Harshil Cherukuri

CS-110 Project

Professor Kevin Ryan

12/13/2020

Sector Analysis Program

Overview and Summary of Project:

For my Final Project, I was particularly interested in using user inputs to determine what users wanted from my Sector Analysis Program. So, I selected CSV data from five sectors of the U.S Stock Market (Healthcare, Technology, Financials, Pharmaceuticals, and Crude Oil). For those sectors, I selected some well-known Funds/ETFs that represent those sectors with well-balanced portfolios. After they select their sector, the program will ask for two days to enter ranging from 2020-030-02 to 2020-10-30. When they enter those days, the date, opening price, and closing price will be displayed for each day. Also, the difference in points and percent for the day is printed on the next line. However, if they enter a day that is not a weekday or either a national holiday, then the program will tell the user to re-run the program. After the data for both days is inscribed, the program will compare both the days based on the opening price, closing price, and overall change in both prices from those days.

The QUESTION I am trying to answer through my program is..

- **How did each sector fare during the coronavirus pandemic and how did that alter the growth/decline of the Fund/ETF associated with the sector?**

Target Audience:

When I designed this type of program, my target audience was people who were amatuer investors in the U.S Stock Market. This program would be suitable for them since it shows the different sectors within the market and key metrics to follow to determine whether a sector was either growing/declining during the pandemic

Specific Programming Techniques Used (1):

```
import csv

def healthcare():
    print("(Fund/ETF: IDNA)")
    print("In this program you will be able to look at data starting from one date to a another date in the")
    print("range starting from 2020-03-02 to 2020-10-30 ")
    num_of_months = 0
    total_diff_healthcare = 0
    total_perc_healthcare = 0
    list_day_1 = ['']
    list_day_2 = ['']
    with open('Healthcare.csv', 'r') as file_1:
        reader_1 = csv.DictReader(file_1)

        healthcare_input_1 = str(input("What is the first day you want to look at? (Range: 2020-03-02 to 2020-10-30): "))
        healthcare_input_2 = str(input("What is the second day you want to look at? (Range: 2020-03-02 to 2020-10-30): "))
        print()

        for row in reader_1:
            temp_date = str(row['Date'])
            x = float(row['Open'])
            y = float(row['Close'])
            diff_healthcare = y - x
            percent_diff_1 = (y - x) / x
            total_diff_healthcare += diff_healthcare
            total_perc_healthcare += percent_diff_1
            num_of_months += 1

            if healthcare_input_1.strip() == temp_date.strip():
                print("----------------------------Printing data for date 1----------------------------")
                print(row)
                print("The Date, Open, & Close for this day: ", row['Date'], row['Open'], row['Close'])
                if x > y:
                    print("The sector experienced a downward trend on this day")
                elif y > x:
                    print("The sector experienced a upward trend this day")
                print("The difference between the closing and opening price on this day is", diff_healthcare)
                print("The percent growth/decline of the sector for this day is %", percent_diff_1 * 100)
                print("The average points growth/decline of the sector until this point is",
                      (total_diff_healthcare / num_of_months), "points")
                print("The average percent growth/decline of the sector until this point is %",
                      (total_perc_healthcare / num_of_months) * 100)
                print()
```

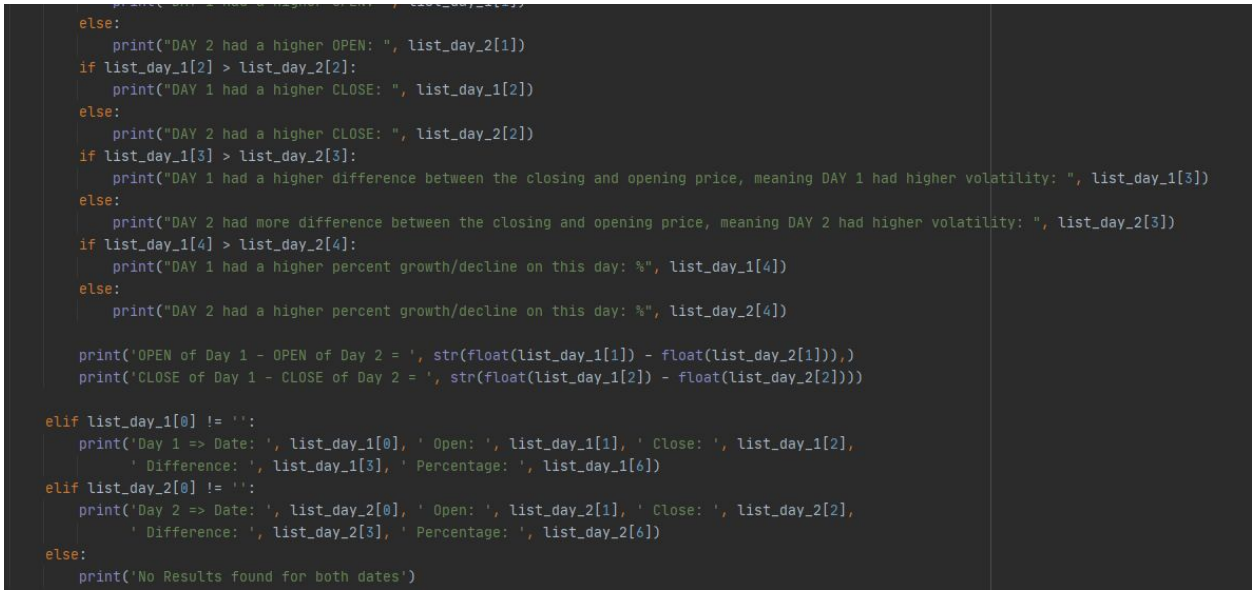
- For the first part of my healthcare function, I set my initial variables and also imported the CSV file containing the stock market data for the Fund/ETF: IDNA. Afterward, I set user inputs as the two datasets that I am going to show for the entered dates ranging from 2020-03-02 to 2020-10-30. For those individual date inputs, I planned to display the “Date”, “Open”, and “Close” rows on those specific days entered. Within the “for” loop, I had an if-elif statement to determine whether, on those respective days, the Fund/ETF experienced either an upward or downward trend. Lastly, I printed out the difference between the closing and opening prices in both points and percents, and the average point and percent growth on those respective dates.

Specific Programming Techniques Used (2):

```
        list_day_1[0] = row['Date']
        list_day_1.append(row['Open'])
        list_day_1.append(row['Close'])
        list_day_1.append(diff_healthcare)
        list_day_1.append((percent_diff_1 * 100))
        list_day_1.append((total_diff_healthcare / num_of_months))
        list_day_1.append(((total_perc_healthcare / num_of_months) * 100))

        continue

    elif healthcare_input_2.strip() == temp_date.strip():
        print("----------------------------Printing data for date 2----------------------------")
        print(row)
        print("The Date, Open, & Close for this day: ", row['Date'], row['Open'], row['Close'])
        if x > y:
            print("The sector experienced a downward trend on this day")
        elif y > x:
            print("The sector experienced a upward trend this day")
        print("The difference between the closing and opening price on this day is", diff_healthcare)
        print("The percent growth/decline of the sector for this day is %", percent_diff_1 * 100)
        print("The average points growth/decline of the sector until this point is",
              (total_diff_healthcare / num_of_months), "points")
        print("The average percent growth/decline of the sector until this point is %",
              (total_perc_healthcare / num_of_months) * 100)
        print()

        list_day_2[0] = row['Date']
        list_day_2.append(row['Open'])
        list_day_2.append(row['Close'])
        list_day_2.append(diff_healthcare)
        list_day_2.append((percent_diff_1 * 100))
        list_day_2.append((total_diff_healthcare / num_of_months))
        list_day_2.append(((total_perc_healthcare / num_of_months) * 100))

        break

print("If there is no data printing for your day, then the date you entered was either not a weekday, a national holiday, or an input out of bounds ")
print()

print("_________________ Summary _________________")
if list_day_1[0] != '' and list_day_2[0] != '':
    if list_day_1[1] > list_day_2[1]:
        print("DAY 1 had a higher OPEN: ", list_day_1[1])
```

- In the above picture, there is the same procedure to print out the data for the second day the user inputs into the program. After the output is printed out, both variables, list_day_1 and list_day_2 append the data that is calculated in the program into their respective lists. The information that is appended includes the “Row”, “Date”, “Close”, point difference, percent difference, average point difference, and average percent difference on those days. After all the details are in their respective lists, I made a summary section within the program. In the summary section, it compares both lists based on the values that were stored in their respective rows.

Specific Programming Techniques Used (3):



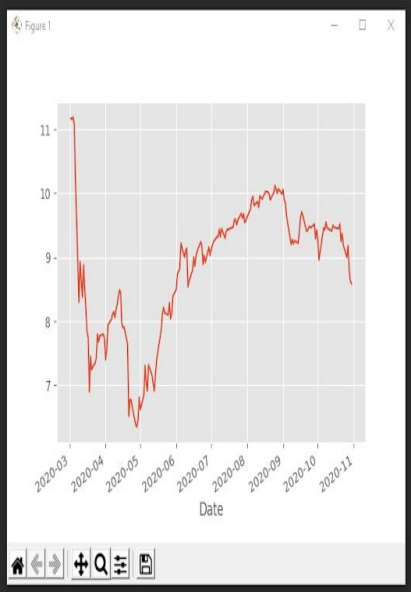
```
    else:
        print("DAY 2 had a higher OPEN: ", list_day_2[1])
    if list_day_1[2] > list_day_2[2]:
        print("DAY 1 had a higher CLOSE: ", list_day_1[2])
    else:
        print("DAY 2 had a higher CLOSE: ", list_day_2[2])
    if list_day_1[3] > list_day_2[3]:
        print("DAY 1 had a higher difference between the closing and opening price, meaning DAY 1 had higher volatility: ", list_day_1[3])
    else:
        print("DAY 2 had more difference between the closing and opening price, meaning DAY 2 had higher volatility: ", list_day_2[3])
    if list_day_1[4] > list_day_2[4]:
        print("DAY 1 had a higher percent growth/decline on this day: %", list_day_1[4])
    else:
        print("DAY 2 had a higher percent growth/decline on this day: %", list_day_2[4])

    print('OPEN of Day 1 - OPEN of Day 2 = ', str(float(list_day_1[1]) - float(list_day_2[1])),)
    print('CLOSE of Day 1 - CLOSE of Day 2 = ', str(float(list_day_1[2]) - float(list_day_2[2])))

elif list_day_1[0] != '':
    print('Day 1 => Date: ', list_day_1[0], ' Open: ', list_day_1[1], ' Close: ', list_day_1[2],
          ' Difference: ', list_day_1[3], ' Percentage: ', list_day_1[6])
elif list_day_2[0] != '':
    print('Day 2 => Date: ', list_day_2[0], ' Open: ', list_day_2[1], ' Close: ', list_day_2[2],
          ' Difference: ', list_day_2[3], ' Percentage: ', list_day_2[6])
else:
    print('No Results found for both dates')
```

- In the mentioned picture, you could see the comparisons that take place between both days…
    - Comparisons:
        - Which day had a Higher OPEN?
        - Which day had a Higher CLOSE?
        - Which day was more volatile?
        - Which day had a higher decline.growth ratio?
        - What was the difference in Opening Price on those two days?
        - What was the difference in Closing Price on those two days?

- Throughout each function, my data was collected from each of the CSV files and analyzed through the “for” loop, which included multiple if-elif-conditions and mathematical operations by using function variables.

Sample Output/User Interface:

```
This program tries to help the user determine the best performing stocks from each sector
The sectors that are all analyzed within this program are healthcare, technology, pharmaceuticals, financials, and oil
Please type in the sector you want to look at (in lowercase): oil
Thanks for choosing to look at the oil sector

Here is the daily data showcasing the general trend of the oil sector from 2020-03-02 to 2020-10-30
(Fund/ETF: HUC.TO)
In this program you will be able to look at data starting from one date to a another date in the
range starting from 2020-03-02 to 2020-10-30
What is the first day you want to look at? (Range: 2020-03-02 to 2020-10-30): 2020-03-02
What is the second day you want to look at? (Range: 2020-03-02 to 2020-10-30): 2020-10-30

--------------------------Printing data for date 1--------------------------
{'Date': '2020-03-02', 'Open': '10.990000', 'High': '11.280000', 'Low': '10.930000', 'Close': '11.170000', 'Adj Close': '11.170000', 'Volume': '8600'}
The Date, Open, & Close for this day:  2020-03-02 10.990000 11.170000
The sector experienced a upward trend this day
The difference between the closing and opening price on this day is 0.17999999999999972
The percent growth/decline of the sector for this day is % 1.6378525932666833
The average points growth/decline of the sector until this point is 0.17999999999999972 points
The average percent growth/decline of the sector until this point is % 1.6378525932666833

--------------------------Printing data for date 2--------------------------
{'Date': '2020-10-30', 'Open': '8.520000', 'High': '8.600000', 'Low': '8.580000', 'Close': '8.590000', 'Adj Close': '8.590000', 'Volume': '10400'}
The Date, Open, & Close for this day:  2020-10-30 8.520000 8.590000
The sector experienced a upward trend this day
The difference between the closing and opening price on this day is 0.07000000000000028
The percent growth/decline of the sector for this day is % 0.8215962441314587
The average points growth/decline of the sector until this point is -0.055857988165680494 points
The average percent growth/decline of the sector until this point is % -0.536805614492943

If there is no data printing for your day, then the date you entered was either not a weekday, a national holiday, or an input out of bounds

------------------ Summary ------------------
DAY 2 had a higher OPEN  8.520000
DAY 2 had a higher CLOSE  8.590000
DAY 1 had a higher difference between the closing and opening price, meaning DAY 1 had higher volatility  0.17999999999999972
DAY 1 had a higher percent growth/decline on this day % 1.6378525932666833
OPEN of Day 2 - OPEN of Day 1 =  -2.4700000000000086
CLOSE of Day 2 - CLOSE of Day 1 =  -2.58
```



Challenges:

- I encountered some challenges when using if-elif statements within the “for” loop when creating the program. The indentation also proved to be difficult since there were a lot of loops and if-elif statements, and I had to be aware of where the individual sections of each program ended. Also, since I had a lot of if-elif statements, I had to know when to continue and break the loops for the program to finish execution. Lastly, the labeling of the variables became vital when formulating my program. Since there were so many variables, I had to name the variables properly, or else it would have been harder to debug the program for any errors.

Future Extensions:

- Some extensions that would be interesting to add would be to have a graphic showing those specific two dates, and using specific indicators to determine future growth/decline for the sector