Trading Volatility Mispricings in the Options Market

Anthony Yang, Lily Liu, Thipok Lertplakorn

Abstract

Since there is no deterministic away to compute implied volatility of an underlying asset – a crucial factor in options pricing – options may trade at prices that may or may not be fair. In this project, we aim to identify these mispricings in the options market using Autoregressive Monte Carlo Simulation on underlying assets, and maximize our expected profit by trading options spreads and combinations. Different options strategies are treated as individual assets and the Markowitz Mean-Variance Optimization model is utilized to optimize our portfolio. We aim to trade these options strategies three hours before their expiration on Friday of every week.

1. Introduction

As derivative securities, options are traded as an effective hedge against a declining stock market to limit downside losses, or as a less-risky way to generate recurring income and for speculative purposes. Option traders usually concern two forms of volatility: historical and implied volatility. While historical volatility represents the past and measures how much the stock price fluctuated on a day-to-day basis over a one-year period, implied volatility is an estimation of the volatility of a stock (or security) in the future based on the market over the time of the option contract.

In the context of this project, we explore implied volatility mispricings in the options market and intend to build an optimal portfolio of option trading strategies given a set of underlying assets. We focus on the options one week from their expiration date and follow the trading workflow: at 1:00PM on each Friday (i.e. three hours before options expiration), we simulate the stock price paths by Monte Carlo simulation using an Auto-Regressive (AR) model. Based on the simulated stock prices and volatilities, we calculate the expected payoff at expiration for each option trading strategy and trade on optimal portfolio of strategies which maximizes the returns. The "three hours before expiration" time point is chosen to balance the tradeoff between time-premium (theta-value) obtained from writing the options and the accuracy of simulation.

By treating the different option trading strategies (i.e. option spreads) as different assets, the Markowitz Mean-variance optimization model can theoretically be applied. The

Email addresses: `axy2@cornell.edu` (Anthony Yang), `xl379@cornell.edu` (Lily Liu), `tl424@cornell.edu` (Thipok Lertplakorn)

model will maximize the expected returns subject to given risk and the expected returns are calculated as the expected payoffs of the option traded minus the premium of the options.

The report is organized in the following manner:

- Section 2: Data collection and cleaning, including historical stock and option prices at minute granularity
- Section 3: Assumptions of the project
- Section 4: Monte Carlo simulation using an Autoregressive Moving Average (ARMA) model
  - To determine the expected payoff of the options, we simulate multiple paths of the underlying asset prices 3 hours before the option expiration (i.e. 1:00PM to 4:00PM) on each Friday
- Section 5: Three chosen payoff functions – Condor, Straddle, and Put Spread
  - Calculation of expected payoffs of each payoff function
  - Creation of our portfolio's covariance matrix
- Section 6: Application of Markowitz Optimization model. Our model will maximize the expected return subject to user-defined risks
- Section 7: Testing methods, results of the optimal portfolio, and an analysis on the strategy
- Section 8: Conclusion and proposal for further exploration in the continuing project

2. Data

2.1. Data Collection

Our data was centered around the following 6 stocks and their respective options:

1. Alibaba (BABA)
2. NASDAQ Index (QQQ)
3. Netflix (NFLX)
4. Facebook (FB)
5. Tesla (TSLA)
6. Apple (AAPL)

As our strategy encompasses combining various options together, we need data at the most granular level, namely minute-by-minute intraday data. Additionally, since we apply Monte Carlo simulation using an Autoregressive Moving Average (ARMA) model 3 hours before options expiration, the more granular the data is, the greater the accuracy and precision will be. Because Bloomberg data's granularity is dependent on the frequency of trade, the tickers chosen were essentially the most popular, and thus, were the most liquid. Furthermore, we focused on examining options that were one week from expiration since longer periods of time are generally much more volatile. While we utilized American options due to convenience and ease of access, one of our major assumptions was to regard each option as European.

Regarding which strike to choose from for each option, we considered the stock price on Friday 1:00 PM (3 hours before expiration) and choose the 5 strikes closest to the price and therefore, most likely to be exercised. The 5 strikes are chosen as [price (rounded), price - 2*strike_interval, price - strike_interval, price + strike_interval, price + 2*strike_interval]. Notice that the strike intervals may vary from one option to another.

As mentioned before, all data collection, stocks and options, were done through the Bloomberg terminal. We used a Bloomberg Python API to obtain stock and option prices and Pandas to store the prices in data frames and output into csv files.

```python
# Import packages
import pdblp
import pandas as pd

# Set up connection
con = pdblp.BCon(debug=False)
con = con.start()
con.timeout = 5000

# Get call option prices for Apple
AAPL_call = con.bdib('AAPL US 03/15/19 C182.5 Equity', '2019-03-11T14:30:00', '2019-03-15T21:00:00',
                     event_type='TRADE', interval=1)

# Get put option prices for Apple
AAPL_put = con.bdib('AAPL US 03/15/19 P182.5 Equity', '2019-03-11T14:30:00', '2019-03-15T21:00:00',
                     event_type='TRADE', interval=1)

# Get stock prices for Apple
AAPL_stock = con.bdib('AAPL US Equity', '2019-03-11T14:30:00', '2019-03-15T21:00:00', event_type='TRADE', interval=1)
```

Figure 1: Sample code for Bloomberg API calls

2.2. Data Issues and Cleaning

One of the issues with the data gathered from Bloomberg was the inconsistencies in granularity. As mentioned before, Bloomberg's granularity scales with liquidity and so, there were often missing data for particular minutes and even hours of a day. This is the key problem in constructing the covariance matrix of the various options since the matrix requires consistent formatting and length of the data. As a result, we have to impute the missing values by taking the last known price and applying for the missing time period, assuming that Bloomberg only updates its data when the price changes.

However, as with any data related project, we were plagued with the problem of too little data, despite our tickers being on the more liquid side. There were cases where we were missing whole days of data and were essentially forced to impute the same price for the entire day (6.5 trading hours/ day * 60 minutes/hour = 390 values). In addition, Bloomberg only keeps a record of granular options data for the last 6 weeks, so our overall quantity was poor as well. Looking forward, we will be continuously gathering more data for the second project.

3. Assumptions

For simplicity and clarity, we have made the following assumptions:

1. Options are only exercised at expiration (i.e. American options treated as European options)
2. No commission fee
3. Options are liquid enough so they can be traded at any time
4. Model allows a fraction of shares to be traded
5. Shorting is allowed

4. Stock Price Simulation Model

In order to calculate the expected payoff of each options trading strategy at expiration, the underlying stock and its volatility must first be simulated. Using the minute log returns that was extracted from Bloomberg for the trading week, we attempt to perform Monte Carlo simulation on an Autoregressive(5) model to simulate multiple paths that the underlying asset could follow from 1:00 PM to 4:00 PM on Friday. The order of the AR model is chosen as such according to recommendations by [1] , which suggest stock minute returns autocorrelation quickly vanishes for large lags, but is present for small intraday time scales.

The Autoregressive (AR) model chosen for this purpose has the following functional form:

$$X_t = c + \sum_{i=1}^{5} \phi_i X_{t-i} + \varepsilon_t$$

Where X_t is the predicted log-return at minute t and ϕ_i is the parameters of the model. The Monte Carlo component of the model involves gaussian sampling the error ε_t for each prediction point.

Furthermore, we chose to omit stock returns data during the first hour and a half of each trading day when fitting the AR(5) model. This is because we are simply interested in the stock path simulation three hours before market close, whereas the volatility of stock returns in the first hour and a half of the market opening tends to be much higher and unrepresentative of returns from 1:00 PM to 4:00 PM.

With the Monte Carlo samples from our AR(5) model (Figure 2), we obtain the predicted price paths and volatility of our desired stock, as plotted in Figure 2. With these results, we can now calculate the expected payoff at expiration for each options trading strategy.

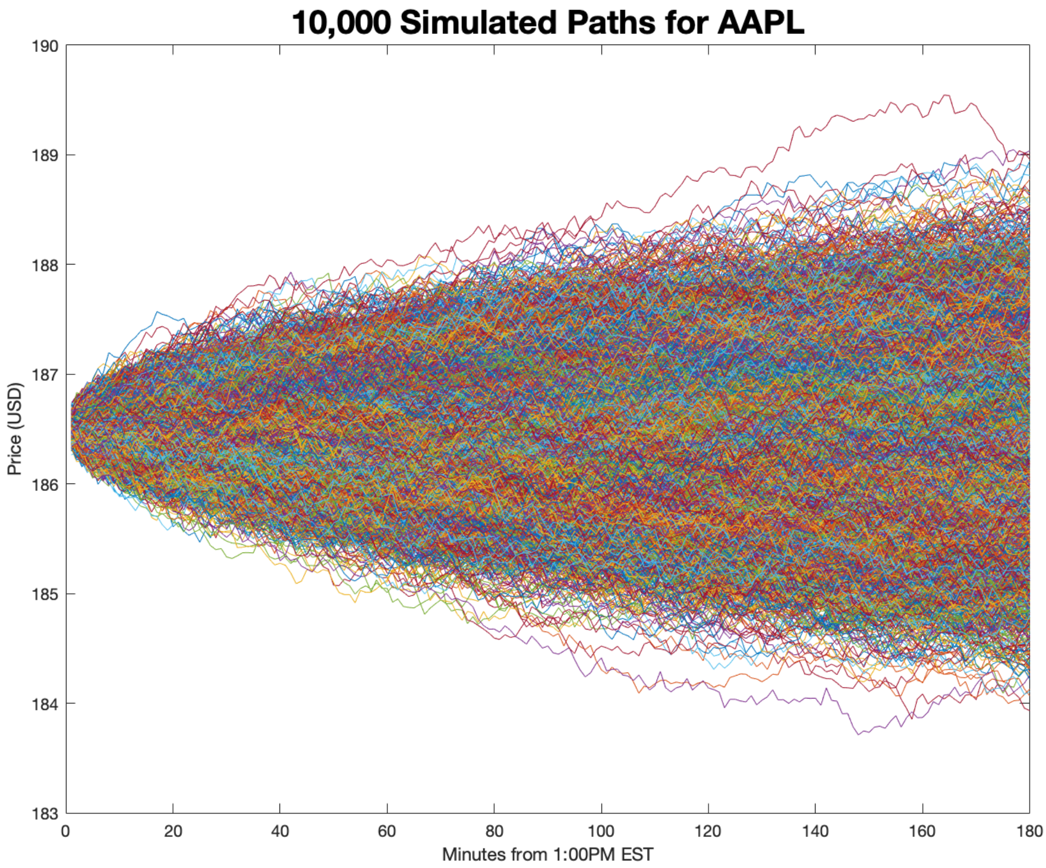

Figure 2: AR(5) Monte Carlo Simulation on AAPL Returns

5. Options Strategies and Expected Payoff at Expiration

Once the stock prices at Friday 4:00 PM are simulated, the expected payoff for each options trading strategy (at expiration) can be calculated.

5.1. Options Strategies

For this project, the Condor, Straddle, and Put Spread strategies are considered. Both the Condor and Straddle strategies are market neutral (meaning the buyer is neither bullish nor bearish on the underlying stock), whereas the Put Spread is directional. An example of how each strategy can be constructed is demonstrated in Table 1.

Underlying Stock Price (at 1:00 PM) = $150					
Strategy/Strike	Strike = $130	Strike = $140	Strike = $150	Strike = $160	Strike = $170
Condor	Sell 1 Put	Buy 1 Put		Buy 1 Call	Sell 1 Call
Straddle			Buy 1 Call, Buy 1 Put		
Put Spread		Sell 1 Put		Buy 1 Put	

Table 1: Options Strategy Construction Example

Figure 3, Figure 4, and Figure 5 plot the payoff of each options strategy (if they were constructed according to the Table 1) at expiration, with respect to the underlying stock price at 4:00 PM.

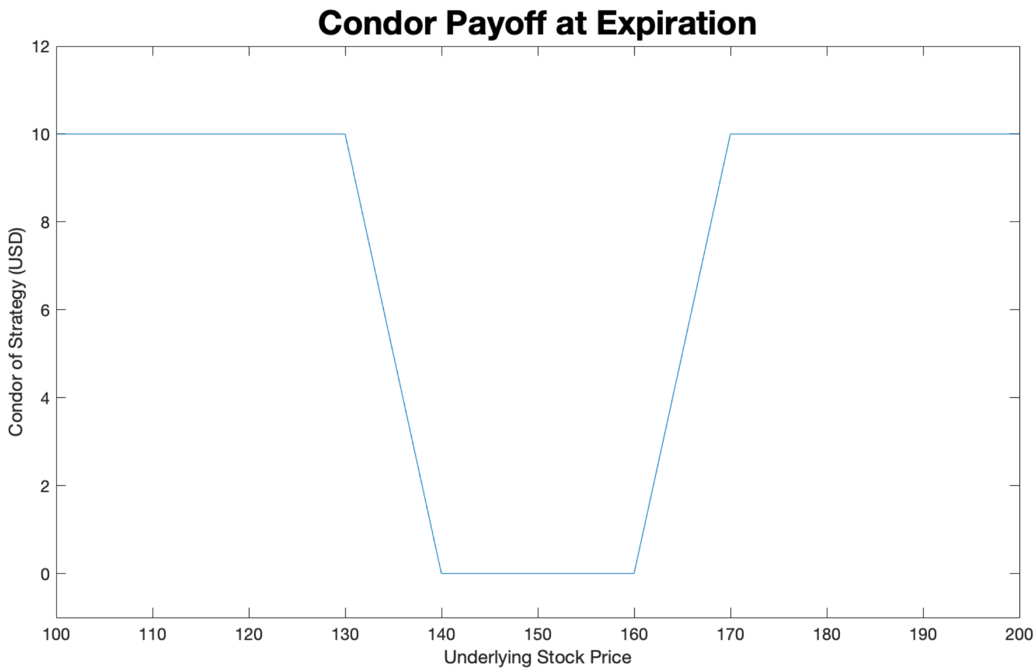

Figure 3: Condor

In the actual simulation, the construction of option strategies abide by the rules seen in Table 2.

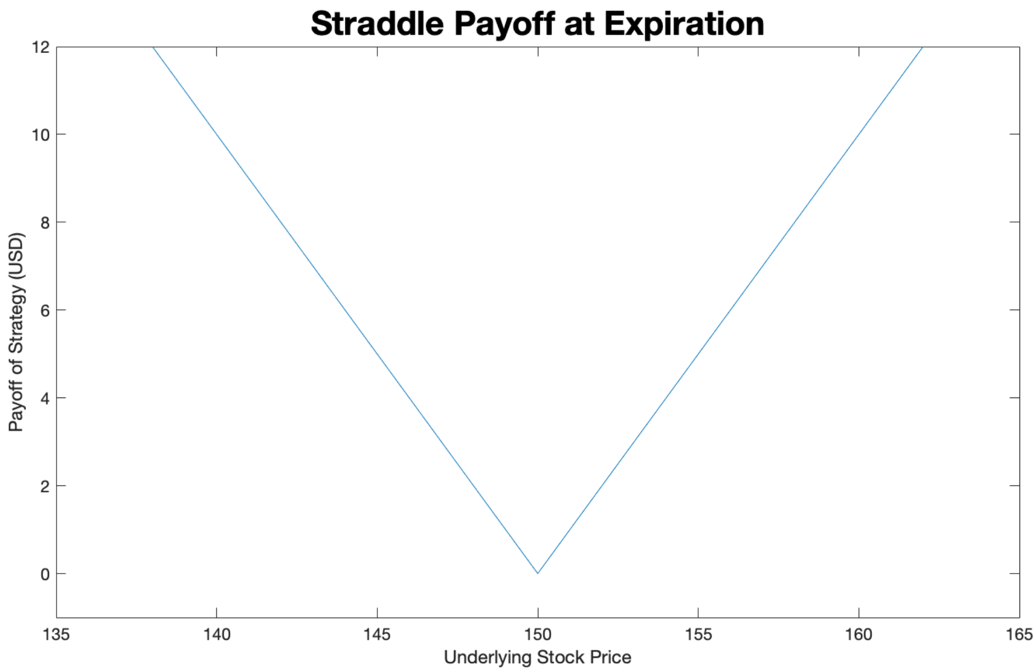

Figure 4: Straddle

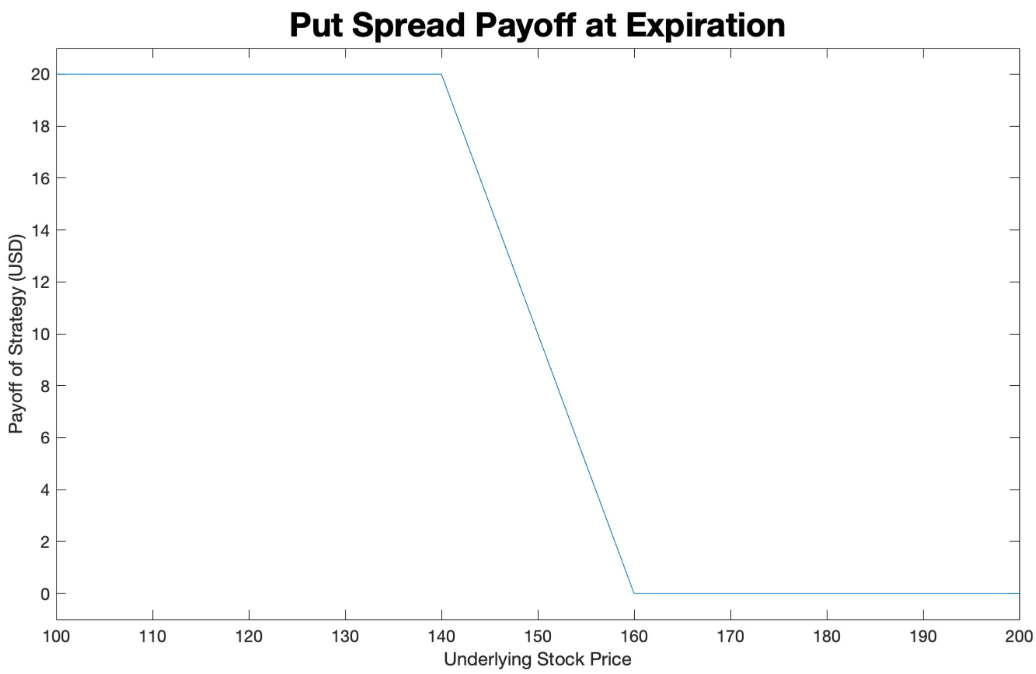

Figure 5: Put Spread

Strategy	Strike Choosing Rule	Example (Stock = $102, Strike Interval = $2.5)
Condor	• Sell 1 Put at two strikes below the stock price • Buy 1 Put at strike below the stock price • Buy 1 Call at strike above the stock price • Sell 1 Call at two strikes above the stock price	• Sell 1 Put at strike $97.5 • Buy 1 Put at strike $100 • Buy 1 Call at strike $102.5 • Sell 1 Call at strike $105
Straddle	• Buy 1 Put at strike closest to stock price • Buy 1 Call at strike closest to stock price	• Buy 1 Put at strike $102.5 • Buy 1 Call at strike $102.5

Table 2: Options Strategy Construction Rules

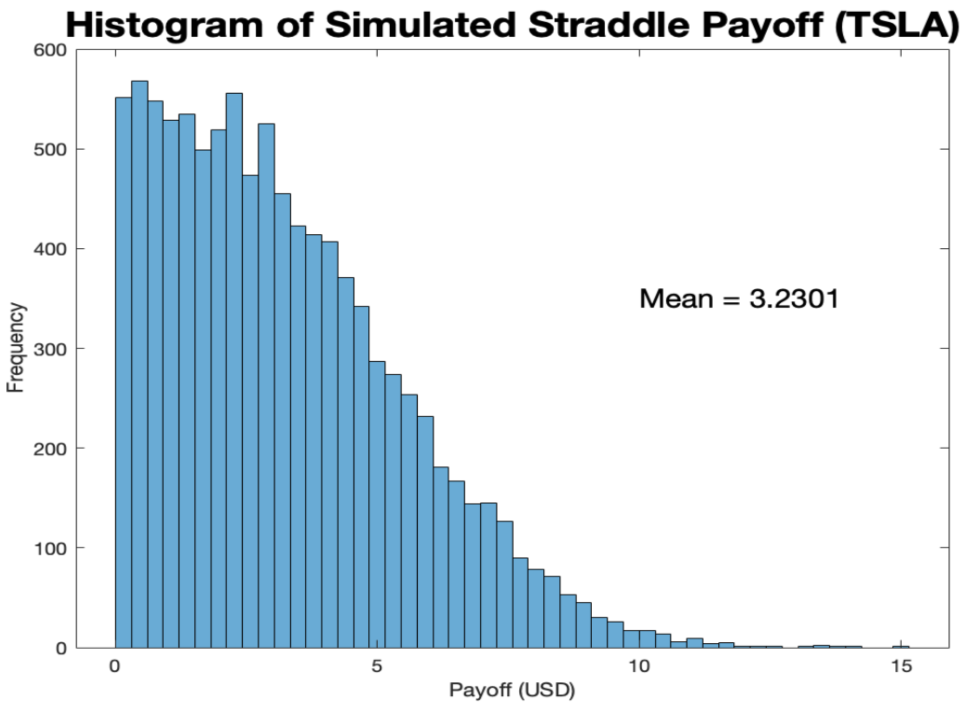

Figure 6: Options Payoff Histogram

5.2. Expected Payoff and Profit

In order to calculate the expected payoff of each options strategy, we input the simulated stock prices from our Monte Carlo simulation into the payoff functions that we created in MatLab, and then take the average.

Strategy	Expected Payoff	Cost	Expected Profit	Expected Return
AAPL Straddle	0.6953	0.8200	-0.1247	-0.1521
AAPL Condor	0.3123	0.3300	-0.0177	-0.0536
AAPL Put Spread	0.3826	0.4500	-0.0674	-0.1498
BABA Straddle	0.7488	1.1400	-0.3912	-0.3431
BABA Condor	0.5553	0.3400	0.2153	0.6331
BABA Put Spread	0.1921	0.7400	-0.5479	-0.7404
FB Straddle	1.8292	1.2100	0.6192	0.5117
FB Condor	0.1658	0.1200	0.0458	0.3815
FB Put Spread	0.8371	1.4600	-0.6229	-0.4267
NFLX Straddle	2.3915	3.0300	-0.6385	-0.2107
NFLX Condor	1.3898	1.2900	0.0998	0.0774
NFLX Put Spread	0.4010	1.1500	-0.7490	-0.6513
QQQ Straddle	0.9784	0.7600	0.2184	0.2874
QQQ Condor	0.2488	0.2700	-0.0212	-0.0784
QQQ Put Spread	0.2893	0.5800	-0.2907	-0.5013
TSLA Straddle	3.2301	2.7800	0.4501	0.1619
TSLA Condor	1.1318	1.0500	0.0818	0.0779
TSLA Put Spread	0.8026	1.3100	-0.5074	-0.3874

Table 3: Strategy Expected Returns at 1:00PM Feb 08, 2019

For example, we can look at the histogram (Figure 6) and expected payoff of the Condor strategy for simulated AAPL prices

After the calculations of the expected payoff for each strategy, we can calculate the expected profit, which is simply the expected payoff minus all costs to set up the position. By using the ticker data we obtained from Bloomberg, we can combine the price of the options with strikes corresponding to the instructions in part 5.1 (Options Strategies) to get cost of position for each strategy at 1:00 PM on Friday.

Additionally, by treating each options strategy as an individual asset, we can also calculate the weekly covariance of each option strategy as we have all the necessary data to calculate the price of each strategy at any time during the trading week.

For example, at 1:00 PM Friday, February 8th, the expected profits and returns when the options expire at the end of day are seen in Table 3.

6. Optimization Model

In order to choose the most optimal portfolio of options strategies to trade on Friday of each week, we resort to the Markowitz model with several additional constraints.

Specifically, we seek to maximize the expected return, such that the standard deviation of our portfolio is no more than 5% of our net assets. Due to the risky nature of options strategies, we also limit both the net credit and net debit of positions to be no more than 5% of our net assets. This ensures positions that expire to zero payoff do not affect our portfolio too much. Lastly, we limit the maximum number of shares of each strategy that would could buy or short to 100 in order to account for realistic market liquidity.

$$
\begin{aligned}
&\max x^T \mu \\
&\text{s.t.} \\
&x^T V x \leq (0.05P)^2 \\
&x^T c \leq 0.05P \\
&x^T c \geq -0.05P \\
&x \geq -100 \\
&x \leq 100
\end{aligned}
$$

Where V is the covariance matrix of each options strategy, P is the current portfolio value, c is the vector of costs to set up each strategy, μ is the expected return vector, and x is the portfolio weights.

7. Testing and Results

After running the trading strategy over entire amount of data (6 weeks), we obtain the following evolution of portfolio net assets and weekly profits/losses as seen in Figure 7.

In Figure 7, we can observe that over 6 weeks, our portfolio has grown from \$10,000 to \$12,016. We make a net profit trading in 5 out of the 6 weeks, with a standard deviation of \$368.21.

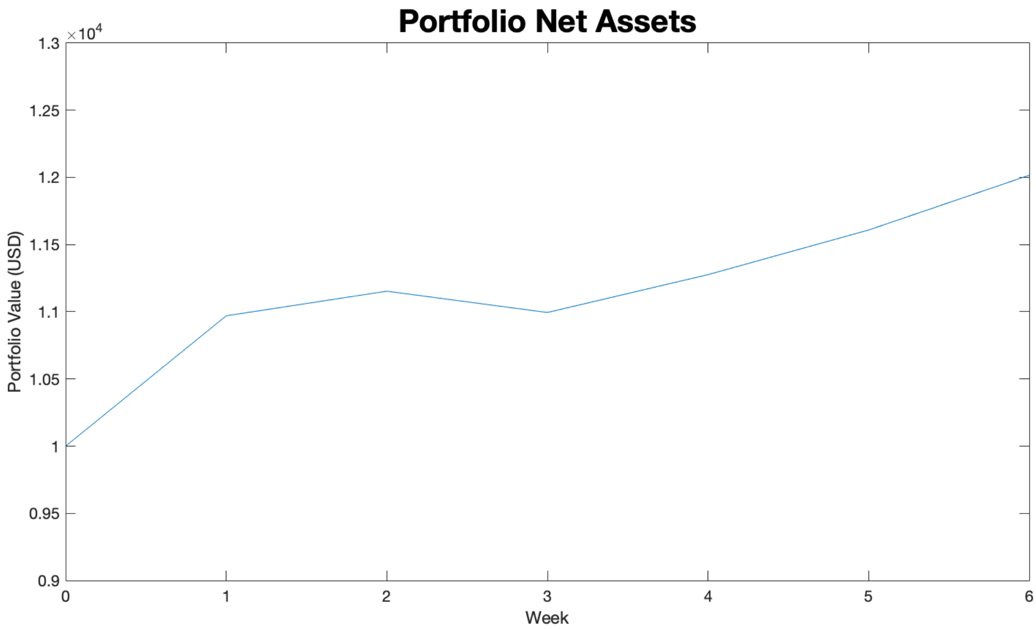

Figure 7: Portfolio Net Assets over Time

Compared to a naive strategy where we simply long implied volatility every week (assume the market under-prices implied volatility), we obtain the following results for our portfolio as seen in Figure 8.

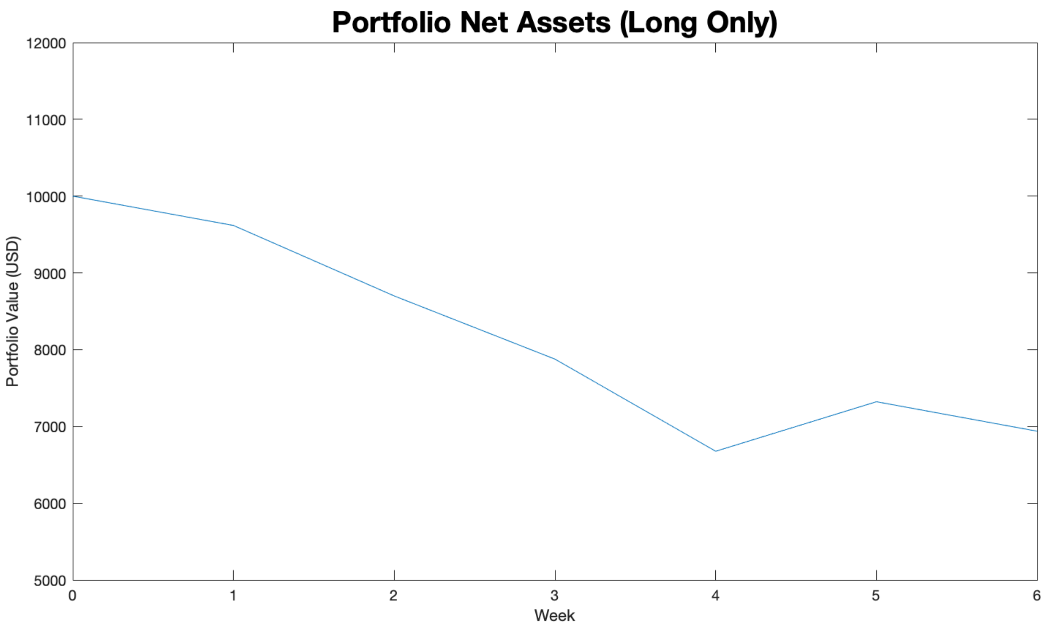

Figure 8: Portfolio Net Assets (Long Volatility Only) over Time

In Figure 8, we can see that the long-volatility only portfolio loses roughly 30% of its net assets after 6 weeks of trading. The standard deviation for the profits/losses is \$649.13, which is much higher than the standard deviation in our Markowitz portfolio.

On the other hand, a portfolio that only shorts volatility (assume market over-prices implied volatility) results in the following portfolio evolution as seen in Figure 9.

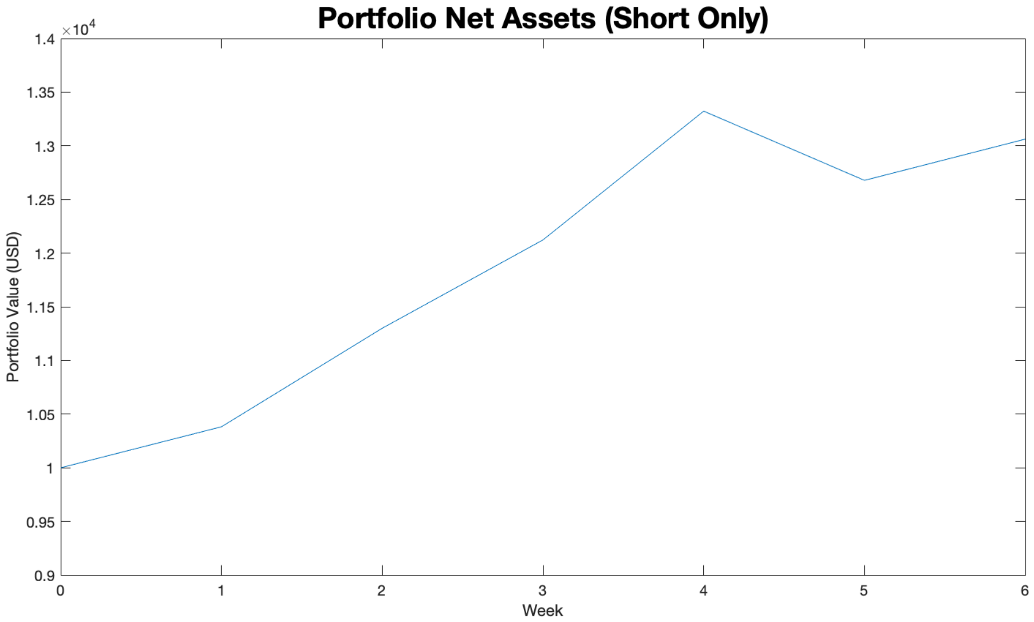

Figure 9: Portfolio Net Assets (Short Volatility Only) over Time

In Figure 9, we can see that the short-volatility only portfolio grows around 30% in assets, which is larger than the gain we see in our Markowitz portfolio. However, the standard deviation of the profit/losses is \$649.13, which is also much higher than the standard deviation of profits/losses in our Markowitz portfolio. Note that the standard deviations of profits/losses in both the long-only and short-only portfolio are the same, since options strategies are zero-sum.

Profit (+) / Losses (-)	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Standard Deviation
Markowitz Portfolio	$970.00	$183.20	-$158.43	$281.70	$332.05	$407.92	$368.21
Long Volatility	-$382.00	-$918.00	-$823.00	-$1200.00	$645.00	-$384.00	$649.13
Short Volatility	$382.00	$918.00	$823.00	$1200.00	-$645.00	$384.00	$649.13

Table 4: The portfolio net assets (profit/losses) and standard deviations for different portfolios.

8. Conclusion

In this project, we focused on implied volatility mispricings in the options market and built an optimal portfolio of option trading strategies using the Markowitz Mean-Variance Optimization Model. Under reasonable assumptions, the objective of the model is to maximize the expected returns subject to given risks.

According to the testing results obtained from running the trading strategy over entire 6-weeks data, we observe that our portfolio leads to a growth in net assets with a relatively low volatility. Compared with two naive strategies that exclusively shorts and longs volatility, our model outperforms the long volatility strategy in both net profit and standard deviation and has a lower standard deviation than the short volatility strategy.

However, due to the lack of data, we could not draw any firm conclusion that our portfolio will outperform the naive strategies under all market conditions. But we are confident enough to conclude that we do find it a viable trading strategy.

For future direction of the second project, we plan on increasing the size of the data by continuously gathering more prices through Bloomberg. Furthermore, we also consider (i) adding constraints on no fractional trades, (ii) including commission fee, (iii) and designing more extensive stock price simulation model. Another goal can be the inclusion of more diversified stocks contingent on the granularity of option prices.

References

[1] R. Cont, Volatility Clustering in Financial Markets: Empirical Facts and Agent–Based Models (2001). URL https://www.lpsm.paris/pageperso/ramacont/papers/clustering.pdf

Author biography

Anthony Yang Cornell Undergraduate: Computer Science Major, Operations Research Minor

Lily Liu Cornell Undergraduate: Computer Science and Operations Research Major

Thipok Lertplakorn Cornell Undergraduate: Operations Research Major