The full capacity of the program allows you to view the price chart of a stock or search the price of a stock based on a specific date (user input). It was built on Pycharm and uses the csv, matplotlib, and os libraries. The following is an input and output:

```
Please choose an energy stock to analyze
1 for Antero Midstream
2 for Valero Energy
3 for EQT Corps
1
Please choose your method of analysis
1 for the company stock price chart
2 for the stock price on a specific day
2
Please enter a date (Ex.2020-11-25) to see the stock price for that day
2020-05-20
2020-05-20 4.38
```

```
2020-05-19,4.040000,4.120000,3.870000,4.030000,3.632588,7431000
2020-05-20,4.090000,4.420000,4.080000,4.380000,3.948074,14474800
2020-05-21,4.380000,4.530000,4.135000,4.300000,3.875963,6220800
```

## Section 2:

The target audience of this program is any individual investors who would like to conduct research on stocks within a variety of different sectors within the equities market. In each sector, there are three stocks, allowing for comparison of different stocks within a sector or using stocks from each sector to understand which industry outperformed/underperformed at this time. On top of the variety of sectors and equities to pick from, there are two different analytical methods to choose from: broad movement

analysis or historical referencing based on the day. Therefore, the user is able to get a big picture understanding as well as gauge price movements due to perhaps, the price after an earnings report on any given day.

### Section 3:

There are decision structures, strings, lists, files, functions, and classes involved in the program. The data was collected from Yahoo Finance. The following functions are created to retrieve the file paths of the three stocks in each sector folder and to return them in a list. The energy() function is expanded to show the function code largely used in each.

```
def energy():
    csv_files = "/Users/tyson/PycharmProjects/CS110Project/Energy/"
    csv_file_path_list_energy = []
    for file_path in os.listdir(csv_files):
        csv_file_path_list_energy.append(csv_files + "/" + file_path)

    return csv_file_path_list_energy

def financials():...

def healthcare():...

def industrial():...

def hotels():...
```

The following functions are set up respectively to import the time series of each stock, to prepare graphing abilities, and to open the graph.

```

def timeseries(csv_file_path):
    date_list, close_list = [], []
    with open(csv_file_path) as csv_file:
        for row in list(csv.reader(csv_file))[1:]:
            date_list.append(row[0])
            close_list.append(float(row[4]))
    return date_list, close_list

def graph(x, y):
    plot(x, y)

def open_graph():
    xlabel("Date")
    ylabel("Closing Price")
    xticks(rotation=90)
    show()

```

Additionally, a class “Day” was set up to receive the date given by the user and the corresponding file path to determine the price of the stock on the day given and yield its output.

```

class Day:

    def __init__(self, user_input, stock):

        self.manual_date = user_input

    def compareValues(self, stock):
        date_list = []
        price_list = []
        csv_files = stock
        with open(csv_files) as csv_file:
            for col in list(csv.reader(csv_file))[1:]:
                if str(col[0]) == self.manual_date:
                    date_list.append(col[0])
                    price_list.append(float(col[4]))
                    print(self.manual_date + " " + str(float(col[4])))

```

At the beginning of the program, each function is run and the results saved to five separate lists, each containing the file paths of the three stocks in the sector. The user

is introduced to their choice of sectors to pick from.

```
This program will allow you to analyze the following equities sectors based on three stocks in each:  
Energy, Financial, Healthcare, Industrial, Hotels
```

```
Please choose a sector:
```

- 1 for Energy
- 2 for Financial
- 3 for Healthcare
- 4 for Industrial
- 5 for Hotels

```
ans = int(input(  
    "Please choose a sector: \n 1 for Energy \n 2 for Financial \n 3 for Healthcare \n 4 for Industrial \n 5 for Hotels \n"))
```

After making their selection, if statements are used to separate the remaining code into five sections (broken up by sector). I will only show the first sector which is energy.

```
1  
Please choose an energy stock to analyze  
1 for Antero Midstream  
2 for Valero Energy  
3 for EQT Corps
```

At this point, the user picks one of three stock choices and is then asked to choose their method of analysis: either viewing a price chart or deciding to pick a date to return the stock's price only on that day. Each choice has error endpoints or completes the task.

```
# energy  
if ans == 1:  
    ans2 = int(input(  
        "Please choose an energy stock to analyze\n1 for Antero Midstream\n2 for Valero Energy\n3 for EQT Corps\n"))  
    if ans2 < 1 or ans2 > 3:  
        print("You've enter an invalid command.")  
        exit()  
    ans3 = int(input(  
        "Please choose your method of analysis\n1 for the company stock price chart\n2 for the stock price on a specific day\n"))  
  
    if ans3 == 1:  
        date1, close1 = timeseries(energyfiles[ans2 - 1])  
        graph(date1, close1)  
        open_graph()  
        exit()  
  
    elif ans3 == 2:  
        stock = energyfiles[ans2 - 1]  
        date = str(input("Please enter a date (Ex.2020-11-25) to see the stock price for that day\n"))  
        price = Day(date, stock)  
        price.compareValues(stock)  
        exit()  
  
    elif ans3 != 1 & ans3 != 2:  
        print("You've enter an invalid command.")  
        exit()
```

The following are examples for Antero stock in each path:

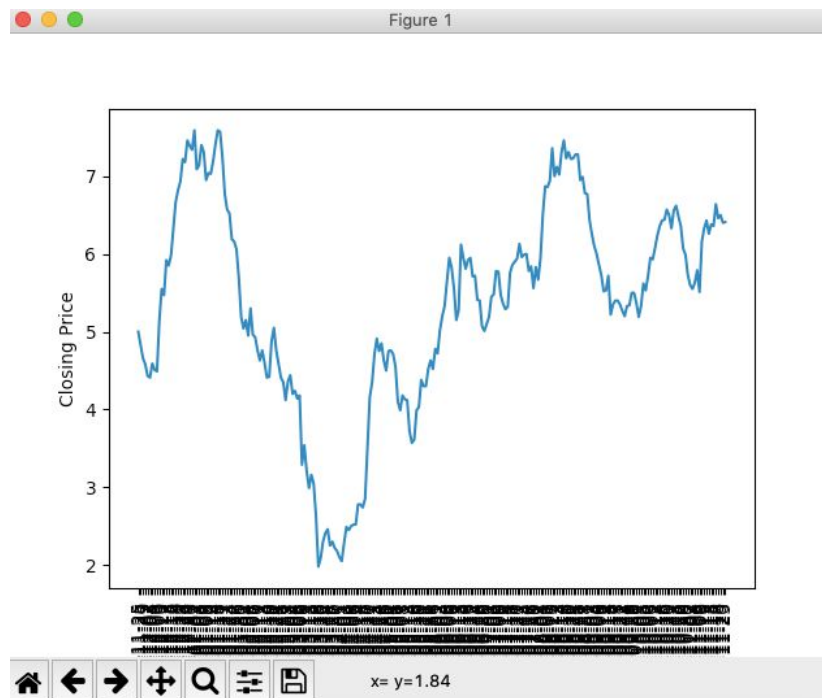
1. Path of price on a particular day

```
Please choose an energy stock to analyze
1 for Antero Midstream
2 for Valero Energy
3 for EQT Corps
1
Please choose your method of analysis
1 for the company stock price chart
2 for the stock price on a specific day
2
Please enter a date (Ex.2020-11-25) to see the stock price for that day
2020-05-20
2020-05-20 4.38
```

This is the date/price in the actual CSV file to verify program correctness.

```
2020-05-19,4.040000,4.120000,3.870000,4.030000,3.632588,7431000
2020-05-20,4.090000,4.420000,4.080000,4.380000,3.948074,14474800
2020-05-21,4.380000,4.530000,4.135000,4.300000,3.875963,6220800
```

2. Path of the general stock price chart



#### Section 4:

The biggest challenge was creating the class that took the user input 'date' and searched the given stock based on the date and yielded the price. Writing the function for this was difficult as there was no direct path to writing the steps to get there. It was also difficult to adjust the stock analysis' based on choosing different industries and breaking up each folder by industry. It paid off for me to focus on organization and structure, as can be seen with the five functions used for industries, the class for "date-stock-price" analysis, and the functions for graphing a stock. This allowed me to avoid repetition where possible in my main function.

#### Section 5:

If I had more time, it would be interesting to add macroeconomic and fiscal components to my program to graph alongside the given stocks. For example, it would be interesting to compare the performance of the stocks to inflation, unemployment, income, coronavirus, and tax rates during the same period. Additionally, when Congress passed a COVID relief bill in the spring, it would be interesting to add a secondary graph with dates of significant news updates on federal fiscal stimulus alongside these other measures to understand any price movements that happened as a result of any legislation.

