EXPLORATORY DATA ANALYSIS

Analyzing and Visualizing Stock Market Trends through Python

FEBUARY 3, 2024

JALIN ROBERTS
Bellarmine University | DS-450-01

Section A: Introduction

For my project, I wanted to work with stock data that used Citadel LLC as a baseline for picking the stocks. The goal of finding out which stocks I was going use became an interesting game of cat and mouse. However, one of quicker solves to this search was the publicly available 13F filings (https://13f.info/manager/0001423053-citadel-advisors-llc). The most recent time span for which I could access an entire year's worth of trades was 2022, and so I downloaded Citadel's Quarter 1, Quarter 2, Quarter 3, and Quarter 4's 13F filings. Thankfully, these .csv files are near perfectly cleaned and ready to be merged. Once I had all four files, I opened up VS Code and threw together some quick code to merge all four .csv files. I set all the file paths, setup a data frame, and appended each quarter's data. After that, I converted this to a .csv file and stored the updated merged_data file so I could call back to it.

From there, I set a variable to the file path's location and read in the updated file. I then selected the 'Sym' column where the ticker symbols (or company names) are located and picked a random cell from the column and repeated this 5 total times. For the purpose of my project, I wanted to pick the top 5 holdings in this file, which I did by visually checking for the top holdings by volume. I did 5 random stocks as to give the weight of the top 5 holdings some potential balance for diversifying the portfolio. Doing this allows for possibility to distribute the volatility of the 10 stocks more evenly, and hopefully can increase accuracy predictions for my ARIMA model. Seen below is the code utilized for picking the 5 random stocks. Note: this file has been ran several times and the 5 outputted values do not reflect the 5 choices that were made for my specific choices.

```python
#Selecting a specific column, in this case it is the Symbol (Or company name) column.
selected_column = 'Sym'
column_data = data[selected_column]

#Pick a random cell from the Symbol column and returning 5 random selections
#Code has been re-ran more than once for stocks that don't have data from 2018-2023
random_values = [random.choice(column_data) for _ in range(5)]
print(f"5 random values in column '{selected_column}': {random_values}")
```

Python

```
5 random values in column 'Sym': ['BJ', 'FRLN', 'OSW', 'LAUR', 'BSMX']
```

I chose my data in such a way that I was able to get my information directly from the source, or in this case, public record. 10 stocks over the course of 5 years provides enough diversity for stocks and time span for prediction purposes. The merged_data file that I created was only made to pick out my stock selections, and was only focused on the first column. I will elaborate on what I did from there and how my new data looks, and how I

went about understanding what I was dealing with and what I wanted to understand more of. Let's take a further dive into the new data set.

Section B: Data Set Description

After figuring out what 10 stocks I will be working with, the next thing to do was to go onto yahoo finance and download the historical data for each stock (https://finance.yahoo.com/quote/MSFT/history?p=MSFT). The time period I will be working with is 2018-2023. After downloading each stock's historical data, I had to insert a column and fill that column with the ticker symbol for the company, I then repeated this process for each company. I repeated the same process as in the introduction and appended all 10 stocks, created a new updated merged file, and got to work on my analysis. What this file looks like is exactly the same as what you can find on Yahoo Finance; It contains open, high, low, close, adjusted close, and volume. Because I am pulling this data directly from Yahoo Finance, there is no missing data (as I expected). However, the data I am working with needs some context so that it can be understood what I am dealing with.

This dataset contains three data types: nominal, interval, and ratio. Nominal data (such as the stock symbol) represents variables that don't have a quantitative value, this is data that can only be categorized. Interval data (like dates) are some-what similar, with the differentiating factor being that the difference in date is important for our analysis. Lastly, the majority of the data is ratio data. Ratio data is a type of data that measures the variables on a continuous scale. In other words, the stock market (even when closed) is still always moving in some direction. So all of our ratios data is Open Price, Low Price, High (Price), Adjusted Close, and Volume. Of each our ratios, the range of values is spread fairly wide, though this is something to be expected. Here is the table of outputs for the data set description. This table contains information for all stock ratios in the data, but doesn't tell me a plethora of what I want to know. As you can see, the range of values contains data for all 10 stocks, and I'll need to look into the statistics to get better, specific questions answered.

	Name	Data Type	Range of Values
0	Sym	nominal	CBSH – TSLA
1	Date	interval	2018-01-02 00:00:00 – 2023-12-29 00:00:00
2	Open	ratio	4.14 – 479.220001
3	High	ratio	4.43 – 479.980011
4	Low	ratio	3.9 – 476.26001
5	Close	ratio	4.13 – 477.709991
6	Adj Close	ratio	4.13 – 476.690002
7	Volume	ratio	1200 – 914082000

Section C: Data Set Summary Statistics

The statistics for this dataset provide more contest that allow the visualization process to have a bit more direction. Overall, I am most interested in the Adjusted Closing Prices because this is what I aim to predict in my Machine Learning model(s). The first thing I wanted to get an insight on was the average adjusted close prices for my stocks. In the event I need to impute any of my data, knowing the average could come in handy, here is the output for the averages:

```
Average Adjusted Closing Prices | 2018-2023:
Sym
CBSH     56.848522
CSTM     13.282863
IWM     169.071485
META    225.102942
MSFT    209.051196
PWR      86.528509
QQQ     264.619278
SPY     342.705646
SRE      56.902392
TSLA    145.981504
Name: Adj Close, dtype: float64
```

After finding the averages, the next thing I wanted more insight on was the percent in adjusted closing price for each of my stocks. This fluctuation gives me an initial understanding of potential volatility and serves as a framework for how difficult or how easy it could be to accurately predict the closing prices for 2023, here is the percent change in adjusted closing price:

```
% Change in Adjusted Closing Price | 2018-2023
Sym
CBSH      21.964637
CSTM      74.323135
IWM       40.996437
META      95.105278
MSFT     368.708310
PWR      459.284878
QQQ      169.292124
SPY       95.542658
SRE      105.335787
TSLA    1062.823981
Name: Adj Close, dtype: float64
```

Looking at this output gives solid direction for where I want to take my analysis. Tesla had the overall largest amount of change through this time span. When I noticed this, it almost immediately made me think about when price changes happen and the relevant prices that follow. Thinking about this from a financial/economics standpoint gave the perspective that when the highest adjusted close occurred may have been a result of a

potential over-valuation. Looking back years later, Tesla was considered to be over-valued by many and as the price of the stock increased, the number of investors increased as well. Logically, the next time I wanted to find out was when the highest adjusted closing price occurred, here is what I found:

```
Stock: SPY, Date: 2023-12-28 00:00:00, Highest Adjusted Close Price: $476.690002
Stock: QQQ, Date: 2023-12-27 00:00:00, Highest Adjusted Close Price: $411.5
Stock: TSLA, Date: 2021-11-04 00:00:00, Highest Adjusted Close Price: $409.970001
Stock: IWM, Date: 2021-11-08 00:00:00, Highest Adjusted Close Price: $234.878174
Stock: MSFT, Date: 2023-11-28 00:00:00, Highest Adjusted Close Price: $382.700012
Stock: SRE, Date: 2022-09-12 00:00:00, Highest Adjusted Close Price: $81.243202
Stock: META, Date: 2021-09-07 00:00:00, Highest Adjusted Close Price: $382.179993
Stock: CBSH, Date: 2021-05-06 00:00:00, Highest Adjusted Close Price: $73.419411
Stock: CSTM, Date: 2021-09-02 00:00:00, Highest Adjusted Close Price: $21.25
Stock: PWR, Date: 2023-12-19 00:00:00, Highest Adjusted Close Price: $217.619492
```

At this point during the analysis, I noticed that around half of the stocks saw their highest adjusted closing price during the Covid-19 pandemic, though why this was is kind of up to interpretation. My theory is that people just had more time to look into stocks, investing, and apps like robinhood gave ease of use to invest, but this is just speculation. This output gave some insight into when these max prices occurred, and my speculation lead me to wanting to see some price changes for 2020-2022, or the bulk of the period we experienced in the pandemic. Out of these outputs, I was simply curious about Tesla and Microsoft. What I did was I looked at the adjusted close price for 2020-2022, and subtracted minimum value from the maximum value. This output doesn't give a ton of overall information, but it shows how much Tesla and Microsoft's price fluctuated during the pandemic:

```
Date Range: 2020-2022
Overall Change in Adjusted Closing Price (in $) for Tesla: 385.88866800000005
Overall Change in Adjusted Closing Price (in $) for Microsoft: 206.0849
```

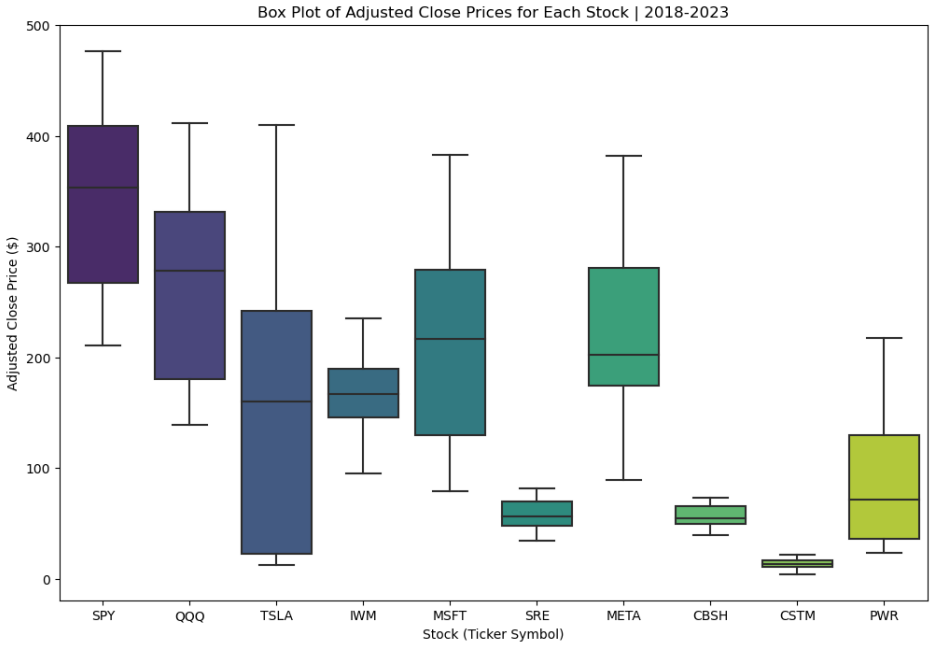
It's good to see this statistic, however I want to calculate standard deviation because it measures volatility much more accurately, and will give an idea for how much prices are changing from the average. Here is the results of that. What this output means is that on average, between 2018 and 2023, Tesla's adjusted closing price changed on average around 4% per day, whereas Microsoft's was only 1.89%. This isn't what I'd consider an extreme amount of volatility, but it's an interesting statistic to investigate. The last thing in my Data Set Summary Statistics that I want to know about is the annual volatility. Basically, I don't need to calculate for total trading days (as I originally expected), I can simply get the total returns for Adjusted Close and find the standard deviation. Volatility is based off of the change in adjusted close from one day to the next, and luckily this can be quickly calculated in Python. I can multiply that value by 100 to get the overall percentage volatility for our date range of 2018-2023:

```
Date Range: 2018-2023
Sym
CBSH    1.848128
CSTM    3.743258
IWM     1.604547
META    2.693981
MSFT    1.897623
PWR     2.147449
QQQ     1.575011
SPY     1.284364
SRE     1.696313
TSLA    4.016929
```

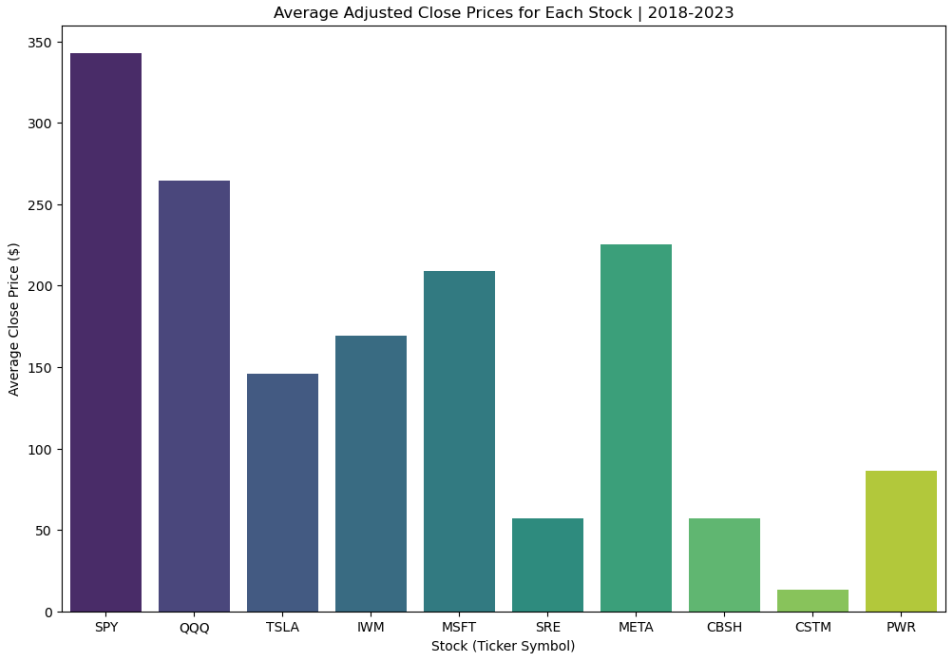
Overall, I'm not entirely sure about how the volatility will factor in once I am trying to make my predictions. While the volatility is important information to have, it may not be entirely useful in all situations. Knowing these values is interesting and provides and insight for what may be smarter investments over this period. In terms of volatility, Tesla is showing pretty moderate levels of risk, it's basically saying that the price fluctuated almost double, sometimes 2.5 times more than the other stocks in the portfolio. I think that overall, Tesla (and CSTM) could be difficult to predict accurately, but calculating the risk may be a necessary part of the predictive analytics process. A thing to note about the above output is that though the range is for 2018-2023, it is an annual calculation, not cumulative. Technically, a cumulative calculation for volatility would not be as reliable and would not serve as beneficial compared to annual returns of my adjusted close price(s).

Part D: Data Set Graphical Exploration

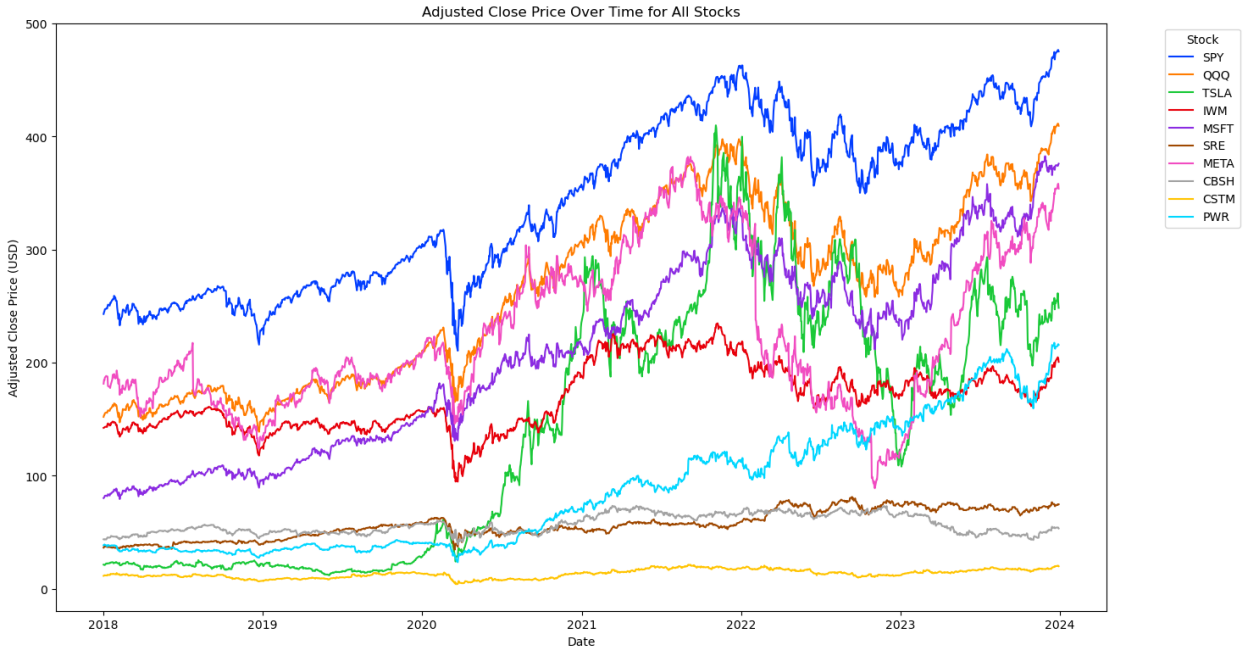
For me, the graphical exploration is much more intuitive and fun, though I wanted to keep my overall visualizations fairly minimal. The first thing that I wanted to get an idea of was a visualization for the overall adjusted closing prices. Similar to my averages in my Summary Statistics, this boxplot is a nice way to show what I can expect in terms of adjusted closing prices for 2018-2023:



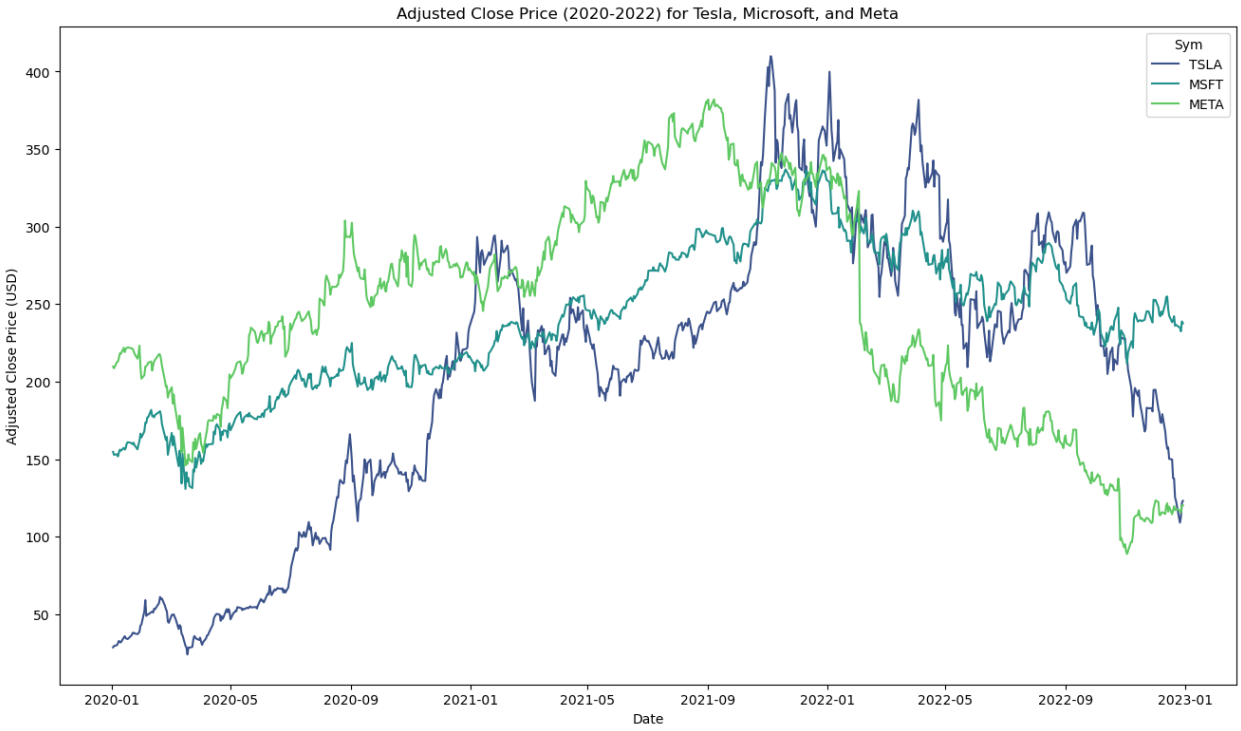
Here is a barplot for the average adjusted closing price, the information in the box plot and bar plot are not new information. However graphically it's interesting to see, what I am more concerned with is the line graph.



This lineplot for the adjusted closing prices is what I am particularly interested in. Over time, the overall fluctuations in these stocks are are very interesting to see. Most of the stocks in the stock market took a hit when Covid happened, and in the time after, most stocks saw a very large return in overall market value and stock price. What is interesting is that around 2022, some stocks experienced a slight drip in price, and look to recover around 2023. From this graph, there are a couple things to note; CSTM(Constellium), CBSH (Commerce Bancshares), and SRE (Sempra) seem to be the most consistent in overall fluctuation. PWR (Quanta Services) looks to be the most notable stock for consistent positive trend over time. I believe that because of the business of this company probably served to be very profitable over the pandemic. PWR (Quanta Services) does infrastructure for networks, electric power, and communication. This business' overall adjusted close price could be (theoretically) easier to predict, but that is just my initial thought. Other companies like TSLA (Tesla), MSFT (Microsoft), and META (Facebook/Meta) can see more wild fluctuation in price, and as a result may be more difficult to accurately predict their adjusted closing prices.



I wanted to get a closer look at Telsa, Microsoft, and Meta/Facebook. I separated these stocks from the group and looked at their overall price fluctuation for 2020-2022, when the pandemic was in full force and their prices seemed to reflect the uncertainty of the economy. One of the things that I am considering looking into for my model building process is to see if there are any other ways to forecast future trends. With price fluctuations this up and down, finding methods and models to train and predict accurately will certainly be a difficult process.



The one thing that I did not add as a visualization to this section is a plot of the volume data. The overall volume fluctuates so wildly day-to-day that plotting this data did not yield any sufficient realizations. In fact, plotting volume in any way was very difficult and did not make me feel like I was uncovering any relevant information as a result. Moving forward, I want to find out more about volume and if there is any connection that may be worthwhile for predictions.

Part E: Summary of Findings

Overall, this analysis resulted in several interesting discoveries about my data and the process for which I must go about understanding my variables, equations, and comparisons. Since I have a balance in this portfolio between the top 5 holdings and 5 random choices, I think that predicting adjusted closing price could be done in two parts. I could build my model and train it for the less volatile stocks and find varying ways to adjust my

approaches to the more volatile stocks like Tesla and Microsoft. I don't think that dropping the Volume column in my dataset is entirely necessary, but I may have to find a way to deal with the volume data like aggregating. I am also considering working with volume and seeing if there could be a relationship between volume and percent change in price. The benefit to my thinking as a result of this analysis is that I have a better understanding of what my data looks like, but it also gives me a framework for how I want to go about structing and tailoring my model for training and testing. However, this process will be a lot of trial and error, and methodologies are subject to change on an as-needed basis. This analysis was very useful though in structuring my curiosities, though.

Another potential issue that I am wondering about is the overall accuracy of predicting adjusted closing price for the stocks that are very volatile. My initial thought is to normalize the data to make the overall distribution of the ratios more normal. I think that normalizing may be a good place to look and assess the accuracy of the predictive components of this and try it on my un-normalized data too. Being able to understand what is going on in my data on a statistical and graphical level helps tailor my assumptions and points my general questions in more specific directions. I think that I have a much more solid foundation to operate on from this analysis, building my initial ARIMA model will be challenging, but I will be able to tailor my variables as needed thanks to what I have discovered in this process. Moving forward, I am going to pay attention diversifying my modeling to account for normalization, aggregation, and potential seasonality. The more evenly distributed I can get my data to be, the easier my model will be to train at first, and I can stem off that to diversify aspects of the model as needed.