

Options Trading Strategies Analyzer - Emphasizing Market Volatility

A PROJECT REPORT FOR FRE-GY 7801 – Real-Time Trading Risk Management



Submitted By (Aum Biyani)
December 17, 2023

OptionsTradingStrategiesProjectReport

Dept. of Finance and Risk Engineering
NYU Tandon School of Engineering

Adjunct Professor, Naresh Malhotra
nm89@nyu.edu

INDEX

Sr. No.	Sections	Pg. No.
1.	Introduction	2
2.	Project Design and Structure	3
3.	Implemented Functions	4
4.	User Interaction	6
5.	Future Market Scenarios	7
6.	Visualization	8
7.	The Strategies	8
8.	Conclusion	11
9.	Future Enhancements	11
10.	Appendix	12

Introduction:

The Options Trading Strategies Analyzer is a sophisticated and user-friendly tool designed to provide investors and traders with a comprehensive platform for analyzing and visualizing a diverse range of options trading strategies. With a focus on real-time data, market volatility, and user empowerment, this tool aims to facilitate informed decision-making in the dynamic landscape of financial markets. The project is structured around three core components: Data, Model, and Visualization.

The Data component encompasses functionalities for fetching historical stock data and option chain details, integrating with the Yahoo Finance API, and validating user-provided data such as ticker symbols and dates.

The Model component includes functions for calculating payoffs for each strategy based on user-defined market scenarios, strategy-specific logic for filtering options and determining strike prices, and defining the initial position of the strategy.

The Visualization component involves creating visual representations of payoff diagrams for each strategy, calculating final positions and P&L for each market scenario, and tabulating the results of each strategy, thereby providing users with a comprehensive analysis of potential outcomes.

User interaction is a key focus of the project, with a straightforward and interactive process designed to empower individuals interested in analyzing options trading strategies for specific stocks. Users are prompted to input a ticker symbol representing a specific stock or financial

OptionsTradingStrategiesProjectReport

Dept. of Finance and Risk Engineering
NYU Tandon School of Engineering

Adjunct Professor, Naresh Malhotra
nm89@nyu.edu

instrument, which triggers the utilization of the Yahoo Finance API to fetch historical stock data and option chain information for the specified symbol. Additionally, users are required to input buy and expiration dates in a specific format for proper processing, ensuring seamless entry of relevant data for analysis.

The tool's analytical capabilities extend to a nuanced understanding of potential payoffs for each strategy across varied market scenarios, including bullish, bearish, and choppy conditions. It dynamically adjusts future stock prices, enabling users to adapt strategies based on evolving market conditions. Furthermore, the tool provides valuable insights into the risk and reward profiles of each trading strategy, aiding users in assessing potential gains or losses. With user-friendly flexibility and customization options, users can input preferred ticker symbols and dates, tailoring analyses to specific stocks and timeframes.

As an educational resource, the project offers hands-on experience in the analysis and visualization of different options trading strategies, contributing to a deeper understanding of the complex dynamics involved. The tool's modular structure not only ensures adaptability but also positions it for future enhancements and feature additions, fostering continuous improvement. It aims to enhance its analytical capabilities, user experience, and range of supported options trading strategies, thereby providing users with a more comprehensive toolkit for crafting diverse and sophisticated trading approaches.

Project Design and Structure:

1. Menu

The menu serves as the main interface for the user to interact with the tool. It includes options for data fetching, model calculation, and visualization.

2. Data

The Data component encompasses functionalities for fetching historical stock data and option chain details, integrating with the Yahoo Finance API, and validating user-provided data such as ticker symbols and dates. It includes the following functions:

- **fetch_stock_data:** fetches historical stock data for a given ticker symbol
- **fetch_option_expiration_dates:** fetches option expiration dates for a given ticker symbol
- **fetch_option_chain:** fetches option chain details for a given ticker symbol and expiration date

The project also includes a User Interaction component, which prompts users to input a ticker symbol representing a specific stock or financial instrument, buy and expiration dates, and market scenarios. It utilizes the Yahoo Finance API to fetch historical stock data and option chain information for the specified symbol and validates user-provided data for proper processing.

3. Model

The Model component includes functions for calculating payoffs for each strategy based on user-defined market scenarios, strategy-specific logic for filtering options and determining strike prices, and defining the initial position of the strategy. It includes the following functions:

- **options_filter:** filters the option chain to the n nearest to ATM strike.
- **straddle:** calculates logic for a straddle strategy and selects the option strikes.
- **strangle:** calculates logic for a strangle strategy and selects the option strikes
- **iron_condor:** calculates logic for an iron condor strategy and selects the option strikes
- **iron_butterfly:** calculates logic for an iron butterfly strategy and selects the option strikes
- **ratio_call_spread:** calculates logic for a ratio call spread strategy and selects the option strikes.

4. Visualization

The Visualization component involves creating visual representations of payoff diagrams for each strategy, calculating final positions and P&L for each market scenario, and tabulating the results of each strategy, thereby providing users with a comprehensive analysis of potential outcomes. It includes the following functions:

- **calculate_payoff2:** calculates payoffs for a 2-leg strategy
- **calculate_payoff4:** calculates payoffs for a 4-leg strategy
- **calculate_ratio_call_spread_payoff:** calculates payoffs for a ratio call spread strategy

Overall, the project structure is designed to provide a comprehensive and user-friendly platform for analyzing and visualizing a diverse range of options trading strategies, empowering users to make informed decisions and gain valuable insights into the risk and reward profiles of various trading strategies.

Implemented Functions:

1. `fetch_stock_data(symbol, period='1d')`:
 - This function fetches historical stock data for a given ticker symbol within a specified date range.
 - It utilizes the Yahoo Finance API to retrieve data and returns a pandas dataframe containing the stock data.
2. `fetch_option_expiration_dates(symbol)`:
 - This function fetches option expiration dates for a given ticker symbol.
 - It utilizes the Yahoo Finance API to retrieve data and returns a list of expiration dates.
3. `fetch_option_chain(symbol, expiration_date)`:
 - This function fetches option chain details for a given ticker symbol and expiration date.
 - It utilizes the Yahoo Finance API to retrieve data and returns a pandas dataframe containing the option chain details.

OptionsTradingStrategiesProjectReport

Dept. of Finance and Risk Engineering
NYU Tandon School of Engineering

Adjunct Professor, Naresh Malhotra
nm89@nyu.edu

4. `options_filter(options, current_stock_price, n=10):`
 - This function filters the option chain to the “n” closest option contracts to the current stock price.
 - “n” has been set to a value of 10 to remove deep ITM and OTM options.
5. `straddle(call_options, put_options, current_stock_price, stock_prices):`
 - This function calculates the logic for the straddle strategy.
 - We first identify the ATM contract for the call and put option using `argsort()[1]`
 - Then we pass the value of the strikes and premiums for the same.
6. `strangle(call_options, put_options, current_stock_price, stock_prices):`
 - This function calculates the logic for the strangle strategy.
 - We first identify the ‘n’ nearest OTM contracts for the call and put option using `options_filter`.
 - From the filtered options we select the contracts with the largest implied volatility using `nlargest()`
 - Then we pass the value of the strikes and premiums for the same.
7. `iron_condor(call_options, put_options, current_stock_price, stock_prices):`
 - This function calculates the logic for the iron condor strategy.
 - We first identify the ‘n’ nearest OTM contracts for the call and put option using `options_filter`.
 - From the filtered options we select the contracts with the largest implied volatility using `nlargest()` and assign them to the short call and short put.
 - Using a defined ± 5 from short strikes, we identify the long call and long put.
 - Then we pass the value of the strikes and premiums for the same.
8. `iron_butterfly(call_options, put_options, current_stock_price, stock_prices):`
 - This function calculates the logic for the iron butterfly strategy.
 - We first identify the ATM contract for the short call and short put option using `argsort()[1]`
 - We then identify the ‘n’ nearest OTM contracts for the call and put option using `options_filter`
 - From the filtered options, using a defined ± 5 from short strikes, we identify the long call and long put.
 - Then we pass the value of the strikes and premiums for the same
9. `ratio_call_spread(call_options, current_stock_price, stock_prices, ratio):`
 - This function calculates the logic for the ratio call spread strategy..
 - We then identify the ‘n’ nearest OTM contracts for the long call option using `options_filter`
 - From the filtered options, using a defined +5 from short strikes, we identify the short call

OptionsTradingStrategiesProjectReport

Dept. of Finance and Risk Engineering
NYU Tandon School of Engineering

Adjunct Professor, Naresh Malhotra
nm89@nyu.edu

- Then we pass the value of the strikes and premiums for the same.
- The ratio is defined as 2, we sell 2 and buy 1 call option.

10. `calculate_payoff2(stock_prices, current_stock_price, long_call_strike, long_put_strike, long_call_price, long_put_price):`

- This function calculates payoffs for a 2-leg strategy.
- This is done for the long call and the long put. (straddle and strangle)
- It returns a numpy array containing the total payoff for the strategy based on the stock prices at different points.

11. `calculate_payoff4(stock_prices, current_stock_price, long_call_strike, long_put_strike, long_call_price, long_put_price):`

- This function calculates payoffs for a 4-leg strategy.
- This is done for the long call, long put, short call and the short put. (iron condor and iron butterfly)
- It returns a numpy array containing the total payoff for the strategy based on the stock prices at different points.

12. `calculate_ratio_call_spread_payoff(stock_prices, current_stock_price, long_call_price, short_call_price, long_call_strike, short_call_strike, ratio):`

- This function calculates payoffs for the ratio call spread strategy.
- This is done for the long call and the short call.
- It returns a numpy array containing the total payoff for the strategy based on the stock prices at different points.

User Interaction:

- The user is prompted to enter a ticker symbol for the stock they wish to analyze.
- The tool fetches historical stock data and option chain details for the specified ticker symbol using the Yahoo Finance API.
- The user is then provided with a list of available option expiration dates.
- The user is prompted to enter the entry date and expiry date in the format "yyyy-mm-dd".
- The tool validates the user-provided data such as the ticker symbol and dates for proper processing.
- The user is presented with a menu of available options trading strategies such as straddle, strangle, iron condor, iron butterfly, and ratio call spread.
- The user selects a strategy from the menu and enters the required parameters such as strike prices or ratio.
- The tool calculates the potential payoffs for the selected strategy based on the user-defined market scenarios such as bullish, bearish, or choppy.
- The tool visualizes the payoffs using matplotlib, providing a clear representation of potential gains or losses.

- The tool tabulates the results of each strategy for the market scenarios and creates a summary table for the final results.
- The user can save the results as a CSV file for further analysis or reference.

Future Market Scenarios:

1. Volatility as the Basis:

- Volatility is a measure of the degree of variation of a trading price series over time, representing the market's uncertainty or risk.
- The basis for determining market scenarios is the implied volatility of options, which is reflected in the calculated future stock prices.
- Higher implied volatility in the option market may indicate an expectation of larger price swings, contributing to a more bullish or bearish outlook.
- Lower implied volatility may suggest a market with less uncertainty, leading to a choppy or stable scenario.

2. Bullish Scenario:

Definition:

A bullish market scenario implies an expectation of increasing stock prices. It suggests optimism among investors, anticipating positive market trends.

Calculation:

The future stock price for the bullish scenario is estimated by applying a predefined bullish factor (e.g., 1.05, indicating a 5% increase) to the current stock price.

3. Bearish Scenario:

Definition:

A bearish market scenario indicates a pessimistic outlook with an expectation of decreasing stock prices. Investors in a bearish market may anticipate negative market trends.

Calculation:

The future stock price for the bearish scenario is estimated by applying a predefined bearish factor (e.g., 0.95, indicating a 5% decrease) to the current stock price.

4. Choppy Scenario:

Definition:

A choppy market scenario implies a lack of clear direction in stock prices. It reflects market conditions characterized by uncertainty, indecision, or a balanced mix of positive and negative factors.

Calculation:

The future stock price for the choppy scenario remains the same as the current stock price, indicating no significant change.

These future market scenarios are essential for evaluating potential outcomes of various options trading strategies and provide users with valuable insights into the associated risk and reward across different market conditions.

Visualization:

1. Payoff Diagrams:

- The tool creates visual representations of payoff diagrams for each strategy.
- Payoff diagrams show the potential profit or loss for a given options trading strategy at different stock prices.
- The x-axis represents the stock price, and the y-axis represents the profit or loss.
- The payoff diagram shows the maximum profit, maximum loss, and breakeven points for the selected strategy.
- The tool uses matplotlib to create these diagrams, providing a clear representation of potential gains or losses.

2. Final Positions and P&L Calculations:

- The initial position is first calculated.
- The tool calculates the final positions and P&L (profit and loss) for each strategy based on the user-defined market scenarios.
- The P&L calculations show the potential profit or loss for each strategy at the expiry date, based on the final positions and the stock price at that time.
- The tool uses matplotlib to create visual representations of the final positions and P&L calculations, providing a clear representation of potential gains or losses.

3. Summary Table:

- The tool creates a summary table for the final results of each strategy for the market scenarios.
- The summary table shows the potential profit or loss for each strategy in each market scenario.
- The table also shows the maximum profit and loss for each strategy.
- The tool uses tabulate to create this table with the pretty format, providing a clear and concise summary of the potential outcomes for each strategy.

These visualization features are essential for helping users make informed decisions by visualizing potential outcomes and understanding the associated risk and reward for each options trading strategy.

The Strategies:

1. Straddle:

A straddle is an options trading strategy where an investor holds long positions in both a call and a put at the ATM strike and have the same expiration date.

This strategy is designed to profit from significant price movement in either direction, regardless of whether it's bullish or bearish.

Bullish Market:

Options Trading Strategies Project Report

Dept. of Finance and Risk Engineering
NYU Tandon School of Engineering

Adjunct Professor, Naresh Malhotra
nm89@nyu.edu

Long Call becomes profitable while **Long Put** expires worthless. *$Profit = Premium \text{ of Long Call} - Initial \text{ Premium}$*

Bearish Market:

Long Put becomes profitable while **Long Call** expires worthless. *$Profit = Premium \text{ of Long Put} - Initial \text{ Premium}$*

Choppy or Sideways Market:

Both options expire worthless resulting in a loss.

Maximum Profit: Unlimited

Maximum Loss: Capped at the initial premium paid

2. Strangle:

In a strangle, an investor simultaneously buys an out-of-the-money (OTM) call option and an OTM put option with different strike prices, but the same expiration date.

The idea behind a strangle is to profit from significant price movement, but it allows for a wider range of price movement compared to a straddle.

Bullish Market:

Long Call becomes profitable while **Long Put** expires worthless. *$Profit = Premium \text{ of Long Call} - Initial \text{ Premium}$*

Bearish Market:

Long Put becomes profitable while **Long Call** expires worthless. *$Profit = Premium \text{ of Long Put} - Initial \text{ Premium}$*

Choppy or Sideways Market:

Both options expire worthless resulting in a loss.

Maximum Profit: Unlimited

Maximum Loss: Capped at the initial premium paid

3. Iron Condor:

An iron condor is an options trading strategy that involves the simultaneous use of two vertical spreads, one using call options and the other using put options.

The strategy is designed to profit from low volatility in the underlying asset. (**Bear Call Spread + Bull Put Spread**)

It is a market-neutral strategy that benefits from the price of the underlying staying within a certain range.

Bullish Market:

Bull Put becomes profitable while **Bear Call** expires worthless. *$Loss = Bull \text{ Put} - Bear \text{ Call}$*

Bearish Market:

Bear Call becomes profitable while **Bull Put** expires worthless. *$Loss = Bear \text{ Call} - Bull \text{ Put}$*

Choppy or Sideways Market:

All options expire worthless resulting in a profit.

Maximum Profit: Capped at the initial premium received.

Maximum Loss: Capped within range of strikes.

4. Iron Butterfly:

An iron butterfly is an options trading strategy that involves using both calls and puts to create a position with limited risk and limited profit potential.

The strategy profits when the underlying asset's price remains within a specific range. (**Short Straddle + Long Strangle**).

Bullish Market:

Short Straddle becomes profitable while **Long Strangle** expires worthless. $Loss = Short Straddle - Long Strangle$

Bearish Market:

Long Strangle becomes profitable while **Short Straddle** expires worthless. $Loss = Long Strangle - Short Straddle$

Choppy or Sideways Market:

All options expire worthless resulting in a profit.

Maximum Profit: Capped at the initial premium received.

Maximum Loss: Capped within range of strikes.

5. Ratio Call Spread:

A ratio call spread is an options trading strategy that involves an imbalance in the number of long and short call options.

It typically consists of selling more call options than the number of call options being purchased.

The strategy aims to take advantage of a moderate bullish price movement in the underlying asset while still providing some downside protection.

Bullish Market:

Long Call becomes profitable while **Short Call** results in losses. $Loss = Premium (Long Call - Short * Ratio)$

Bearish Market:

Both options expire worthless resulting in a profit..

Choppy or Sideways Market:

Short Call becomes profitable while **Long Call** results in losses.. $Profit = Premium (Short * Ratio - Long Call)$

Maximum Profit: Capped at the initial premium received.

Maximum Loss: Unlimited

These options trading strategies are popular among investors and traders and provide different ways to profit from market movements. The Options Trading Strategies Analyzer allows users to

OptionsTradingStrategiesProjectReport

Dept. of Finance and Risk Engineering
NYU Tandon School of Engineering

Adjunct Professor, Naresh Malhotra
nm89@nyu.edu

analyze these strategies tailored to specific market scenarios and make informed decisions based on potential outcomes.

Conclusion:

The Options Trading Strategies Analyzer is a powerful platform for comprehensive analysis and visualization of diverse options trading strategies. The tool leverages historical stock data and option chain information to deliver a nuanced understanding of potential payoffs for each strategy across varied market scenarios.

The tool provides valuable insights into the risk and reward profiles of each trading strategy, aiding users in assessing potential gains or losses. With user-friendly flexibility and customization options, users can input preferred ticker symbols and dates, tailoring analyses to specific stocks and timeframes.

The tool's modular structure ensures adaptability and positions it for future enhancements and feature additions, fostering continuous improvement. As an educational resource, the project offers hands-on experience in the analysis and visualization of different options trading strategies, contributing to a deeper understanding of the complex dynamics involved.

Future Enhancements:

The project aims to enhance its analytical capabilities and user experience by incorporating real-time data, expanding the range of supported options trading strategies, and improving the user interface.

Real-time data will provide more up-to-date insights, allowing users to make timely decisions in the rapidly changing financial markets.

Expanding the range of supported options trading strategies will provide users with a more comprehensive toolkit for crafting diverse and sophisticated trading approaches.

Improving the user interface aims to create a more user-friendly experience, ensuring that both novice and experienced users can easily navigate and leverage the tool's functionalities.

The references to documentation for key libraries such as yfinance, matplotlib, numpy, and tabulate underscore the project's reliance on well-established and reputable resources, contributing to the robustness and reliability of the implemented functionalities.

OptionsTradingStrategiesProjectReport

Dept. of Finance and Risk Engineering
NYU Tandon School of Engineering

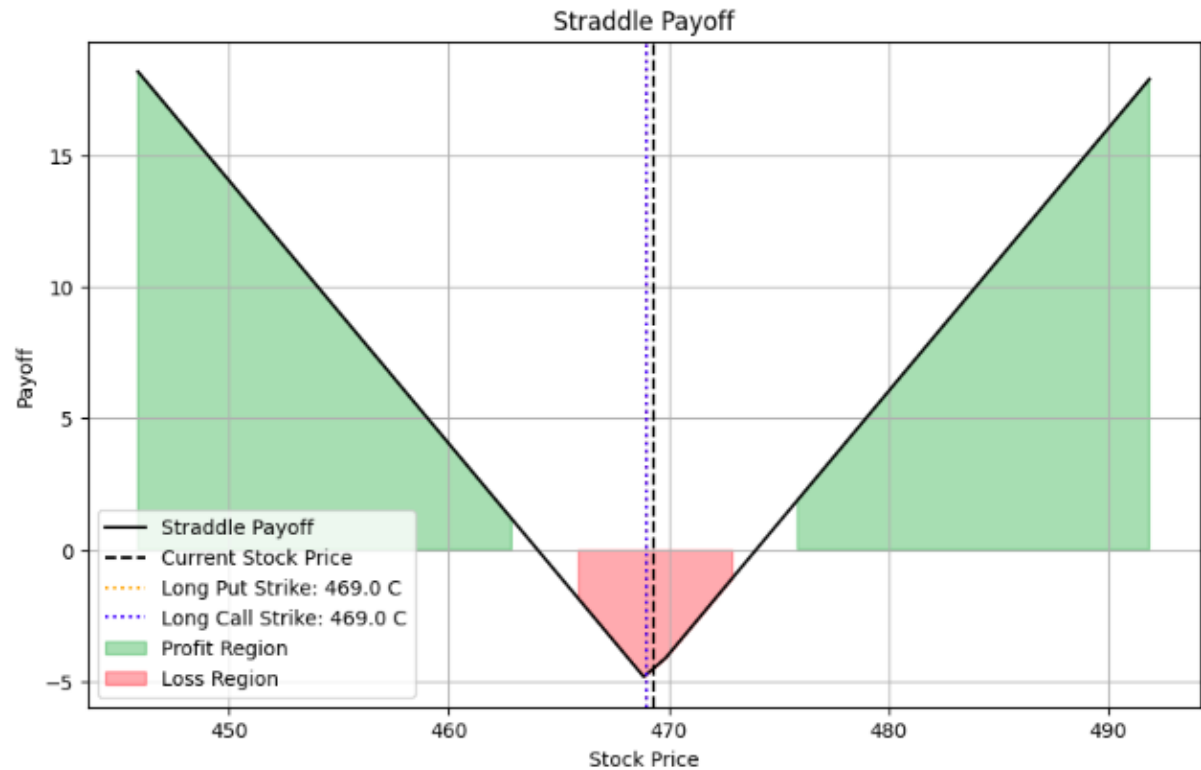
Adjunct Professor, Naresh Malhotra
nm89@nyu.edu

Appendix:

User Input for Option Chain

Enter a ticker symbol: spy
Available Expiration Dates: ['2023-12-18', '2023-12-19', '2023-12-20', '2023-12-21', '2023-12-22', '2023-12-29', '2024-01-05', '2024-01-12', '2024-01-19', '2024-01-26', '2024-01-31', '2024-02-16', '2024-02-29',
Enter the buy date (yyyy-mm-dd): 2023-12-22
Enter the expiry date (yyyy-mm-dd): 2024-01-19

Straddle



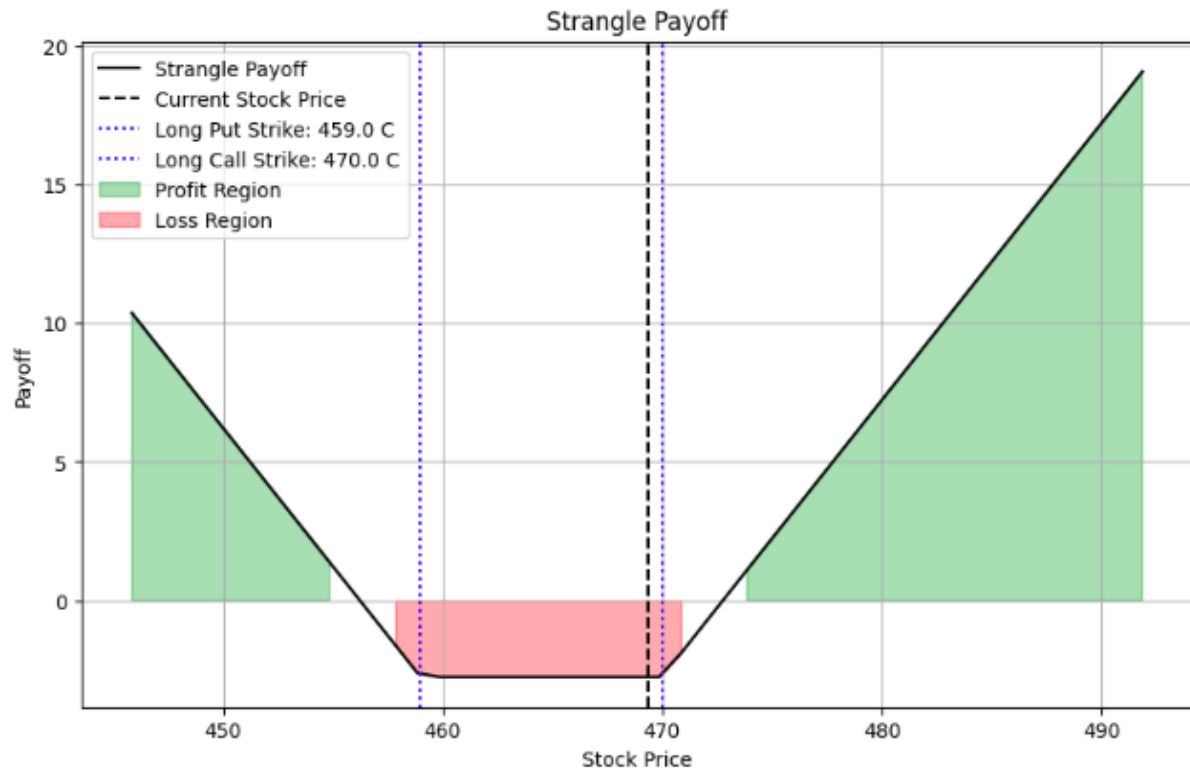
Strategy	Long Call Strike	Long Put Strike	Stock Price	Position	Potential Profit	Maximum Profit	Maximum Loss
Entry	469.0	469.0	469.33	-4.97			
Bullish	469.0	469.0	492.8	26.26	2129.0	Unlimited	497.0
Bearish	469.0	469.0	445.86	23.05	1808.0	Unlimited	497.0
Choppy	469.0	469.0	469.33	0	-497.0	Unlimited	497.0

OptionsTradingStrategiesProjectReport

Dept. of Finance and Risk Engineering
NYU Tandon School of Engineering

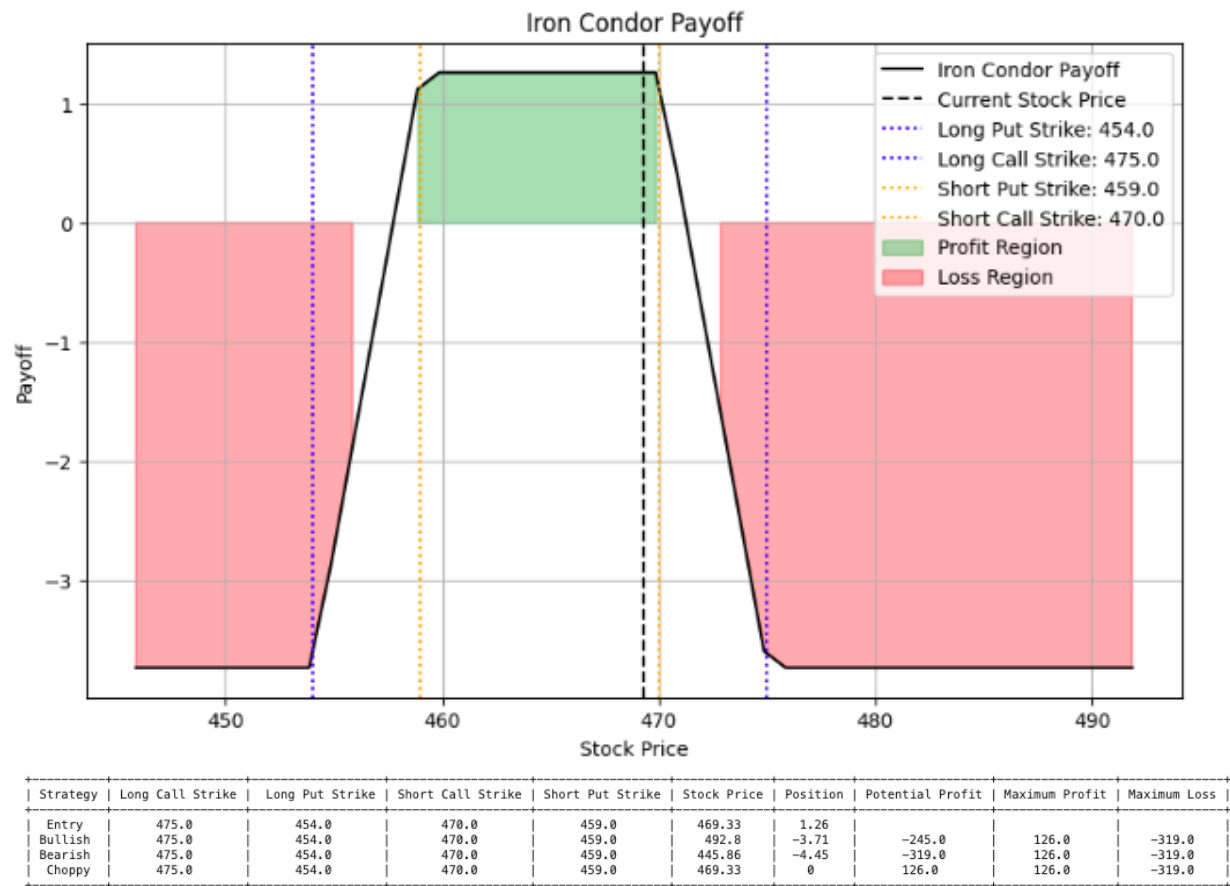
Adjunct Professor, Naresh Malhotra
nm89@nyu.edu

Strangle



Strategy	Long Call Strike	Long Put Strike	Stock Price	Position	Potential Profit	Maximum Profit	Maximum Loss
Entry	470.0	459.0	469.33	-2.77			
Bullish	470.0	459.0	492.8	26.26	2349.0	Unlimited	277.0
Bearish	470.0	459.0	445.86	23.05	2028.0	Unlimited	277.0
Choppy	470.0	459.0	469.33	0	-277.0	Unlimited	277.0

Iron Condor

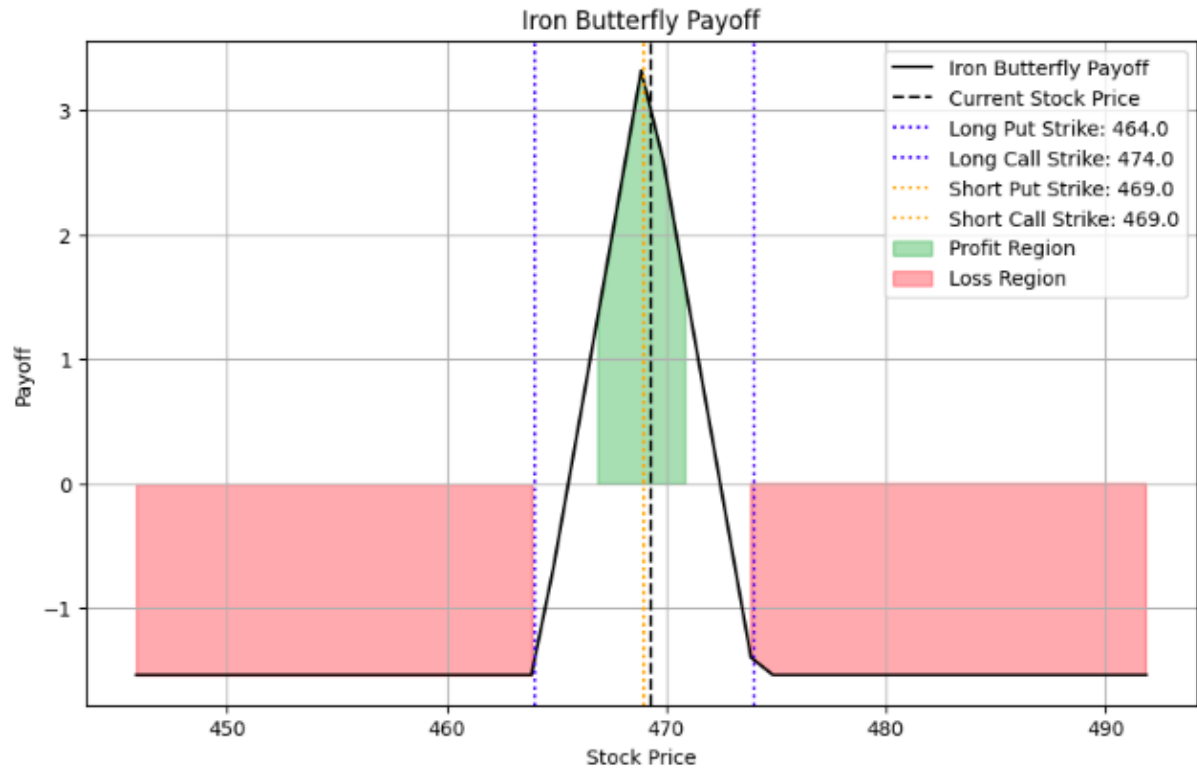


OptionsTradingStrategiesProjectReport

Dept. of Finance and Risk Engineering
NYU Tandon School of Engineering

Adjunct Professor, Naresh Malhotra
nm89@nyu.edu

Iron Butterfly



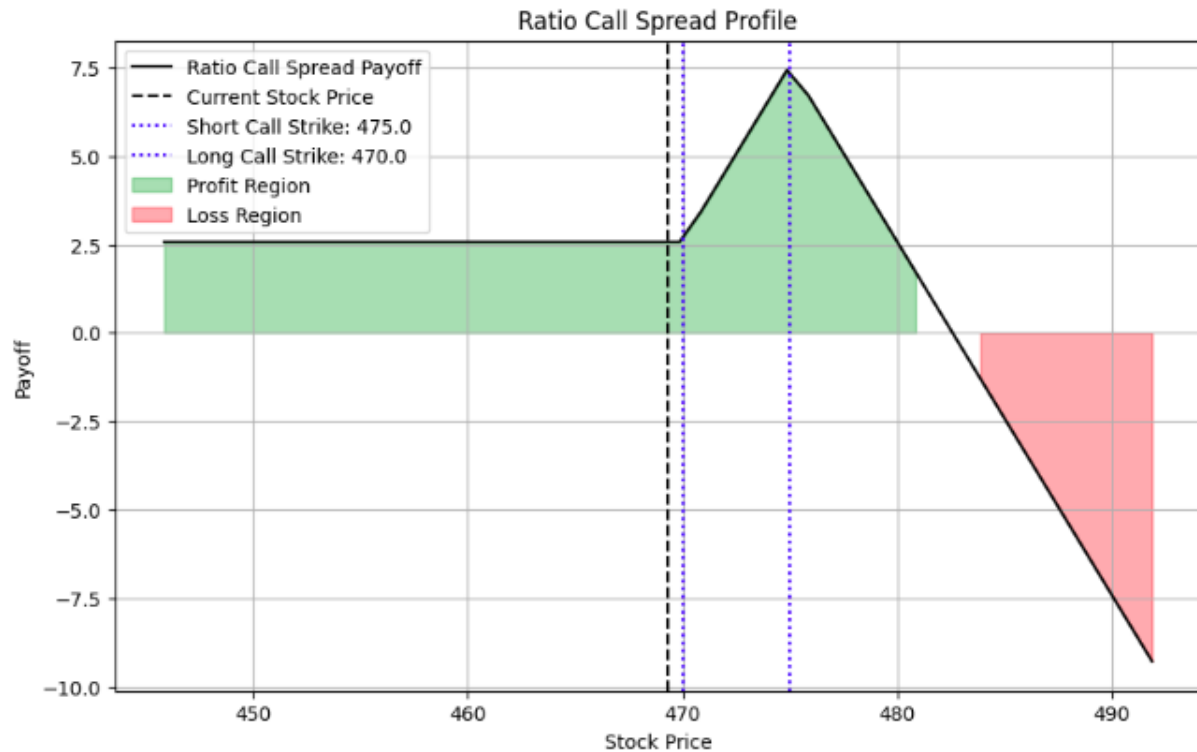
Strategy	Long Call Strike	Long Put Strike	Short Call Strike	Short Put Strike	Stock Price	Position	Potential Profit	Maximum Profit	Maximum Loss
Entry	474.0	464.0	469.0	469.0	469.33	3.46			
Bullish	474.0	464.0	469.0	469.0	492.8	-3.71	-25.0	346.0	-99.0
Bearish	474.0	464.0	469.0	469.0	445.86	-4.45	-99.0	346.0	-99.0
Choppy	474.0	464.0	469.0	469.0	469.33	0	346.0	346.0	-99.0

OptionsTradingStrategiesProjectReport

Dept. of Finance and Risk Engineering
NYU Tandon School of Engineering

Adjunct Professor, Naresh Malhotra
nm89@nyu.edu

Ratio Call Spread



Strategy	Long Call Strike	Short Call Strike	Stock Price	Position	Potential Profit	Maximum Profit	Maximum Loss
Entry	470.0	475.0	469.33	2.57			
Bullish	470.0	475.0	492.8	-26.26	-2369.0	1000.0	Unlimited
Bearish	470.0	475.0	445.86	0	257.0	1000.0	Unlimited
Choppy	470.0	475.0	469.33	7.43	1000.0	1000.0	Unlimited

Summary Table

Strategy	Long Call Strike	Long Put Strike	Short Call Strike	Short Put Strike	Current Stock Price	Bullish Profit	Bearish Profit	Choppy Profit
Straddle	469.0	469.0			469.33	2129.0	1808.0	-497.0
Strangle	470.0	459.0			469.33	2349.0	2028.0	-277.0
Iron Condor	475.0	454.0	470.0	459.0	469.33	-245.0	-319.0	126.0
Iron Butterfly	474.0	464.0	469.0	469.0	469.33	-25.0	-99.0	346.0
Ratio Call Spread	470.0		475.0		469.33	-2369.0	257.0	1000.0