Options premium shorting; A statistical and Historical Analysis On Weekly Options Contracts

By: Parker Christenson

“In the Animal Kingdom the rule is,
eat or be eaten, in the human kingdom,
define or be defined.”

- Thomas Szasz

DISCLAIMER

Parker Christenson provides this paper for informational purposes only. Use of and access to the Site and the information, materials, services, and other content available on or through the Site ("Content") are subject to these terms of use and all applicable laws.

INVESTMENT RISKS

There are risks associated with investing in securities. Investing in stocks, bonds, exchange traded funds, mutual funds, and money market funds involve risk of loss. Loss of principal is possible. Some high risk investments may use leverage, which will accentuate gains & losses. Foreign investing involves special risks, including a greater volatility and political, economic and currency risks and differences in accounting methods. A security's or a firm's past investment performance is not a guarantee or predictor of future investment performance

THIRD PARTY LINKED SITES

As a convenience to you, this paper may provide hyperlinks to web sites operated by third parties. Because Parker Christenson has no control over such sites or their content, Parker Christenson is not responsible for the availability of such external sites or their content, and Parker Christenson does not adopt, endorse or nor is responsible or liable for any such sites or content, including advertising, products or other materials, on or available through such sites or resources. Other web sites may provide links to the Site or Content with or without our authorization. Parker Christenson does not endorse such sites and shall not be responsible or liable for any links from those sites to the Site or Content, or for any content, advertising, products or other materials available on or through such other sites, or any loss or damages incurred in connection therewith.

YOUR USE OF THIRD-PARTY WEB SITES AND CONTENT, INCLUDING WITHOUT LIMITATION, YOUR USE OF ANY INFORMATION, DATA, ADVERTISING, PRODUCTS, OR OTHER MATERIALS ON OR AVAILABLE THROUGH SUCH WEB SITES, IS AT YOUR OWN RISK AND IS SUBJECT TO THEIR TERMS OF USE.

About Me and my “Day Trading Process”

I am Parker Christenson. I’ve been day trading since late 2018, and I thought to myself when I started this journey, “It’s just a click here and click there, boom profit. How Hard could this be?”. I was 100% wrong. As I funded my first account, I quickly realized the Click-Click mentality, was going to ruin me in the long run. I took 4 months after losing my ass on my first account, and just studied the financial markets. When studying the market, I realized that the market I was attempting to trade was so volatile, and so hard to make money in the long run being a bull. Around two years ago, 2021 peak Covid-19 I did a deep dive into market manipulation and the process, and forms that banks file, to ‘screw’ over the investor who thinks the stock will gain in value. I Changed my thought process and looked deep into market manipulation. I took the red pill. Nothing is what it seems to be and there is always a hidden agenda. Looking into my own personality, I realize why I started this journey, which is ending me up talking to you, one of a few readers who will ever get their hands on this analysis paper un-censored. I hate being wrong. I cannot accept being wrong, nor will I bet against something that has a higher odd of me loosing. I don’t gamble, I hate the Idea of gambling. I used to think that I was an outlier in a data set and the mean and average never applied to me. As I came to realize it, statistics, are almost never wrong. One can only improve their odds to become different, become the outlier, excluded from the data set. Over the past two months, December 2022, up until January 2023, I have almost mastered the most basic coding language, (python) all in the efforts of wanting to be right in the financial markets. If you want to be any sort of day-trader, I think that you need this tool in your tool kit to make it.

The Cold Honest Truth

Your favorite Wallstreet-Bets moderator, Day-trading YouTuber, Stock Guru, Twitter-chart-poster, day-trading-Course-Seller will not tell you this: Market manipulation is present. The bigger fish will see your order flow and will end up taking your money in hopes of getting rich. They will take your money with no questions asked and no remorse. The Tactics they use, are some of the slimiest and grossest in the world. To make it in the financial markets, you need to play on the side of the big fish and be predictive rather than reactionary. **Eat or get ate.**

Introduction:

Options pricing and their structure is somewhat of a complex task in general to sort and find an extremely probable outcome, to securely achieve consistent profits. There are some points I would like to make. To start off, upwards of nighty-percent of options expire worthless. Therefore in theory, we can make plays and make money off of the majority of stock options expiring worthless.

Main Thesis:

My main thesis is that the banks are pulling stock prices up to bring in more buyers over a given price, inducing FOMO (Fear of Missing out) in either a bull market or a bear market by having the opposite player pile into one option, and pulling the option the opposite way. We can use this theory to sell options at a higher premium and buy back the barrowed options at a lower price.

Explanation of this paper and Its uses for the general person:

This paper will show a couple of things and the basics of shorting options for a profit. On the way I will describe the ways that options pricing move, and the advantages of shorting options will have as compared to stock pricing on the account. In this report I will show the standard pricing of the option, and translate that into showing how much money, a potential option can make you, along with showing some of the data analysis process. This report is going to be as transparent as possible, and I would like to implement the data set into an options premium shorting dashboard, that lets the user know the best option to short for the week. If that option shorting dashboard become available to select users, It will show the option to short only for one or two companies.

Basic Terms Used in this paper

Bullish- to think that the market will continue to go higher.

Bearish- to think that the market will continue to go lower.

Long- to buy and hold and wait for a higher point to sell.

Short- to sell, then buy back the market derivative for a lower price, pocketing the difference.

Strike Price- Price that the contracts become valid at when in the money.

Bagged- to be in a position of unimaginable loss, at the hands of the market and the market makers.

Open interest- the number of open contracts sitting at a given strike price, on one side of the money.

Options Basics

This next section will go somewhat quickly but is a needed foundation if you want to follow this report on shorting, I will not be going in depth, but I will be going over the fundamentals of what each type of type option is and the general option pricing.

What is a stock option?

A stock Option is a contract to buy or sell 100 shares of an underlying security at a given set price point. Which is determined by the Strike price. A stock option has the possibility to end up worthless which as mentioned earlier, nighty percent of the options do.

Types of options

Call- A call when bought is a bet that the underlying stock price will end up above the strike price.

PUT- A put when bought is a bet that the underlying stock price will end up below the strike price.

Types of Actions with Options

<u>BUY A CALL</u> - A call when bought is a bet that the underlying stock price will end up above the strike price. (BULLISH)

<u>BUY A PUT</u>- A put when bought is a bet that the underlying stock price will end up below the strike price. (BEARISH)

<u>SELL A CALL</u>- A call when bought is a bet that the underlying stock price will end up above the strike price. (BEARISH)

<u>SELL A PUT</u> - A put when bought is a bet that the underlying stock price will end up below the strike price. (BULLISH)

Option Pricing Structure

When buying an option, you are paying 100 times the options price times the Number of contracts. The full Equation is below, CA being the number of contracts and the P being the Price you buy on.

P x 100 x CA = <u>Total Cost of Contracts</u>

Example of Pricing structure

Tesla Is trading at $250, and I think the Stock will end up above $260. I am going to buy the $260 call, 10 contracts all trading at $2.25. The equation that follows will be how to price the option and your total cost on how much that bet will cost.

$2.25 x 100 x 10 = <u>$2,500 Total Cost</u>

Calculating Options Premium Profits

Using a similar structure to calculate the cost of the options contracts, this is how we calculate the total profit
Ep = Entry price
EXp = Exit price
CA= Contract Amount
U= Underlying amount of shares held by contracts
P= Profit

((Ep-EXp) x CA) x U = P

The Tesla $250.00 Call is trading at $25.00, and I think that Tesla stock will go below $250.00. It is currently at $255.00. I'm going to sell this call and cover when the option contract is only worth $5.00. I'm going to short sell 10 contracts.

(($25.00-5.00) x 10) x 100 = <u>$20,000.00 Profit</u>

Strategy and continuation of Thesis

Continuation of the Thesis

When the banks are writing the contracts, the person who is buying the options contract is buying at whats called a premium. A premium is the $2.25 charge that the options writer is charging you, to buy the options contract. In some cases, the options contracts that we will be betting against will have 30,000 in open interest. Open interest is just a way of indicating the way in which the majority of the market participants will have the market above or below. By using open interest we can bet against the 90% of options, and end up pocketing the massive difference in the options premium being bought and sold. With out getting to in depth, if an option is trading on the bid at $10.75, and the likely hood that the option will expire worthless, thus meaning that the option pricing on the bid will come back down to approx. $1.29, why not take the ~95% return on the initial bet?

What is the basics of the strategy?

We are going to look for the highest options premium to short and bet against the most people. As it would make sense that nighty-percent of the options would expire worthless. People buy into the the option making it shown on the "open interest". When looking at the highest open interest contract, we can easily bet against the largest stack of people all betting the stock would end up on Friday at a certain price point.

Simple explanation on the strategy:

Monday- looking at the swift reaction of the market and looking at the price action of the market

Tuesday- Looking for the continuation of Mondays reactional direction.

Wednesday- Looking for a general chop or a flat day, End up less than a percent up or down, locating the highest open interest option to short

Thursday- Entering the trade and Possibly closing out the trade if the options premium hits the specified price point

Friday- Mandatory closing out the trade and covering all sold options contracts.

Getting into data collection

Orats API

The hardest part of finding the data needed and the data that we will be acquiring is the “open interest”. There are very little of companies that actively Track open interest, making it very hard to find a reliable source. The source I decided to use Was called ORATS.com which allow for their data to be downloaded as a JSON output from their API. In layman's terms I am able to download their data from their server showing all of the options chain’s information based off of the market close data. I think that this was the hardest data source to find, because without the creation of a massive data pipeline, It makes the historical downloading of data extremely efficient, and only takes one request per date, per company.

Polygon.io API

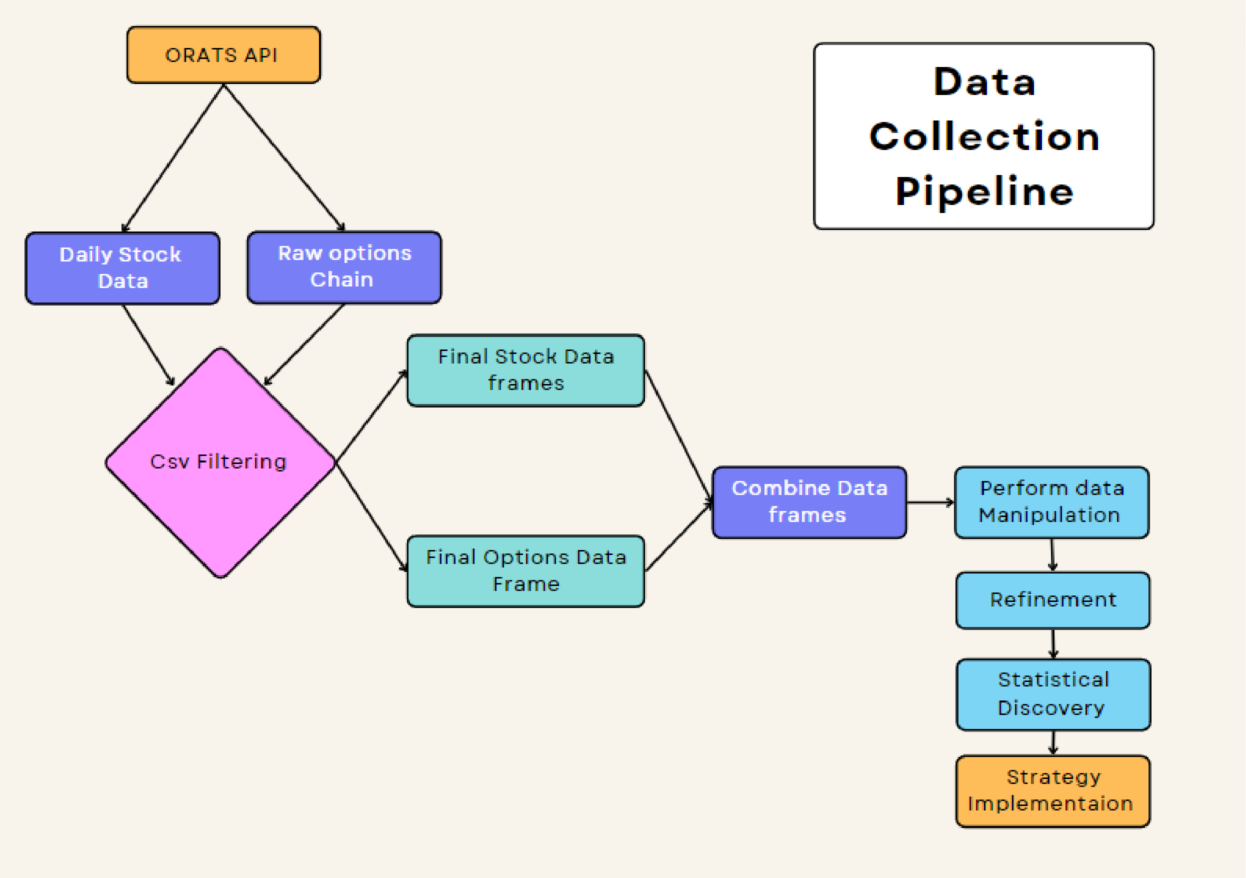
The API offered by polygon.io is a very good beginner API and is simple to use. When looking at the docs and using the API, the data requested gets returned in a JSON output, and makes the data collection process and sorting easy. I used Polygon.io for Stock pricing, and the Option pricing, as they made the use of their API very easy.

Problems with the Data and API collection

I think that the hardest part with the data and the API collections process was the fact that both data providers both collect and store their data differently. I would have loved to use one API for the entire report, But ORATS and Polygon were both very limited in the responses and the endpoints that the offer. Below will be outputs of both APIS and will be explained in simple terms. There is no uniform data collection standard, and there is no way that the SEC and federal government will regulate the data collection of financial securities. The only way that an individual could have gotten around this project would have been by using a broker API, which would have been pricey, and unreasonable for the overall cost of this project. In a 6-day time span I ran over 60,000 API requests which was roughly returned 47 million different data points. Over 1.2 terabytes of data in CSV files. I will show very small amounts of data and code, throughout this project.

The scope of this strategy will go back and look at approximately 3 years to January 2nd, 2019, all the way up to December 7th, 2022. The dates were chosen as this is when major volatility was introduced into the market and the major price swings have been introduced economically, almost making it easier for the market maker and the options writer to hide their actions, within the price action of the day.

Outline of the data collection pipeline and filtering



This is a very rough outline of the way data was collected and transformed into the final statistical analysis at the end of this paper. Understanding how the numbers came to be and understanding why the data we collected matters, along with the process, will be explained in the upcoming pages as well.

Here is a brief number summary

1. Orats json output
2. Filtering the CSVs to get the Weekly expiry
3. Filtering the CSVs to get only In the Money Options
4. Filtering CSVs to get the highest open interest contracts
5. Separating the calls and puts (direction of returns inversed for Puts)
6. Mathematical calculations
7. Plotting and graphing calculated returns
8. Graphing and plotting the Options charts
9. Plotting the stock and option Data

Quick Statistics:

- The option that we are betting against has a 51.8058% chance likelihood of ending up out of the money, thus making the trading strategy profitable given the 2 risk to one reward method.
- 1295 of the options we bet against were calls. Shorting only the highest interest calls yields a 55.4408%-win rate
- 513 options we shorted against were puts. Shorting only the highest interest puts yields a 44.8849%-win rate.

So what can we do to get the win rate higher??

The initial thought behind the project was that the options would expire out of the money, by Thursday or Friday at market close. So doing some calculations we can figure out that **61%** of all options we tracked, go out of the money, either on Thursday or Friday at some point.

What can we do to make that win rate percentage even better?

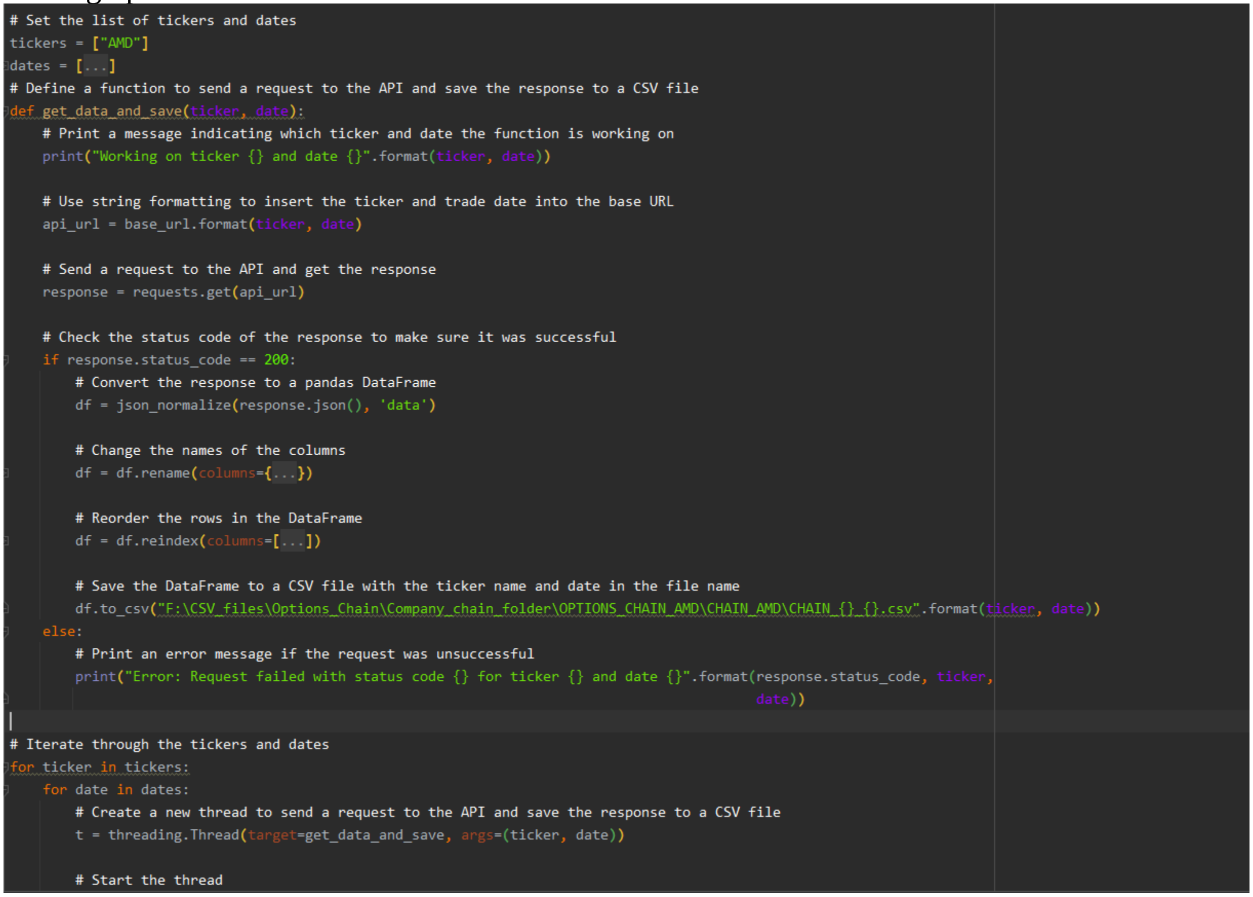
We can look at the percent change from Monday to Tuesday. We are using some trading psychology here and doing the exact opposite of "dumb money" and playing with that casino.

The thought process behind the trader is reactionary and is not thinking more than 1 or two steps ahead. Therefore the regular retail trader will end up piling into one option, going in the direction of the market, thinking the market will continue to go down or up.

After Filtering the stock data, to show the Monday to Wednesday % change being over 3% or Less than -3% we get the win rate up to 71%.

Small Pieces of Code

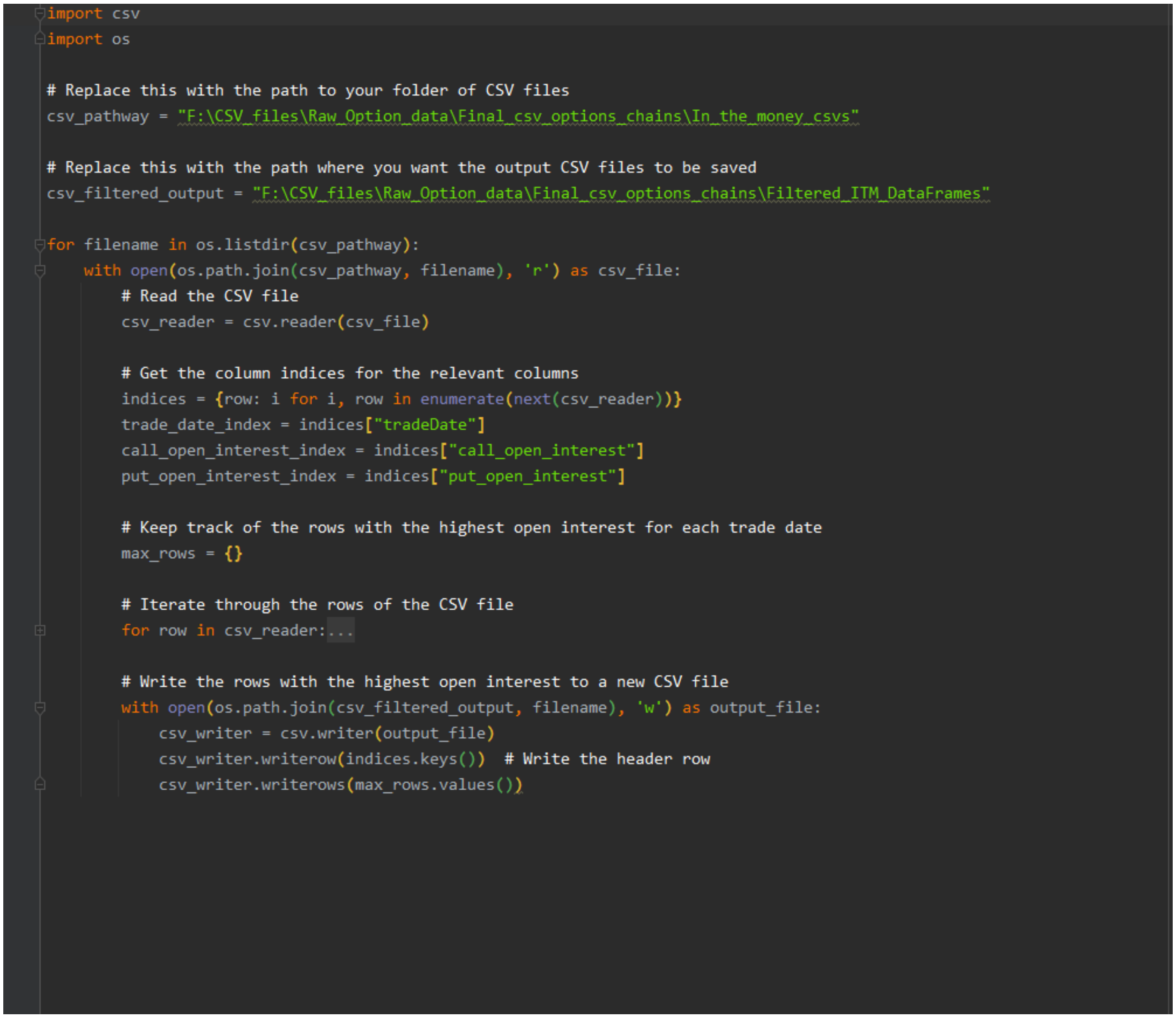
Getting options Chains from Orats.com



```python
# Set the list of tickers and dates
tickers = ["AMD"]
dates = [...]
# Define a function to send a request to the API and save the response to a CSV file
def get_data_and_save(ticker, date):
    # Print a message indicating which ticker and date the function is working on
    print("Working on ticker {} and date {}".format(ticker, date))

    # Use string formatting to insert the ticker and trade date into the base URL
    api_url = base_url.format(ticker, date)

    # Send a request to the API and get the response
    response = requests.get(api_url)

    # Check the status code of the response to make sure it was successful
    if response.status_code == 200:
        # Convert the response to a pandas DataFrame
        df = json_normalize(response.json(), 'data')

        # Change the names of the columns
        df = df.rename(columns={...})

        # Reorder the rows in the DataFrame
        df = df.reindex(columns=[...])

        # Save the DataFrame to a CSV file with the ticker name and date in the file name
        df.to_csv("F:\CSV_files\Options_Chain\Company_chain_folder\OPTIONS_CHAIN_AMD\CHAIN_AMD\CHAIN_{}_{}.csv".format(ticker, date))
    else:
        # Print an error message if the request was unsuccessful
        print("Error: Request failed with status code {} for ticker {} and date {}".format(response.status_code, ticker,
                                                                                            date))

# Iterate through the tickers and dates
for ticker in tickers:
    for date in dates:
        # Create a new thread to send a request to the API and save the response to a CSV file
        t = threading.Thread(target=get_data_and_save, args=(ticker, date))

        # Start the thread
```

Getting the initial options chains would have been painstakingly hard to do if I did not use threading. Since every single options chain has a date attached to the name of the csv file as well, I could use threading to speed the process up overall. While using ORATS.com for their Options historical data Api, I got an email from them because they saw the extremely high volume amount of API calls, and asked if I would like to upgrade. I politely declined. I made approximately 1,000,000 API calls using this code in a two day time period. Making sure just that the files and everything saved properly, and all of the data was formatted in the correct format.

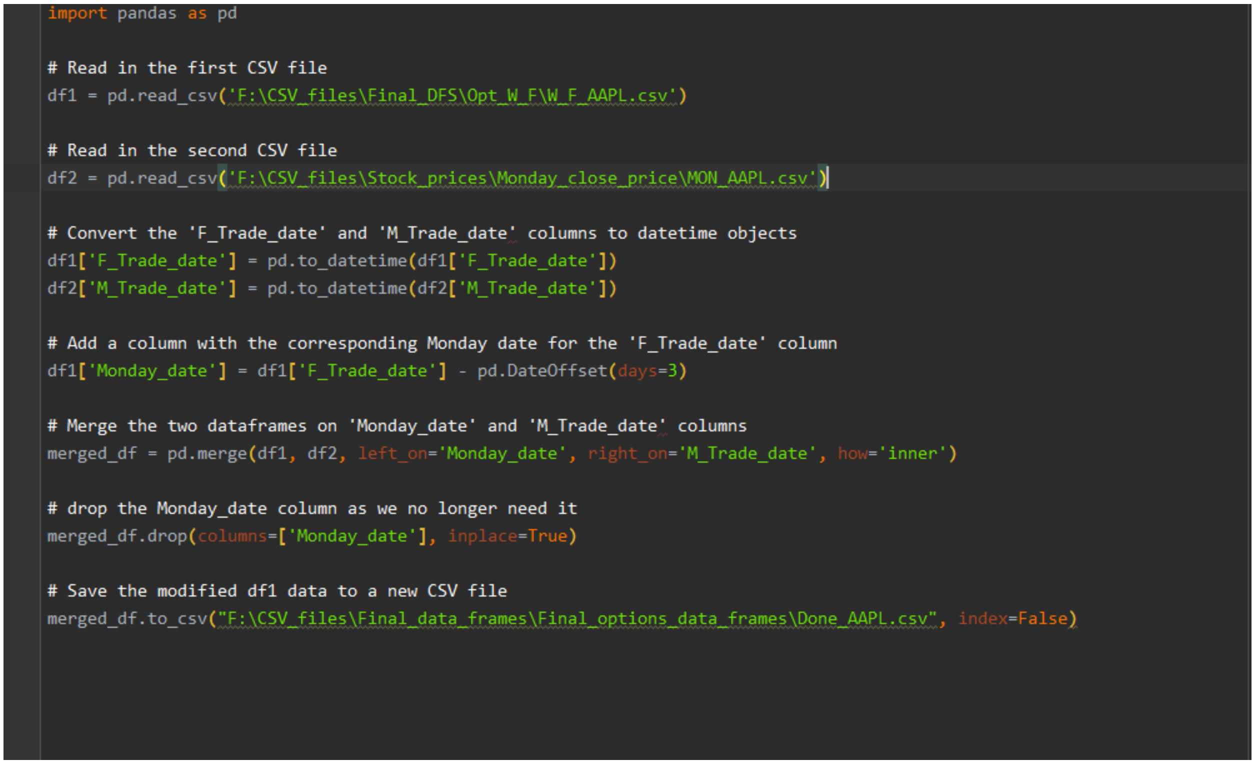
Filtering the Highest In the money options



```
import csv
import os

# Replace this with the path to your folder of CSV files
csv_pathway = "F:\CSV_files\Raw_Option_data\Final_csv_options_chains\In_the_money_csvs"

# Replace this with the path where you want the output CSV files to be saved
csv_filtered_output = "F:\CSV_files\Raw_Option_data\Final_csv_options_chains\Filtered_ITM_DataFrames"

for filename in os.listdir(csv_pathway):
    with open(os.path.join(csv_pathway, filename), 'r') as csv_file:
        # Read the CSV file
        csv_reader = csv.reader(csv_file)

        # Get the column indices for the relevant columns
        indices = {row: i for i, row in enumerate(next(csv_reader))}
        trade_date_index = indices["tradeDate"]
        call_open_interest_index = indices["call_open_interest"]
        put_open_interest_index = indices["put_open_interest"]

        # Keep track of the rows with the highest open interest for each trade date
        max_rows = {}

        # Iterate through the rows of the CSV file
        for row in csv_reader:...

        # Write the rows with the highest open interest to a new CSV file
        with open(os.path.join(csv_filtered_output, filename), 'w') as output_file:
            csv_writer = csv.writer(output_file)
            csv_writer.writerow(indices.keys())  # Write the header row
            csv_writer.writerows(max_rows.values())
```

This was my code for filtering all of the csv files, and getting the row with the highest open inrest to be sent to a final data frame. At this point I wanted to see the average win rate of the strategy and see what percentage of options ended up out of the money on either Thursday or Friday.

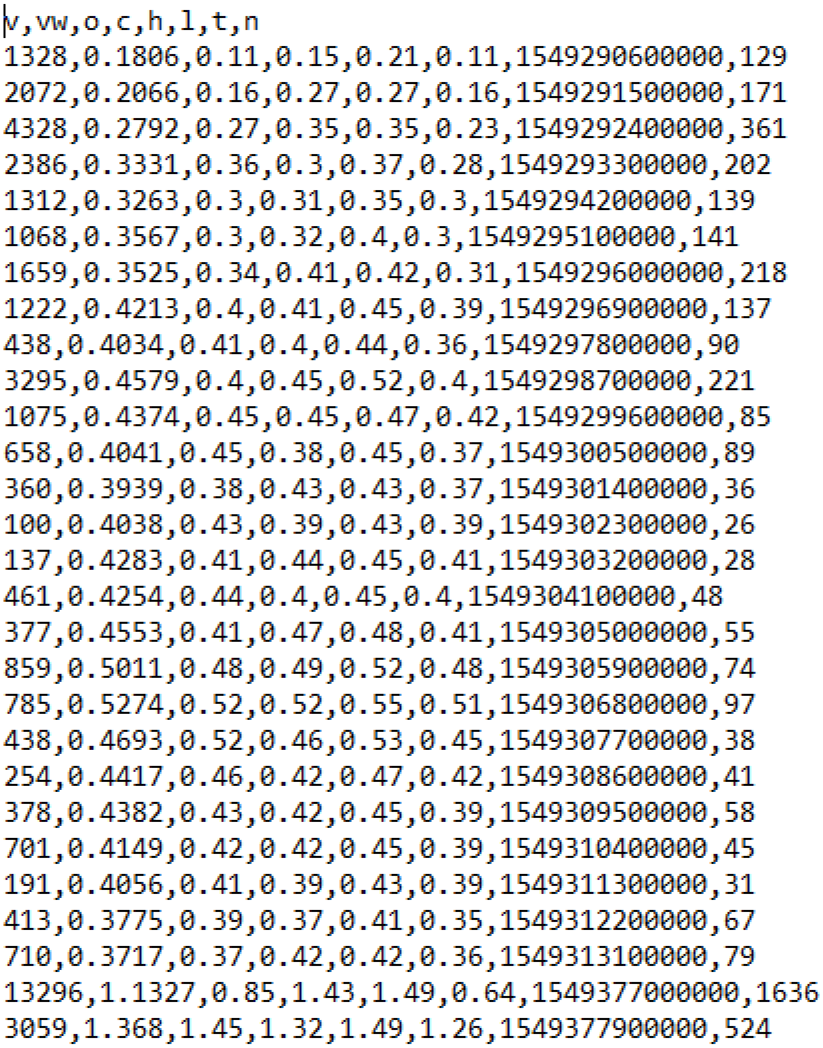
Getting stock data for first data frame to filter and narrow the strategy down to specifics.



```
import pandas as pd

# Read in the first CSV file
df1 = pd.read_csv('F:\CSV_files\Final_DFS\Opt_W_F\W_F_AAPL.csv')

# Read in the second CSV file
df2 = pd.read_csv('F:\CSV_files\Stock_prices\Monday_close_price\MON_AAPL.csv')

# Convert the 'F_Trade_date' and 'M_Trade_date' columns to datetime objects
df1['F_Trade_date'] = pd.to_datetime(df1['F_Trade_date'])
df2['M_Trade_date'] = pd.to_datetime(df2['M_Trade_date'])

# Add a column with the corresponding Monday date for the 'F_Trade_date' column
df1['Monday_date'] = df1['F_Trade_date'] - pd.DateOffset(days=3)

# Merge the two dataframes on 'Monday_date' and 'M_Trade_date' columns
merged_df = pd.merge(df1, df2, left_on='Monday_date', right_on='M_Trade_date', how='inner')

# drop the Monday_date column as we no longer need it
merged_df.drop(columns=['Monday_date'], inplace=True)

# Save the modified df1 data to a new CSV file
merged_df.to_csv("F:\CSV_files\Final_data_frames\Final_options_data_frames\Done_AAPL.csv", index=False)
```

This is me combining two files, rather than thousands of files, because I wanted to understand how to minipulate the data a little bit better before combining thousands of rows from thousands of csv files. I think that This process of making sure the data was going into the correct rows, and properly being merged was the hardest part of this project.

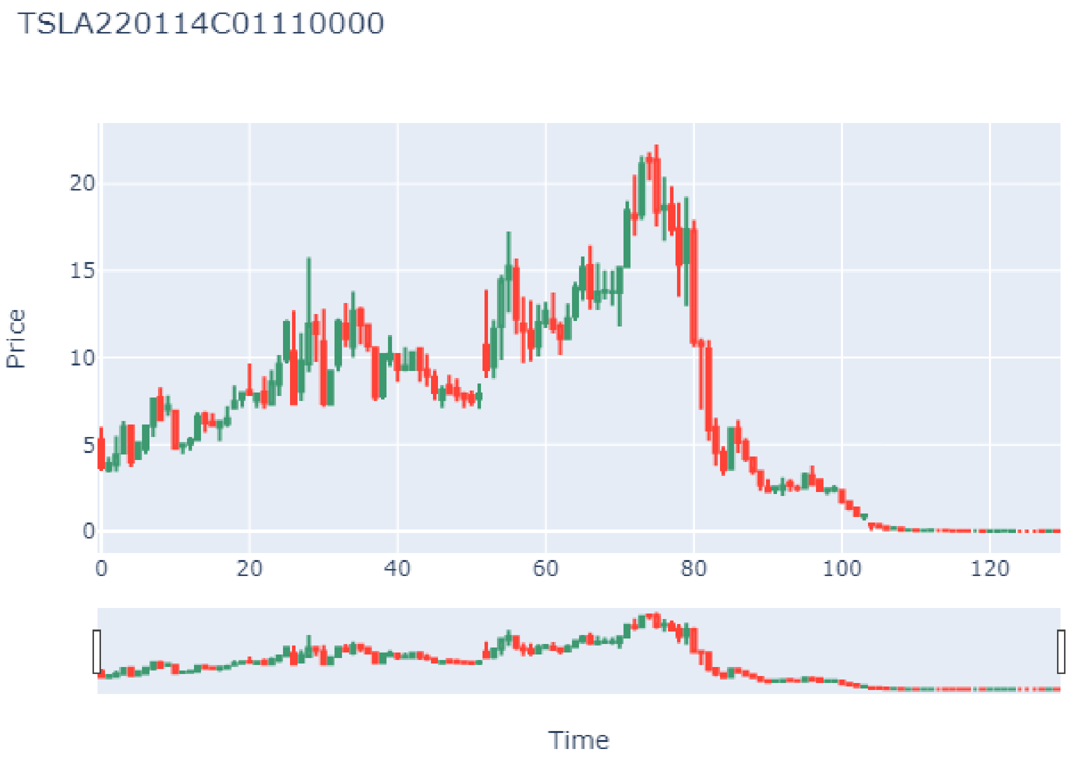
The data frame for API calls to get graphs



```
v,vw,o,c,h,l,t,n
1328,0.1806,0.11,0.15,0.21,0.11,1549290600000,129
2072,0.2066,0.16,0.27,0.27,0.16,1549291500000,171
4328,0.2792,0.27,0.35,0.35,0.23,1549292400000,361
2386,0.3331,0.36,0.3,0.37,0.28,1549293300000,202
1312,0.3263,0.3,0.31,0.35,0.3,1549294200000,139
1068,0.3567,0.3,0.32,0.4,0.3,1549295100000,141
1659,0.3525,0.34,0.41,0.42,0.31,1549296000000,218
1222,0.4213,0.4,0.41,0.45,0.39,1549296900000,137
438,0.4034,0.41,0.4,0.44,0.36,1549297800000,90
3295,0.4579,0.4,0.45,0.52,0.4,1549298700000,221
1075,0.4374,0.45,0.45,0.47,0.42,1549299600000,85
658,0.4041,0.45,0.38,0.45,0.37,1549300500000,89
360,0.3939,0.38,0.43,0.43,0.37,1549301400000,36
100,0.4038,0.43,0.39,0.43,0.39,1549302300000,26
137,0.4283,0.41,0.44,0.45,0.41,1549303200000,28
461,0.4254,0.44,0.4,0.45,0.4,1549304100000,48
377,0.4553,0.41,0.47,0.48,0.41,1549305000000,55
859,0.5011,0.48,0.49,0.52,0.48,1549305900000,74
785,0.5274,0.52,0.52,0.55,0.51,1549306800000,97
438,0.4693,0.52,0.46,0.53,0.45,1549307700000,38
254,0.4417,0.46,0.42,0.47,0.42,1549308600000,41
378,0.4382,0.43,0.42,0.45,0.39,1549309500000,58
701,0.4149,0.42,0.42,0.45,0.39,1549310400000,45
191,0.4056,0.41,0.39,0.43,0.39,1549311300000,31
413,0.3775,0.39,0.37,0.41,0.35,1549312200000,67
710,0.3717,0.37,0.42,0.42,0.36,1549313100000,79
13296,1.1327,0.85,1.43,1.49,0.64,1549377000000,1636
3059,1.368,1.45,1.32,1.49,1.26,1549377900000,524
```

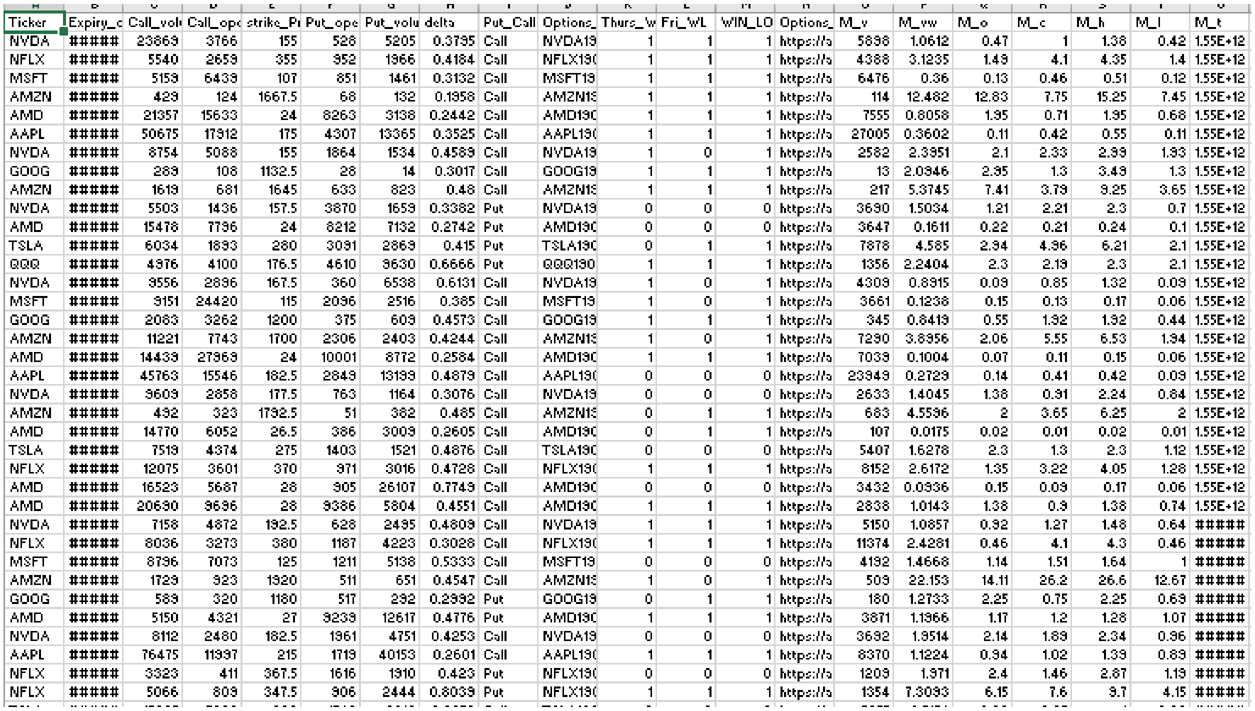
Once the final data frame was made, to see if the options did in fact expire out of the money on Friday, or go out of the money on Thursday and go back into the money on Friday, I was able to make the final data frame using the API urls to and make threading calls, from a csv file. The 'results' were saved as the picture for above. Everything that met the initial criteria of that starting data frame, was saved as follows. I then proceeded to make a data pipeline that took all of these csv files, and made charts out of them. I think that This was the most unique process of the entire project because you got to visualize the price action and price movement.

The clearest chart of market manipulation I have seen



From $23 all the way down to zero on Friday. The Options writers collected so much money in premiums on this one single play, and they straight bagged every single open interest holder, and thousands of people who did not hold the contract over night and bought the Wednesday pop around the 70-75 mark on the Y axis. A reminder that every dollar an option contract goes up or down is $1,000 in options premium gained or lost.

Final data frame for statistical discovery



Ticker	Expiry_c	Call_vol	Call_ope	strike_P	Put_ope	Put_volu	delta	Put_Call	Options_	Thurs_W	Fri_WL	WIN_LO	Options_	M_v	M_vw	M_o	M_c	M_h	M_l	M_t
NVDA	#####	23863	3766	155	528	5205	0.3795	Call	NVDA19	1	1	1	https://a	5898	1.0612	0.47	1	1.38	0.42	1.55E+12
NFLX	#####	5540	2659	355	952	1966	0.4184	Call	NFLX190	1	1	1	https://a	4388	3.1235	1.49	4.1	4.35	1.4	1.55E+12
MSFT	#####	5159	6439	107	851	1461	0.3132	Call	MSFT19	1	1	1	https://a	6476	0.36	0.13	0.46	0.51	0.12	1.55E+12
AMZN	#####	429	124	1667.5	68	132	0.1958	Call	AMZN19	1	1	1	https://a	114	12.482	12.83	7.75	15.25	7.45	1.55E+12
AMD	#####	21357	15633	24	8263	3138	0.2442	Call	AMD190	1	1	1	https://a	7555	0.8058	1.95	0.71	1.95	0.68	1.55E+12
AAPL	#####	50675	17912	175	4307	13365	0.3525	Call	AAPL190	1	1	1	https://a	27005	0.3602	0.11	0.42	0.55	0.11	1.55E+12
NVDA	#####	8754	5088	155	1864	1534	0.4589	Call	NVDA19	1	0	1	https://a	2582	2.3951	2.1	2.33	2.99	1.93	1.55E+12
GOOG	#####	289	108	1132.5	28	14	0.3017	Call	GOOG19	1	1	1	https://a	13	2.0946	2.95	1.3	3.49	1.3	1.55E+12
AMZN	#####	1619	681	1645	633	823	0.48	Call	AMZN19	1	1	1	https://a	217	5.3745	7.41	3.79	9.25	3.65	1.55E+12
NVDA	#####	5503	1436	157.5	3870	1659	0.3382	Put	NVDA19	0	0	0	https://a	3690	1.5034	1.21	2.21	2.3	0.7	1.55E+12
AMD	#####	15476	7796	24	8212	7132	0.2742	Put	AMD190	0	0	0	https://a	3647	0.1611	0.22	0.21	0.24	0.1	1.55E+12
TSLA	#####	6034	1893	280	3091	2869	0.415	Put	TSLA190	0	1	1	https://a	7878	4.585	2.94	4.96	6.21	2.1	1.55E+12
QQQ	#####	4976	4100	176.5	4610	9630	0.6666	Put	QQQ190	1	1	1	https://a	1356	2.2404	2.3	2.19	2.3	2.1	1.55E+12
NVDA	#####	9556	2896	167.5	360	6538	0.6131	Call	NVDA19	1	0	1	https://a	4309	0.8915	0.09	0.85	1.32	0.09	1.55E+12
MSFT	#####	9151	24420	115	2096	2516	0.385	Call	MSFT19	1	0	1	https://a	3661	0.1238	0.15	0.13	0.17	0.06	1.55E+12
GOOG	#####	2083	3262	1200	375	609	0.4573	Call	GOOG19	1	1	1	https://a	345	0.8419	0.55	1.92	1.92	0.44	1.55E+12
AMZN	#####	11221	7743	1700	2306	2403	0.4244	Call	AMZN19	1	0	1	https://a	7290	3.8956	2.06	5.55	6.53	1.94	1.55E+12
AMD	#####	14439	27969	24	10001	8772	0.2584	Call	AMD190	1	1	1	https://a	7039	0.1004	0.07	0.11	0.15	0.06	1.55E+12
AAPL	#####	45763	15546	182.5	2849	13199	0.4879	Call	AAPL190	0	0	0	https://a	23949	0.2729	0.14	0.41	0.42	0.09	1.55E+12
NVDA	#####	9609	2858	177.5	763	1164	0.3076	Call	NVDA19	0	0	0	https://a	2633	1.4045	1.38	0.91	2.24	0.84	1.55E+12
AMZN	#####	492	323	1792.5	51	382	0.485	Call	AMZN19	0	1	1	https://a	683	4.5596	2	3.65	6.25	2	1.55E+12
AMD	#####	14770	6052	26.5	386	3009	0.2605	Call	AMD190	0	1	1	https://a	107	0.0175	0.02	0.01	0.02	0.01	1.55E+12
TSLA	#####	7519	4374	275	1403	1521	0.4876	Call	TSLA190	0	0	0	https://a	5407	1.6278	2.3	1.3	2.3	1.12	1.55E+12
NFLX	#####	12075	3601	370	971	3016	0.4728	Call	NFLX190	1	1	1	https://a	8152	2.6172	1.35	3.22	4.05	1.28	1.55E+12
AMD	#####	16523	5687	28	905	26107	0.7749	Call	AMD190	0	0	0	https://a	3432	0.0936	0.15	0.09	0.17	0.06	1.55E+12
AMD	#####	20690	9696	28	9386	5804	0.4551	Call	AMD190	1	1	1	https://a	2838	1.0143	1.38	0.9	1.38	0.74	1.55E+12
NVDA	#####	7158	4872	192.5	628	2495	0.4809	Call	NVDA19	1	1	1	https://a	5150	1.0857	0.92	1.27	1.48	0.64	#####
NFLX	#####	8036	3273	380	1187	4223	0.3028	Call	NFLX190	1	1	1	https://a	11374	2.4281	0.46	4.1	4.3	0.46	#####
MSFT	#####	8796	7073	125	1211	5138	0.5333	Call	MSFT19	0	0	0	https://a	4192	1.4668	1.14	1.51	1.64	1	#####
AMZN	#####	1729	923	1920	511	651	0.4547	Call	AMZN19	1	0	1	https://a	509	22.153	14.11	26.2	26.6	12.67	#####
GOOG	#####	589	320	1180	517	292	0.2992	Put	GOOG19	0	1	1	https://a	180	1.2733	2.25	0.75	2.25	0.69	#####
AMD	#####	5150	4321	27	9239	12617	0.4776	Put	AMD190	1	1	1	https://a	3871	1.1966	1.17	1.2	1.28	1.07	#####
NVDA	#####	8112	2480	182.5	1961	4751	0.4253	Call	NVDA19	0	0	0	https://a	3692	1.9514	2.14	1.89	2.34	0.96	#####
AAPL	#####	76475	11997	215	1719	40153	0.2601	Call	AAPL190	1	1	1	https://a	8370	1.1224	0.94	1.02	1.39	0.89	#####
NFLX	#####	3323	411	367.5	1616	1910	0.423	Put	NFLX190	0	0	0	https://a	1209	1.971	2.4	1.46	2.87	1.19	#####
NFLX	#####	5066	809	347.5	906	2444	0.8039	Put	NFLX190	1	1	1	https://a	1354	7.3093	6.15	7.6	9.7	4.15	#####

This is a very small data frame picture, as the final data frame contained over 38,000 individual cells. The final data frame was 58 wide and I performed some of the statistical calculations in excel, as my computer was able is able to handle some of the data processing of this size in excel. I think that If the final file size was much larger, I would have needed to use a pandas library and some coding to be able to perform some of the statistical discovery.

The Official Options Premium Shorting Strategy

Limiting Factors

- MUST ENTER SHORT ON WEDNESDAY
- Underlying Company must go up [redacted]% from Monday to [redacted]
- The week length must be 5 days long.
- [redacted]
- Can not trade the individual stock.

Entry

- **The best entry for this premium short selling strategy is when the options contract is up [redacted]%-[redacted]% for the week.**
    - Looking at each of the individual charts, the options contracts is trading at the highest volume, and the highest amount of trades. The Individual trading this setup, needs and should be entering, around the high give or take 5%-10%. The individual that is shorting the stock should see the buy and sell orders speeding up, as the volume increases. The Trader should also be looking at the underlying stock and understand the news at play, if there is any. The Entry, even when done right still might have a harder time entering the trade. Sizing in, and properly shorting the stock, going up rather than going down, will help the individual get more contracts sold off.

Exit

- MUST COVER ALL OPTIONS SOLD, BY [redacted]
- When the underlying stock goes below or above the strike price. If the Option is a call, when the Underlying stock goes below the strike. For Puts, when the underlying stock goes above the strike price.
    - **The Average draw down including all of the losers is approximately a 70% return rate per trade. But when looking at all of the winners, on average returns 86% per trade.**

Stop Losses

- **Stop Losses when the trade is placed on Wednesday is going to be set approximately [redacted]**
    - With the average return of the shorting position being 87%, the 30% stop loss gives us a 3:1 ratio. For every win, we can encounter approximately 3 losses. By doing so, this keeps our account very healthy and in the long run can maintain the positive upsloping curve.

[redacted]

Re-Entry after stop out on Thursday

- After taking a position on Wednesday, if we [redacted] Thursday, We are able to re-short the same options contract, for an approximate 62%-90% gain. The largest being 99.98% and our largest loss being, -65%.

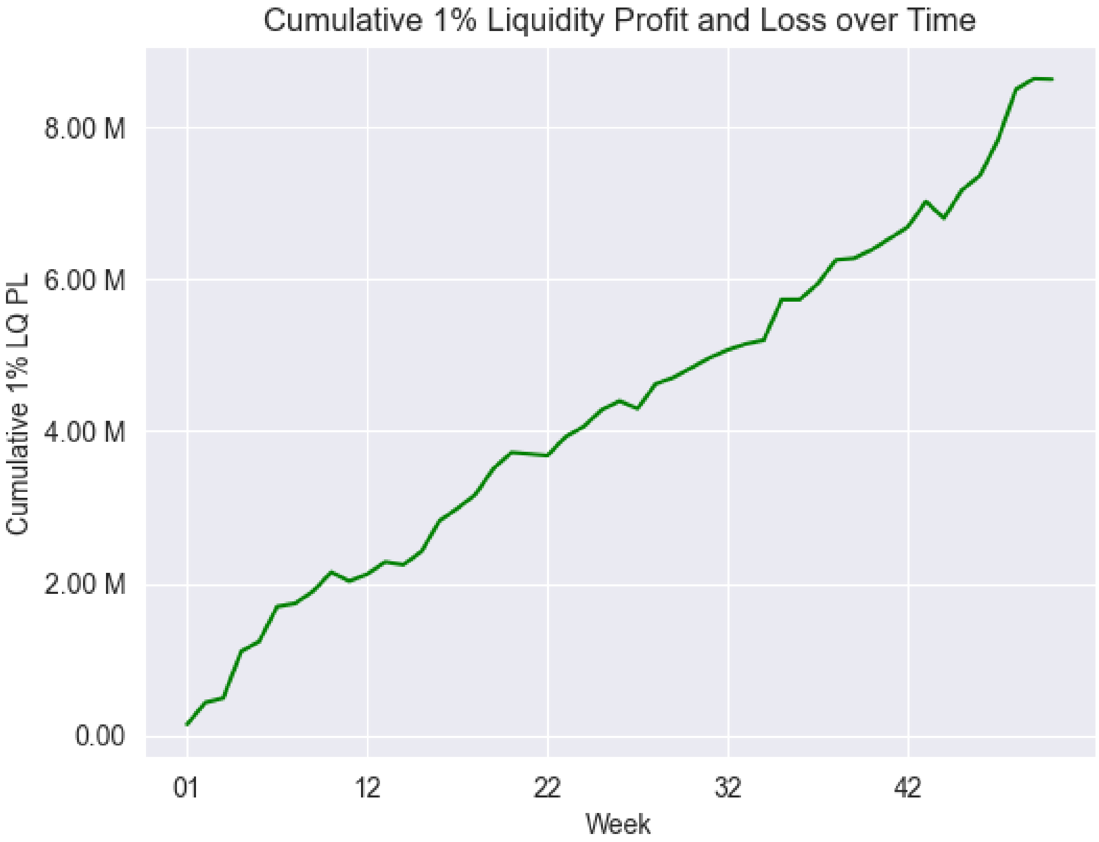
Re-Entry Stop loss.

- Our stop losses re-entry is set at [redacted] The Entry Price is going to be there so the largest loss we can take on one ticker as a whole is, [redacted] of the initial position taken on Wednesday.

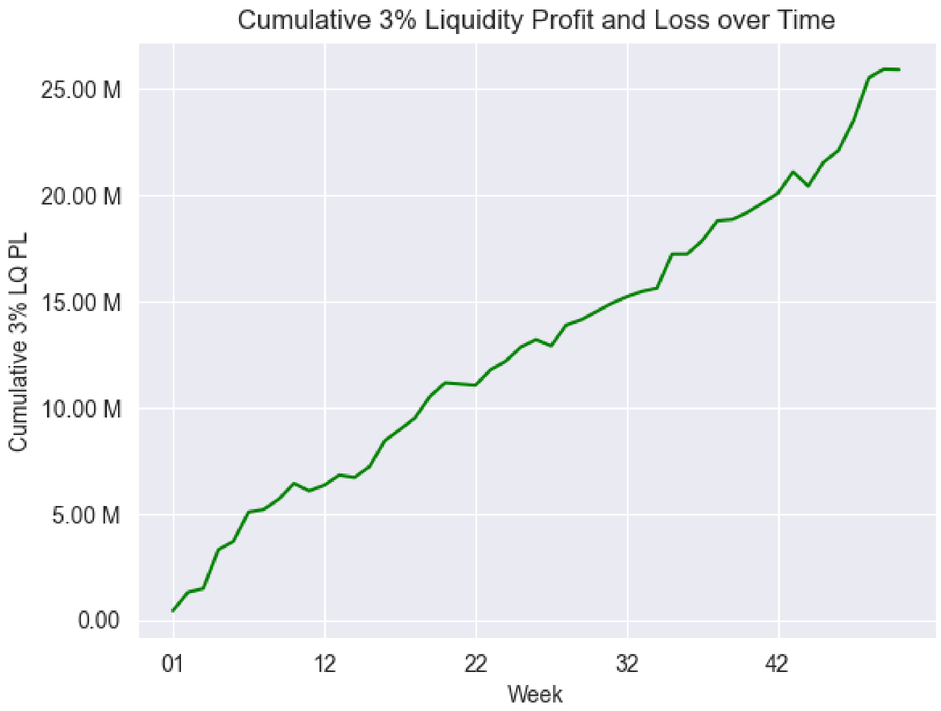
Graphing Section of Profit and Loss Curve

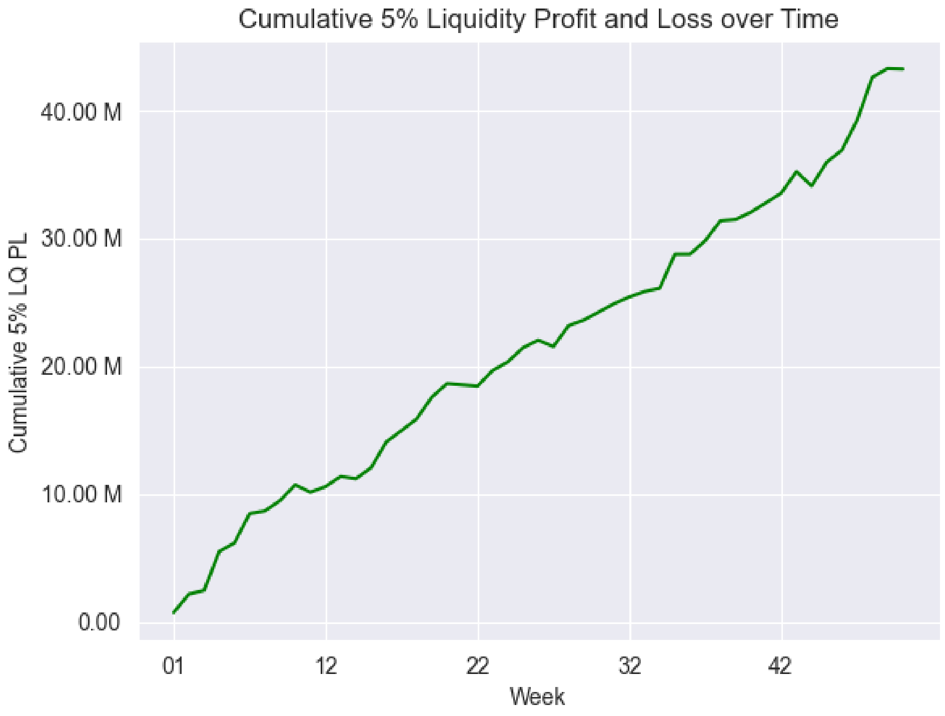
Please note that these Profit and loss curves, assume the individual has a large enough trading account in order to take the liquidity of the Open Interest, and has enough capital that they are often times able to take more than one position at a time.

Please also note that these charts do not imply the size of the orders being traded, and do not account for being able to go un-detected in the market. The ‘Entry’ Prices, and the ‘Exit’ prices are all done as if it was one massive order, this is not an extremely realistic entry and exit, as these orders will need to be entered in with smaller orders. The process of making one order into the full position size is also known as ‘Sizing in’ amongst traders. The trader will also need to ‘size out’ of the order due to the volume of the option contract being traded with every single candle.



Looking at the profit and loss you can see that the individual trading this strategy is very quickly making money and passes the $2 million dollar threshold taking one percent of the open interest liquidity. This is done cumulatively and is quickly pulling the trader in the right direction, making sure that the 25% stop loss from entry. The Trader also is able to preserve capital, by re-entering the trade, and making up some of the losses that they could encounter if they just stayed out of the Option. There is never a massive drawdown in the account or a down month. Just down weeks.

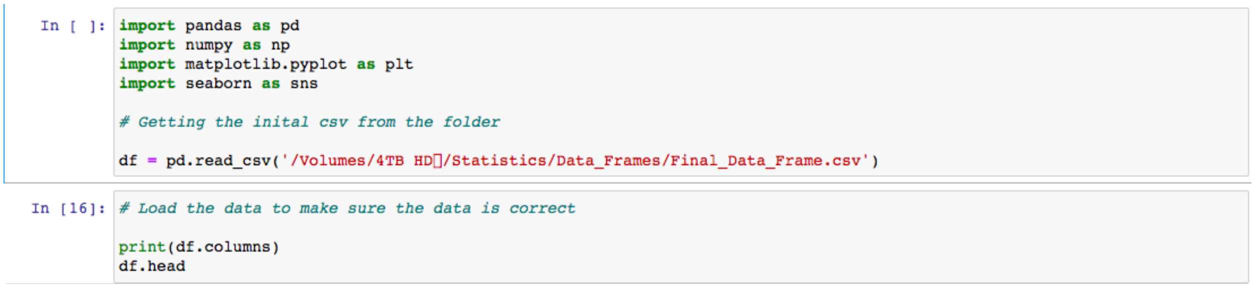




When looking at the 5% liquidity, the chart and the strategy speaks for itself. With a nearly 71% win rate, the trader can very easily make millions of dollars a week. The losses are very limited, and the trader does face more liquidity constraints when having this much money on the line, and this many contracts. If the trader is unable to cover all of his position, they should have no problem exersizing the options contract, and covering the position via stock.

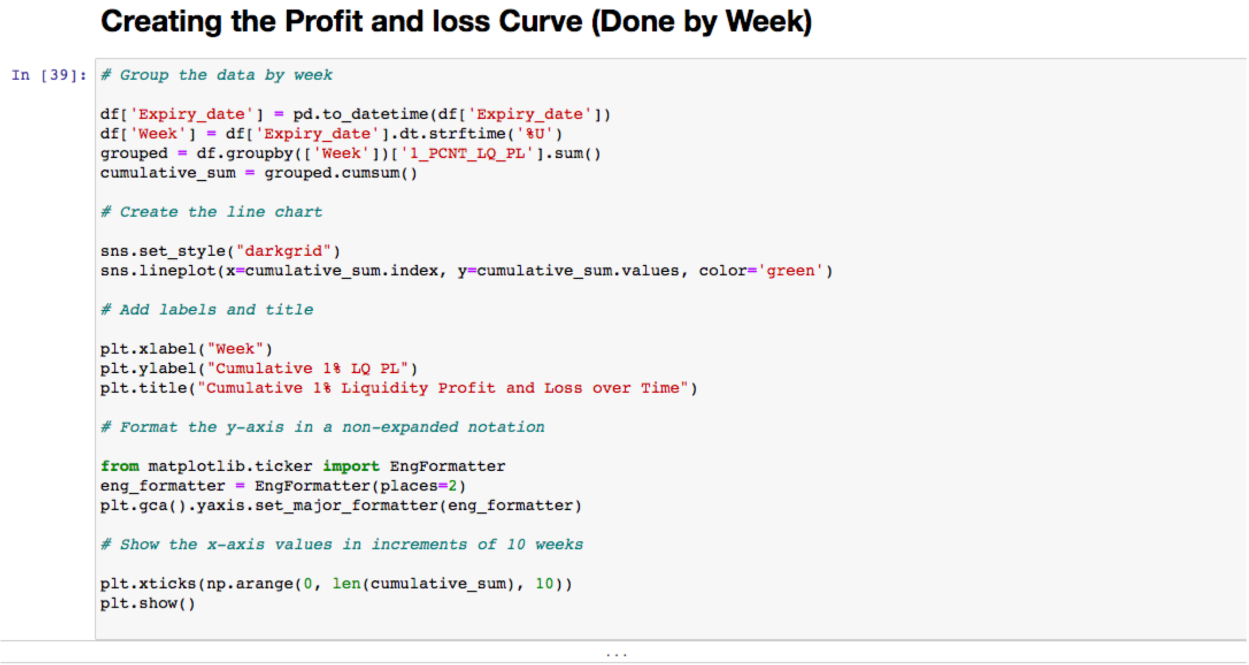
Code to Create the Charts

Environment Setup



```
In [ ]:  import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         import seaborn as sns

         # Getting the inital csv from the folder

         df = pd.read_csv('/Volumes/4TB HD□/Statistics/Data_Frames/Final_Data_Frame.csv')

In [16]: # Load the data to make sure the data is correct

         print(df.columns)
         df.head
```

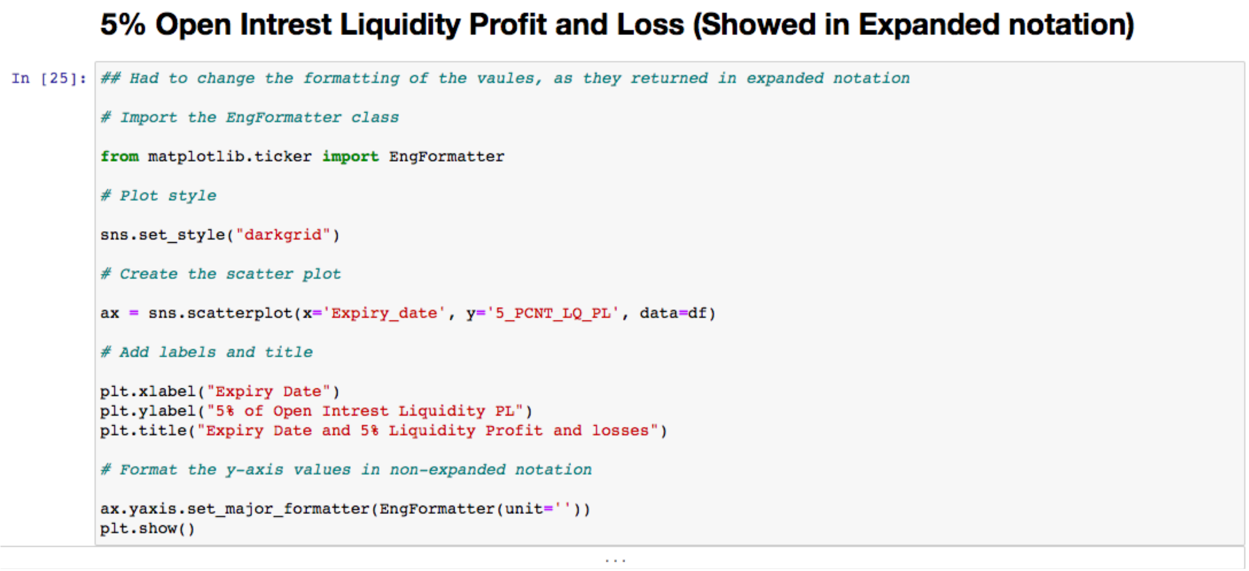
I do set up the environment in each Juypter notebook in the very first cell, and add them at the top of the cell if I need them in additional packages. I also think that when working with data manipulation what suits me best is having and working with one data frame at a time rather than having, df1, df2, df3 and so on. However, there are times where you need to work with multiples, but I think that it should always be loaded and assigned as a variable in the top cell. I do also always print the headers to make sure I am working with the proper data.



Creating the Profit and loss Curve (Done by Week)

```
In [39]: # Group the data by week

         df['Expiry_date'] = pd.to_datetime(df['Expiry_date'])
         df['Week'] = df['Expiry_date'].dt.strftime('%U')
         grouped = df.groupby(['Week'])['1_PCNT_LQ_PL'].sum()
         cumulative_sum = grouped.cumsum()

         # Create the line chart

         sns.set_style("darkgrid")
         sns.lineplot(x=cumulative_sum.index, y=cumulative_sum.values, color='green')

         # Add labels and title

         plt.xlabel("Week")
         plt.ylabel("Cumulative 1% LQ PL")
         plt.title("Cumulative 1% Liquidity Profit and Loss over Time")

         # Format the y-axis in a non-expanded notation

         from matplotlib.ticker import EngFormatter
         eng_formatter = EngFormatter(places=2)
         plt.gca().yaxis.set_major_formatter(eng_formatter)

         # Show the x-axis values in increments of 10 weeks

         plt.xticks(np.arange(0, len(cumulative_sum), 10))
         plt.show()
```

This Profit and Loss curve was done using seaborn and plotly. Seaborn is an additional package you can add onto the matplot.lib library, that makes writing the lines of code for a single chart to much easier. I think that matplot.lib is great to begin with but I think that the lack of charting and color options is what makes is a little bit lack luster. I recommend seaborn, as the color combinations and the code writing are streamlined.



5% Open Intrest Liquidity Profit and Loss (Showed in Expanded notation)

In [25]:

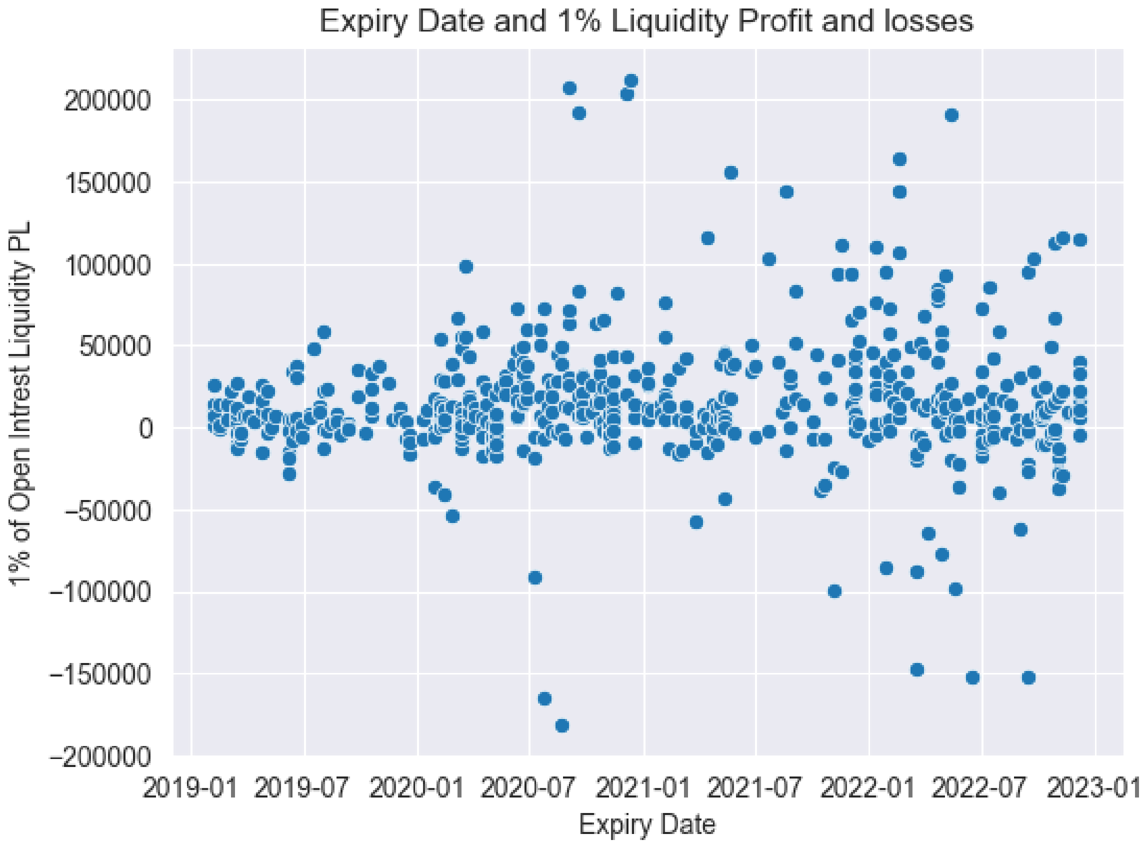
```
## Had to change the formatting of the vaules, as they returned in expanded notation

# Import the EngFormatter class

from matplotlib.ticker import EngFormatter

# Plot style

sns.set_style("darkgrid")

# Create the scatter plot

ax = sns.scatterplot(x='Expiry_date', y='5_PCNT_LQ_PL', data=df)

# Add labels and title

plt.xlabel("Expiry Date")
plt.ylabel("5% of Open Intrest Liquidity PL")
plt.title("Expiry Date and 5% Liquidity Profit and losses")

# Format the y-axis values in non-expanded notation

ax.yaxis.set_major_formatter(EngFormatter(unit=''))
plt.show()
```

...

This is me solving a problem I had with the initial 5% liquidity cumulative chart. I ran the chart using the old formula just changing some of the header columns and names, but the values were so large that on the Y axis they were in expanded notation. I added the package in this cell alone, so I could be able to show the true scale of the profitability, along with making it easier to read.

Why Stop losses are important to the trader to protect capital

A trader needs to have a stop loss. Without going into extreme detail, trading with a stop loss preserves massive losses when short selling. There is a capped amount when short selling, but there is an unlimited loss potential. Demonstrated below are the wins and losses covering solely just on Friday at the close. The 1% liquidity chart is what I used to demonstrate the importance of the stop loss.



Losses happen and they are a part of the trading game. No matter if you are Warren Buffet, or Chase bank, there will be money made and money lost but, risk mitigation, is key to making sure that you do not blow the account up. When growing the account and maintaining the account balance it is imperative that the trader avoids those massive losses and keep the larger losses to a minimum.

Final summary

With the Statistics at hand, and projecting that the win rate is kept for a very long period of time, and the individual trading the options contract is able to have a very long term success in the stock market. Considering the initial capital start up needed to trade, this strategy can be done and started with only a few thousand dollars. However, if an individual where to start trading this strategy and grow the account to a large enough size, the person would need to make sure that his orders are slipped in to the normal market feed very small. When considering that the market changes extremely fast, the trader must make sure that he still has the edge against the people he is betting against. From what I know, and what has been shared about actual proper trading strategies that are not done by an organization or algorithm, this trading strategy is very basic at the core of it. But the results and return per trade are extremely ‘abnormal’.

My Personal Thoughts

I think that this project taught me a lot about how to manage my time, and what actually needs to be focused on when doing a project like this. I was only able to work on this for 4 hours a day. The majority of my time was allocated to reading, and being able to take the knowledge I had previously found out, and apply it in a more scientific approach. By my estimation I spent approximately 218 hours on the project overall. I think that the project could have been done in a much shorter time span, from December 7th 2022 was the official start date of the first lines of code, up until February 10, 2023, if I had more time to dedicate to the day.

The possibility of more research

This entire paper only focused on having a solid set in place stop loss system, but I would like to consider in the future having a more dynamic stop loss system. A stop loss system that can move keep more profit in the hands of the trader, and then track the profit and loss curve. I think that the retainment of capital is not talked about enough by any sort of day trader, or publicly known day trader for that matter. Secondly, I would love to put every single highly liquid options contract into a LSTM neural network and have the neural network correctly and accurately predict the outcomes of various options pricing and movements. I would like to have the user just input the name of the options contract, and show the predicted week end outcome given the market conditions and the current order flow for that options contract. I am going to be saving all of the csv files in hopes of some day making a state of the LSTM neural network.

Lastly, all the code, or the bulk of the code for filtering and processing the data has been deleted. The last thing I would personally want is to see someone read this report, and make their own system to replicate what I found, and turn around and sell it. The small screenshots of code are all that I will show. However they are the foundational blocks of this project, and statistical discovery in my eyes. I will have a small picture of the final data frame, I used, and a link to some of the charts that I created. But other than that, the project files have been deleted. Thank you for reading.