Quantitative Analysis and Strategies in Options and Equity Markets

Statistical modeling, backtesting, and coding-driven experiments in derivatives pricing, volatility, and stock dynamics

Adam Kainikara

Email: adamkainikara@gmail.com
LinkedIn: linkedin.com/in/adam-kainikara-12697020a
GitHub: github.com/adamkai7/projects

Introduction & Author Background

The following document is a report outlining independent data analysis and research in quantitative finance that I conducted over a large portion of the summer (June – September 2025), after completing my first year of undergraduate studies at the University of California, Davis. Entering my sophomore year, I currently have junior standing and am majoring in Mathematics and Scientific Computation, with plans to either double major in Computer Science and Engineering or complement my degree with double minors in Computer Science and Economics and/or a physical science. All work presented here, from coding and data analysis to writing, was completed independently by me.

My interest in quantitative and mathematical finance, as well as data analysis, has developed over the past three years alongside a growing curiosity in machine learning, artificial intelligence, and algorithms—applied across fields such as finance, healthcare, and computer vision. I also have practical experience, including familiarity with the Thinkorswim trading platform and its scripting language, Thinkscript. This project marked an important step in transitioning from informal discussions to formally applying financial concepts and analyzing data.

Through the process of completing this research and writing this report, I developed important skills in Python programming, data analysis, and quantitative/statistical modeling. Equally important, producing this document of more than fifty pages gave me practice in technical writing and communication, an area I have not always been confident in but one where I made substantial progress through this project.

All analyses were conducted in Python, and unless otherwise specified, market data was scraped from Yahoo Finance. For this project, all referenced code and data files can be found in the code/ and results/ folders within the main GitHub repository. This work reflects my long-term interest in learning and applying mathematical methods to various domains, and my ongoing effort to build the technical and analytical skills needed for a career. My technical experience also includes building machine learning models using scikit-learn for data mining applications across various domains. One such project, an analysis of health data for a Kaggle contest, achieved a strong result and the accompanying report is available for review on my GitHub.

Abstract

This research presents a quantitative framework developed to analyze equities, option pricing, risk dynamics, systematic trading strategies, and algorithms. The initial report is based on the Black-Scholes-Merton

model, which was utilized to conduct large-scale simulations of option return distributions under both static and dynamic implied volatility regimes. The initial studies systematically evaluated thousands of trade scenarios across varying strikes, maturities, and underlying price paths to quantify the drivers of extreme profit and loss.

The project extends beyond classical options by integrating advanced quantitative methods. These include the implementation of rolling regression models to estimate time-varying beta and momentum factors, correlation matrix analysis to map market structure, and the application of Differential Evolution, a stochastic optimization algorithm, to identify highly correlated asset clusters. Further analyses explored seasonality in equity returns using Kernel Density Estimation and incorporated alternative macroeconomic data, such as TSA passenger volumes, to contextualize market behavior.

This report and code is meant to establish a rigorous, multi-disciplinary approach to quantitative research by bridging financial data analysis with statistical modeling and computational optimization.

Contents

1. The Black-Scholes Model: Foundations of Modern Derivatives Pricing and the Greeks

1.1. **Overview and Significance.** The Black-Scholes Model (also known as the Black–Scholes–Merton model) is a mathematical framework for pricing European options and derivative investments. It is based on a partial differential equation that estimates option prices by incorporating factors such as the underlying asset's price, volatility, time to expiration, interest rates, and dividends.

Formally, the equation is the following:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

where:

- $V(S,t)$ is the option price as a function of stock price S and time t
- $\frac{\partial V}{\partial t}$ is the rate of change of the option value with respect to time (time decay)
- $\frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}$ represents the effect of stock price volatility on the option value
- $(r-q)S\frac{\partial V}{\partial S}$ captures the sensitivity of the option price to changes in the underlying stock, adjusted for dividend yield q
- $-rV$ reflects the effect of discounting at the risk-free rate r

The model's core insight is that options can be perfectly hedged through dynamic trading of the underlying asset and risk free bonds, eliminating arbitrage opportunities and yielding a unique theoretical price. This principle transformed derivatives from speculative instruments to sophisticated risk management tools used by institutions worldwide.

The Black–Scholes–Merton model was developed in the the late 1960s to early 1970s by Fischer Black, Myron Scholes, and Robert Merton. In 1997, Scholes and Merton were awarded the Nobel Prize in Economics for this work in option pricing. Black passed away in 1995 and was not eligible as the prize is not awarded posthumously.

1.2. **The Greeks.** The practical application of Black-Scholes extends beyond pricing through the **"Greeks"** - partial derivatives that measure option price sensitivities to various market factors. These metrics are essential for portfolio risk management and hedging strategies.

There are several key variables in the Black-Scholes model:

- S is the current stock price
- K is the strike price
- r is the risk-free interest rate
- q is the dividend yield
- t is the time to expiration
- σ is the volatility of the stock

The Greeks are the first and second-order partial derivatives of the Black-Scholes formula. They measure how the option price V responds to changes in different variables.

1.3. **Delta (Δ) — Sensitivity to Price.** Delta measures the rate of change of the option price with respect to the underlying stock price.

$$\Delta = \frac{\partial V}{\partial S}$$

For call options, delta ranges from 0 to 1, while put options range from -1 to 0. A delta of 0.5 indicates that for every $1 increase in the underlying stock, the option price increases by approximately $0.50. Portfolio managers use delta to create delta neutral positions, where overall portfolio value remains relatively stable despite small price movements in the underlying assets.

1.4. **Gamma (Γ) — Sensitivity of Delta.** Gamma measures how Delta changes as the stock price moves.

$$\Gamma = \frac{\partial^2 V}{\partial S^2}$$

High gamma positions require frequent rebalancing to maintain delta neutrality. Gamma is highest for at-the-money options approaching expiration, creating both opportunity and risk for traders.

1.5. **Theta (Θ) — Time Decay.** Theta measures how much the option price decreases per unit of time.

$$\Theta = \frac{\partial V}{\partial t}$$

Options lose value as expiration nears. Option sellers benefit from positive theta, while buyers face the constant headwind of time decay. Understanding theta is important for timing strategies and managing the profitability of option positions.

1.6. **Vega (ν) — Sensitivity to Volatility.** Vega measures how an option price changes with respect to volatility:

$$\nu = \frac{\partial V}{\partial \sigma}$$

Higher Vega means the option price is more sensitive to changes in implied volatility. During market stress periods, implied volatility typically increases, benefiting long vega positions.

1.7. **Rho (ρ) — Sensitivity to Interest Rate.** Rho measures how the option price changes with the risk-free interest rate, which is usually the two-year treasury rate.

$$\rho = \frac{\partial V}{\partial r}$$

While often the least significant Greek in normal market conditions, rho becomes important for long dated options and during periods of significant interest rate changes. Recent Federal Reserve policy shifts (with changing interest rates) have renewed focus on rho management in portfolios.

These risk metrics form the foundation of trading strategies employed by hedge funds, investment banks, proprietary trading firms, and ordinary people. Portfolio construction often involves balancing multiple Greeks to achieve desired risk profiles while maximizing expected returns. The mathematical rigor of the Black-Scholes framework demonstrates how quantitative methods drive modern financial markets.

2. Risk Profiles of Options and Derivatives

A **risk profile** is a graph that illustrates the potential profit and loss of a financial position at many different underlying asset prices. The horizontal axis represents the asset price, while the vertical axis represents profit or loss. Risk profiles often resemble piecewise-linear functions: for certain price ranges, the slope may be 0 (no change in profit/loss), and beyond certain points, the slope becomes 1 (profit increases linearly with stock price). However, prior to expiration, the curve is smoother due to time decay (theta) and changes in implied volatility. It only becomes perfectly piecewise-linear at expiration.

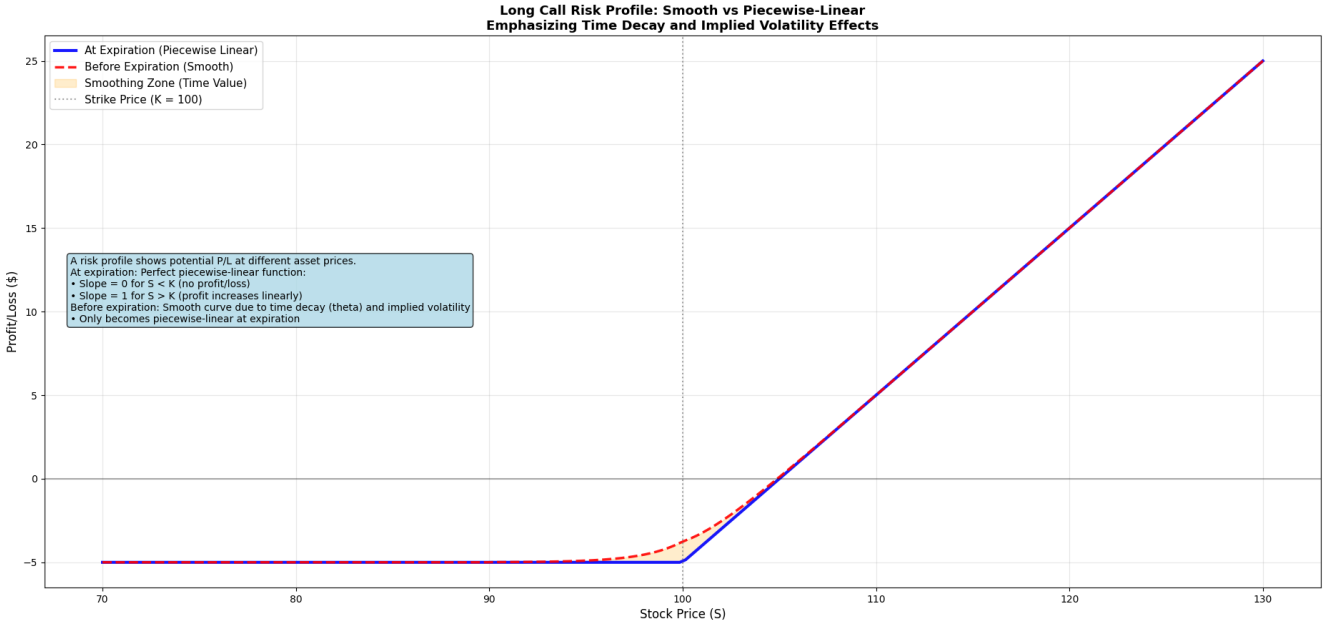

Figure 2.1. Long Call Risk Profile: Piecewise-Linear at Expiration vs Smooth Before Expiration.

Highlights the effect of time decay (θ) and implied volatility. Before expiration, the P&L curve is smooth around the strike price; at expiration, it becomes perfectly piecewise-linear.

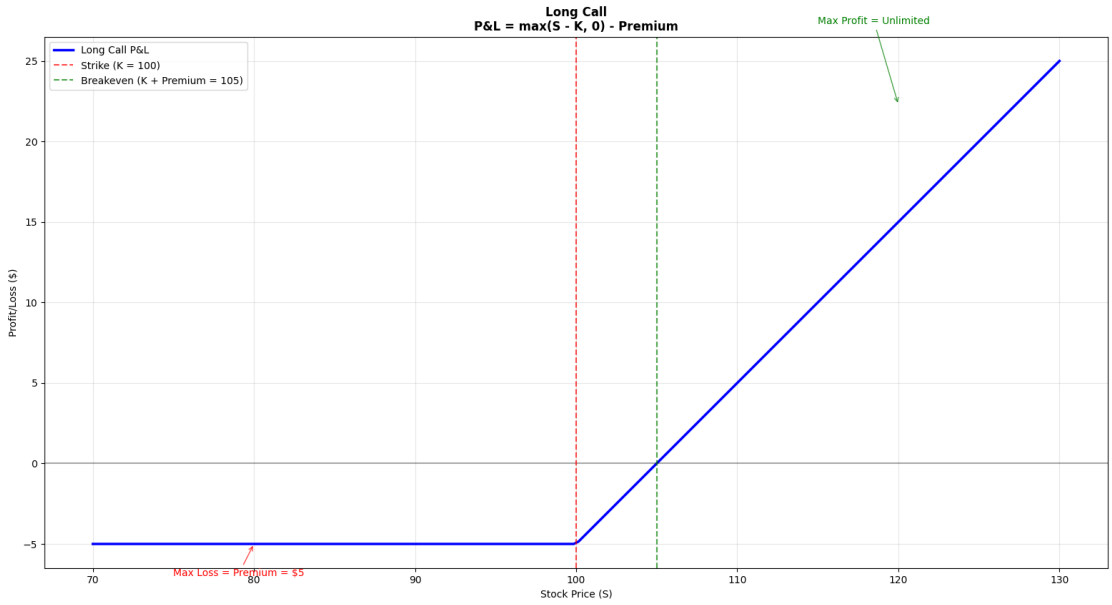

FIGURE 2.2. Long Call Risk Profile

2.1. **Risk Profile of a Long Call.** A Long Call is an option purchased for a premium, granting the right to buy the stock at a strike price (k)

- Break even point: $k +$ premium paid
- Maximum profit: theoretically unlimited (profit grows linearly as stock price rises)
- Maximum loss: limited to the premium paid

The profit/loss function is:

$$\text{profit/loss} = \max(s - k, 0) - \text{premium}$$

where (s) is the stock price.

- If $s < k$, the option expires worthless, and the loss is the premium
- If $s > k$, profit increases linearly beyond the break even point

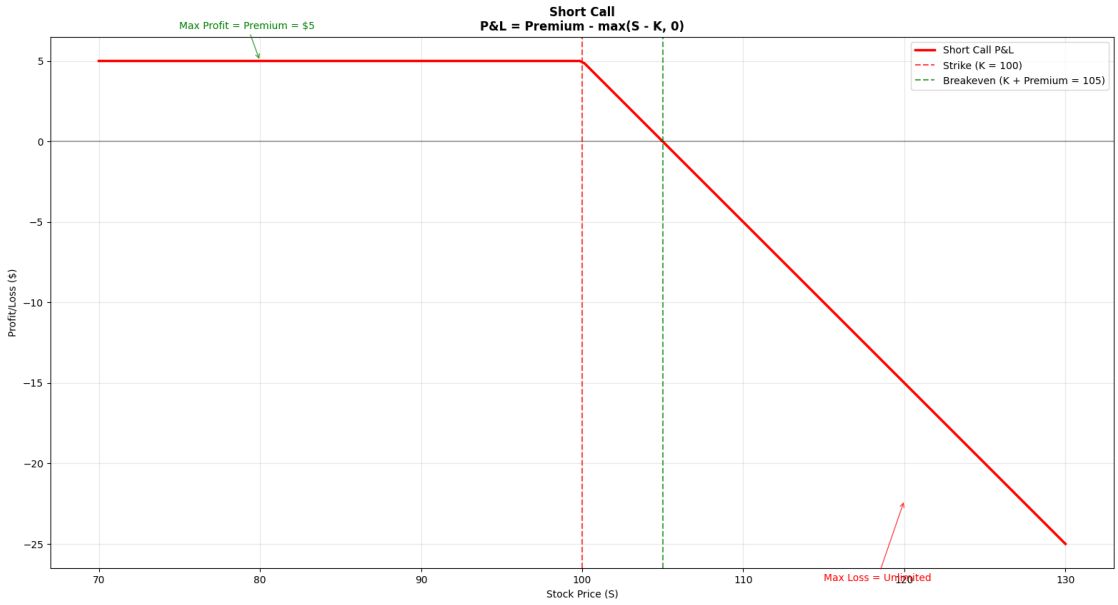

FIGURE 2.3. Short Call Risk Profile

2.2. **Risk Profile of a Short Call.** A Short Call is the opposite of a Long Call, where the seller collects a premium.

- Break even: $k +$ premium received
- Maximum profit: limited to premium collected
- Maximum loss: theoretically unlimited

The profit/loss function is:

$$\text{profit/loss} = \text{premium} - \max(s - k, 0)$$

- If $s < k$, the call expires worthless, and the seller keeps the premium
- If $s > k$, the seller experiences increasing losses as the stock price rises

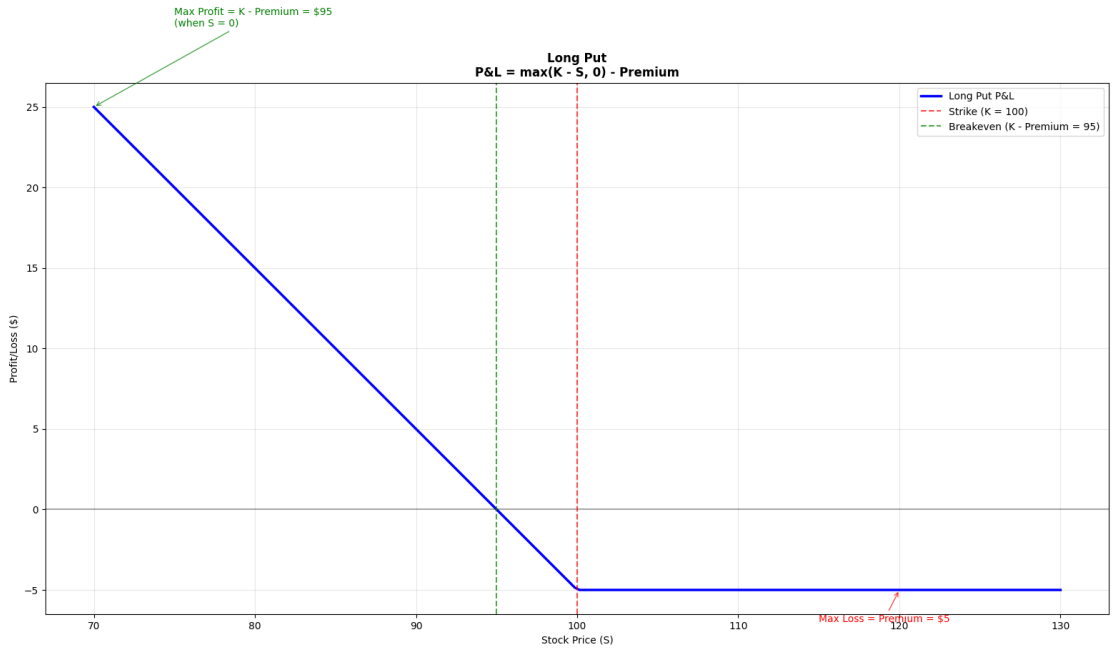

FIGURE 2.4. Long Put Risk Profile

2.3. **Risk Profile of a Long Put.** A Long Put benefits when the stock price falls. The investor pays a premium for the right to sell at strike price (k)

- Break even point: $k -$ premium paid
- Maximum profit: $k -$ premium (if stock falls to zero)
- Maximum loss: limited to the premium

The profit/loss function is:

$$\text{profit/loss} = \max(k - s, 0) - \text{premium}$$

- If $s > k$, the put expires worthless; the loss is the premium
- If $s < k$, profit grows as the stock falls, reaching maximum when $s = 0$

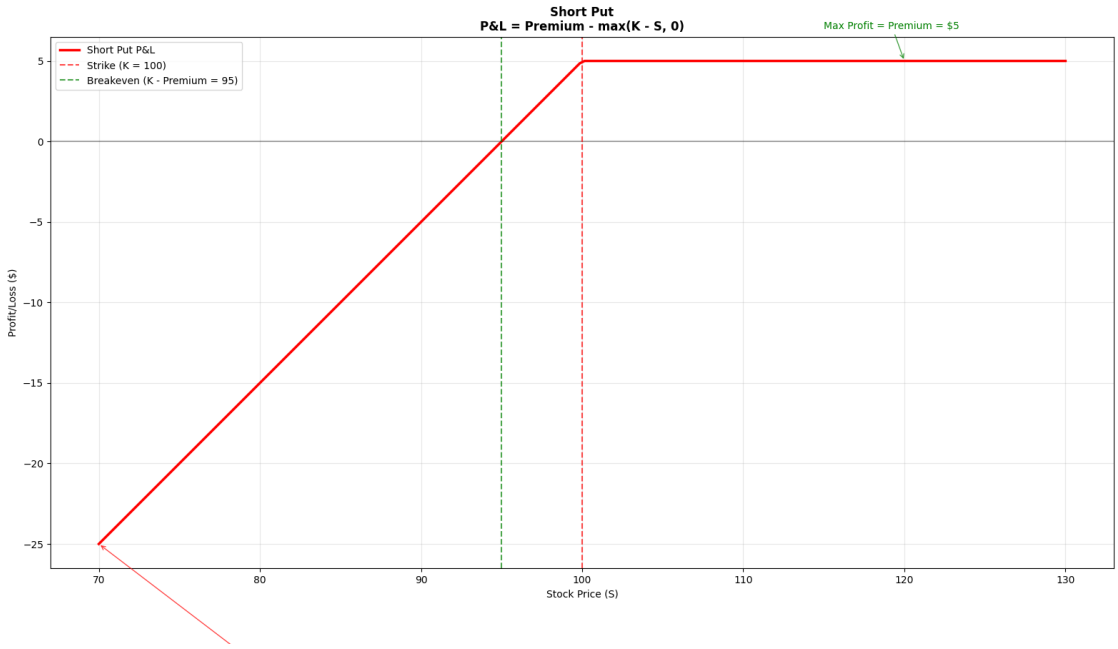

FIGURE 2.5. Short Put Risk Profile

2.4. **Risk Profile of a Short Put.** A Short Put is the opposite of a Long Put, where the seller receives a premium upfront

- Break even point: $k -$ premium received
- Maximum profit: premium received
- Maximum loss: $k -$ premium received (if stock falls to zero)

The profit/loss function is:

$$\text{profit/loss} = \text{premium} - \max(k - s, 0)$$

- If $s > k$, the put expires worthless, and the seller keeps the premium
- If $s < k$, the seller loses as the stock price falls further

3. Optimizing Option Returns Under Fixed Implied Volatility

This analysis relates to the file called **constant_iv_analysis.py**

3.1. **Scenario.** This code analyzes the performance of an options trading strategy by calculating the best and worst returns achievable under a set of controlled conditions. The strategy focuses on buying and selling call options using the Black-Scholes pricing model, while keeping implied volatility (IV) and risk free rate (r_f) fixed and varying other parameters of the equation. The varying parameters are strike (K), final underlying price (S_t), and days to expiry (DTE). Return is calculated by $\frac{\text{Price sell - Price buy}}{\text{Price buy}}$. The trading simulation involves buying an option at a given DTE and selling it after a fixed holding window period of time ($wsize$), while varying the underlying price and strike level to evaluate their effect on returns. For example, if we bought an option that has $dte = 50$ and we set $wsize = 10$, then the option would be sold 10 days after buying it (ie: $dte = 40$).

This analysis was done to confirm a hypothesis that high positive returns when buying call options occur when buying options far out of the money (OTM) and selling them far into the money (ITM) and major negative losses occur with options bought far ITM and sold far OTM. While to someone who knows options this may sound intuitive, it is important to confirm this hypothesis. We want to quantify and determine which combinations of strike prices, underlying price movements, and DTE lead to the most and least profitable option trades, under the simplified assumption of constant volatility and risk-free rate. This experiment helps analyze the behavior of option returns and provides insights into how different market movements and option choices impact profitability.

3.2. **Experimentation.** In my experimentation, I used the exchange traded fund (ETF) QQQ which at the time had an initial price of $s_0 = 556.22$. To model future price movements, a price band of 15% was chosen ($S_f \in [0.85 \times 556.22, 1.15 \times 556.22]$). The strikes considered where $K \in [400, 700]$. This range was chosen because these strikes exhibited reasonable open interest levels on the Thinkorswim trading platform for practical relevance and liquidity. The maximum DTE was set to 100 days and $wsize$ was 10 days. To avoid trades that are not realistically actionable or are likely to be illiquid, a minimum price floor of \$0.35 was applied to both the buy and sell option prices. This helps eliminate extremely low-priced contracts that could distort return calculations due micro jumps or low liquidity. The user can adjust to any ticker of their choosing as long as the parameters are inputted.

The function `analyze_option_strategy()` begins by setting inputs like the underlying price, strike price range, days to expiry, implied volatility, risk-free rate, dividend yield, and a price floor to exclude illiquid options. The main analysis occurs through a series of nested loops: for each simulated final underlying price within a range of ±15% of the starting price, the code iterates over a range of strike prices and holding periods. For each combination, it computes option prices at both the entry and exit points using the Black-Scholes model. The percentage return for each simulated trade is then calculated based on these prices. Finally, all results are visualized through scatter plots, providing a clear picture of how returns vary with strike prices, holding periods, and final underlying prices. This process allows for comprehensive exploration of option price behavior and return profiles under different market scenarios.

Across over 13,000 simulated trades, option returns ranged from –99.3 % to 6000+ %. The lower losses are capped at -99.3% due to the limits applied from the experiment meaning total losses of

-100% do not occur due to the floor mechanism and the limited premium paid. Meanwhile, gains can be multiples of the initial investment.

3.3. **Analysis and Results.**

Table 1: Top 20 Scenarios (Highest Returns)

Rank	K	BuyDTE	Initial S	Final S	Price Buy	Price Sell	Moneyness	Und. Return	Opt. Return
1	605	20d	556.22	639.65	0.52	35.71	OTM	15.00%	6752.51%
2	625	30d	556.22	639.65	0.37	21.80	OTM	15.00%	5755.99%
3	605	20d	556.22	634.09	0.52	30.49	OTM	14.00%	5750.87%
4	600	20d	556.22	639.65	0.78	40.51	OTM	15.00%	5104.63%
5	605	20d	556.22	628.53	0.52	25.48	OTM	13.00%	4789.30%
6	625	30d	556.22	634.09	0.37	18.04	OTM	14.00%	4746.62%
7	620	30d	556.22	639.65	0.52	25.38	OTM	15.00%	4744.37%
8	600	20d	556.22	634.09	0.78	35.15	OTM	14.00%	4416.63%
9	640	40d	556.22	639.65	0.36	15.95	OTM	15.00%	4369.72%
10	620	30d	556.22	634.09	0.52	21.31	OTM	14.00%	3966.72%
11	615	30d	556.22	639.65	0.73	29.24	OTM	15.00%	3911.90%
12	605	20d	556.22	622.97	0.52	20.76	OTM	12.00%	3883.90%
13	595	20d	556.22	639.65	1.14	45.39	OTM	15.00%	3882.96%
14	625	30d	556.22	628.53	0.37	14.66	OTM	13.00%	3838.05%
15	635	40d	556.22	639.65	0.48	18.60	OTM	15.00%	3774.78%
16	600	20d	556.22	628.53	0.78	29.94	OTM	13.00%	3747.16%
17	640	40d	556.22	634.09	0.36	13.17	OTM	14.00%	3590.50%
18	595	20d	556.22	634.09	1.14	39.95	OTM	14.00%	3405.31%
19	615	30d	556.22	634.09	0.73	24.87	OTM	14.00%	3313.09%
20	630	40d	556.22	639.65	0.64	21.51	OTM	15.00%	3259.36%

The best-performing trades are characterized by:

- Far out of the money (OTM) strikes at the time of purchase
- Short-term maturities (DTE of 20 to 30 days)
- Large favorable underlying price movements of underlying price increasing. For the experiment, underlying price movements where capped at ±15% and the top performers had an underlying movement of +13% to +15% within the 10-day window.
- Low initial premium cost, which allows significant leverage on a small upfront investment
- Sell/Expiring far into the money (ITM)

Table 2: Bottom 20 Scenarios (Lowest Returns)

Rank	K	BuyDTE	Initial S	Final S	Price Buy	Price Sell	Moneyness	Und. Return	Opt. Return
1	500	20d	556.22	472.79	57.36	0.38	ITM	-15.00%	-99.33%
2	505	20d	556.22	478.35	52.46	0.44	ITM	-14.00%	-99.17%
3	515	30d	556.22	472.79	44.11	0.41	ITM	-15.00%	-99.07%
4	510	20d	556.22	483.91	47.61	0.49	ITM	-13.00%	-98.96%
5	530	40d	556.22	472.79	33.15	0.35	ITM	-15.00%	-98.94%
6	520	30d	556.22	478.35	39.66	0.46	ITM	-14.00%	-98.84%

Rank	K	BuyDTE	Initial S	Final S	Price Buy	Price Sell	Moneyness	Und. Return	Opt. Return
7	495	20d	556.22	472.79	62.30	0.74	ITM	-15.00%	-98.82%
8	530	20d	556.22	500.60	29.34	0.38	ITM	-10.00%	-98.72%
9	515	20d	556.22	489.47	42.84	0.56	ITM	-12.00%	-98.70%
10	535	30d	556.22	489.47	27.43	0.36	ITM	-12.00%	-98.69%
11	535	40d	556.22	478.35	29.43	0.39	ITM	-14.00%	-98.67%
12	510	30d	556.22	472.79	48.69	0.66	ITM	-15.00%	-98.64%
13	525	40d	556.22	472.79	37.07	0.52	ITM	-15.00%	-98.59%
14	500	20d	556.22	478.35	57.36	0.82	ITM	-14.00%	-98.56%
15	540	50d	556.22	472.79	27.89	0.40	ITM	-15.00%	-98.56%
16	525	30d	556.22	483.91	35.38	0.51	ITM	-13.00%	-98.55%
17	520	20d	556.22	495.04	38.18	0.63	ITM	-11.00%	-98.36%
18	515	30d	556.22	478.35	44.11	0.73	ITM	-14.00%	-98.34%
19	540	40d	556.22	483.91	25.94	0.44	ITM	-13.00%	-98.32%
20	535	20d	556.22	506.16	25.25	0.43	ITM	-9.00%	-98.32%

The worst performing trades are mostly the opposite:

- Far in-the-money (ITM) strikes at the time of purchase
- Short-term maturities (DTE of 20 to 30 days)
- Large unfavorable underlying price movements of underlying price decreasing
- High initial premium cost, and expiring with little to no cost leading to a significant loss on investment.
- Expiring far out of the money

  From these results, it may be tempting to assume that investors should buy cheap options, such as those priced at only a few cents, with far OTM strikes, hoping for a large market movement and selling once the option is deep in the money over a short time frame due to option decay. At the point of sale, the option may be worth several dollars, resulting in a high percentage return. While this strategy can generate the highest returns, it should be noted that buying extremely far OTM options does not always lead to profitable outcomes. These exceptionally high returns depend on significant underlying price movements, which occur with low probability. Most of the time, buying cheap options results in losses because the premium is paid regardless, and the option can expire worthless. Profit only arises when there is a substantial jump in the underlying asset, which, according to the assumed normal distribution, is unlikely.Investors have a higher probability of making money by purchasing options that are lower OTM but still relatively close to the money and selling after smaller movements that bring the option into the money. While the returns are lower (10–40%), the likelihood of making a profit is much higher than with low probability, high-return options (300%+). Additionally, buying options near the money may cost more, but they are more likely to expire or be sold near the strike price, allowing investors to recover at least part of their investment if the market moves against them.

  Investors who pursue these low probability, high-return strategies, such as Nassim Taleb in The Black Swan, accept the potential for months or even years of losses, with the expectation of earning a huge profit when a rare, large market movement occurs. However, real world implementation of such strategies faces additional challenges including wider bid ask spreads on far OTM options, transaction costs that disproportionately impact low priced contracts, or liquidity constraints.

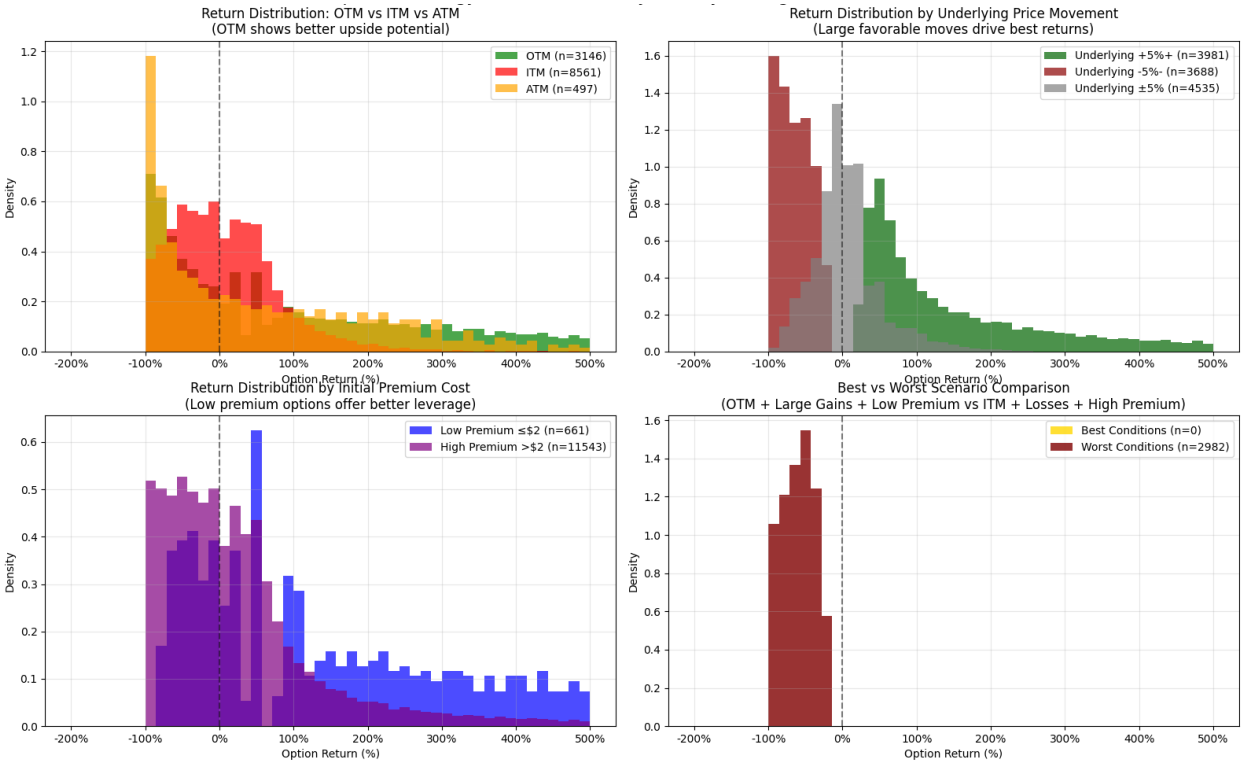

FIGURE 3.1. Distribution of Option Returns by Key Trading Characteristics

The four-panel histogram analysis (Figure 3.1) provides evidence for the key findings regarding optimal call option trading characteristics.

Panel A: Moneyness Impact (Top Left) - This figure shows the different distribution shapes between option moneyness categories. ATM options (orange, n=497) show an extremely concentrated distribution with a sharp peak around -80% to -100% returns, indicating these options consistently expire nearly worthless with losses due to the premium paid. ITM options (red, n=8561) display a broader but still predominantly negative distribution centered around -60% to 60% returns. In contrast, OTM options (green, n=3146) exhibit a much flatter, more spread-out distribution with higher probability density in positive return regions, particularly beyond +100

Panel B: Underlying Movement Impact (Top Right) - This figure provides the clearest validation of how underlying movements affect returns. The distributions show a complete reversal based on underlying price direction. When underlying prices decline (-5% or more, red), options show extremely concentrated losses around -80% to -100%. Conversely, when underlying prices rise significantly (+5% or more, green), the distribution shifts rightward with substantial density extending into +20% to +400% returns. The neutral underlying movement (gray) falls between these extremes, demonstrating that large favorable underlying moves are indeed the primary driver of option returns.

Panel C: Premium Cost Impact (Bottom Left) - This figure shows the leverage effect of low-cost options. High premium options (>$2, purple, n=11543) show a concentrated distribution heavily

weighted toward losses, with peak density around -60% returns. Low premium options (<$2, blue, n=661) demonstrate a different profile with much higher probability density in positive return regions and a flatter distribution extending well into the +200% to +500% range. This validates our finding that low initial premium costs enable superior leverage opportunities. This, however, reflects that low option contracts can lead to exceptional returns even if the probability is low.

Panel D: Combined Scenario Analysis (Bottom Right) - This figure shows that the most telling result is that no trades met our "best conditions" criteria (n=0), indicating that the combination of OTM + large gains + low premium was extremely rare in our simulation parameters (meaning that the probability is low). However, the "worst conditions" (ITM + losses + high premium, maroon, n=2982) show the expected concentrated distribution of severe losses around -80% returns due to premium paid, confirming that these characteristics can lead to poor outcomes.

Below is selected rows and their return scenarios. For the complete list (14,000+ results), please see the file called: **fixed_iv_analysis_results.txt**

Table 3: Sample of More Probabilistic Positive Return Scenarios

Rank	K	BuyDTE	Initial S	Final S	Price Buy	Price Sell	Moneyness	Und. Return	Opt. Return
5038	520	20d	556.22	572.91	38.18	53.46	ITM	3.00%	40.00%
5280	490	90d	556.22	584.03	73.03	98.59	ITM	5.00%	35.00%
5508	430	40d	556.22	595.16	127.96	166.44	ITM	7.00%	30.08%
5746	450	60d	556.22	584.03	109.02	136.30	ITM	5.00%	25.02%
6073	530	60d	556.22	567.34	36.55	43.86	ITM	2.00%	20.00%
6341	490	40d	556.22	567.34	68.62	78.91	ITM	2.00%	15.00%
6567	455	70d	556.22	567.34	104.62	115.16	ITM	2.00%	10.07%
6866	455	20d	556.22	561.78	102.15	107.25	ITM	1.00%	4.99%

Table 4: Sample of More Probabilistic Negative Return Scenarios

Rank	K	BuyDTE	Initial S	Final S	Price Buy	Price Sell	Moneyness	Und. Return	Opt. Return
3613	500	70d	556.22	528.41	62.19	37.30	ITM	-5.00%	-40.02%
3916	515	90d	556.22	533.97	52.10	33.86	ITM	-4.00%	-35.00%
4209	500	100d	556.22	533.97	65.39	45.75	ITM	-4.00%	-30.03%
4502	425	80d	556.22	522.85	134.67	100.99	ITM	-6.00%	-25.00%
4786	420	100d	556.22	528.41	140.48	112.39	ITM	-5.00%	-20.00%
5063	445	60d	556.22	539.53	113.96	96.86	ITM	-3.00%	-15.01%
5343	625	60d	556.22	561.78	2.17	1.95	OTM	1.00%	-9.99%
5613	440	90d	556.22	550.66	120.37	114.35	ITM	-1.00%	-5.00%

4. Optimizing Option Returns Under Increasing Implied Volatility

This analysis relates to the file called **variable_iv_analysis.py**

4.1. **Scenario.** Building on the methodology from Section 3, this section investigates the effect of increasing implied volatility (IV) on option returns. In particular, we simulate scenarios where IV rises between the time of purchase and the time of sale, which is often observed in the days leading up to significant events such as earnings announcements. Implied volatility represents the market's expectation of future asset price fluctuations, and an increase in IV tends to increase the price of options which could lead to potential gains for the option holder.

The strategy continues on buying call options but now accounts for a jump in IV. The purchase occurs at a baseline IV (iv_buy) and the sale occurs at a higher IV (iv_sell) A range of iv_sell was generated using the equations iv_sell_max = iv_buy $*$ (1+iv_jump) and iv_sell_levels = arrange(iv_buy, iv_sell_max + 1e-6, 0.01). As before, the analysis varies strike prices, days to expiration (DTE), and final underlying prices to determine how these factors interact with IV increases to produce optimal returns. Return is calculated as: $\frac{\text{Price sell - Price buy}}{\text{Price buy}}$.

This scenario allows us to quantify the additional impact of volatility increase on returns and identify which option characteristics are most sensitive to IV increases.

4.2. **Experimentation.** In my experimentation, I used the exchange traded fund (ETF) QQQ which at the time had an initial price of $s_0 = 556.22$. To model future price movements, a price band of 10% was chosen ($S_f \in [0.9 \times 556.22, 1.1 \times 556.22]$). The strikes considered where varied from ± 10 the initial price to capture both ITM and OTM options. The holding window (rwsize) was set to 20 days, with a holding period of 3 days. This means that we would sell the option within 3 days of the event we where anticipating (such as earnings). The purchase IV was set at iv_buy = 20% with a possible jump of iv_jump = 50% at sale. A minimum option price floor of \$0.10 was applied to prevent illiquid or near zero priced contracts from distorting results.

The function `analyze_IV_sell_strategy()` begins by setting the main inputs, including the initial underlying price (s0), the percentage change range for final prices (percent_change), the holding window (rwsize), DTE extension (dte_window), strike price range (strike_pct), initial implied volatility (iv_buy), potential IV jump (iv_jump), risk free rate (rf), dividend yield (q), and a price floor (price_floor) to exclude illiquid options.

The function then defines the minimum and maximum strike prices and generates a list of possible final underlying prices within the specified percentage change. It also calculates a range of possible sale IV levels from iv_buy up to iv_buy * (1 + iv_jump) in increments of 0.01, allowing the simulation to capture the effect of increasing implied volatility. The main analysis occurs through nested loops: for each sale IV, each final underlying price, each strike price, and each DTE in the holding period window, the function calculates the option purchase price using the Black-Scholes model and the sale price as the intrinsic value at the final price. Only trades with prices above the floor are considered, and the percentage return is calculated and stored along with all trade parameters. Over 30,000 results are produced.

4.3. **Analysis and Results.**

Table 5: Top 20 Positive Return Scenarios with IV Jump

Rank	Strike	DTE	Final S	IV Buy → IV Sell	Price Buy	Price Sell	Return (%)
1	599	21	611.842	0.200→0.220	0.813	12.842	1480.491

Rank	Strike	DTE	Final S	IV Buy → IV Sell	Price Buy	Price Sell	Return (%)
2	599	21	611.842	0.200→0.270	0.813	12.842	1480.491
3	599	21	611.842	0.200→0.230	0.813	12.842	1480.491
4	599	21	611.842	0.200→0.250	0.813	12.842	1480.491
5	599	21	611.842	0.200→0.240	0.813	12.842	1480.491
6	599	21	611.842	0.200→0.280	0.813	12.842	1480.491
7	599	21	611.842	0.200→0.210	0.813	12.842	1480.491
8	599	21	611.842	0.200→0.260	0.813	12.842	1480.491
9	599	21	611.842	0.200→0.290	0.813	12.842	1480.491
10	599	21	611.842	0.200→0.300	0.813	12.842	1480.491
11	599	21	611.842	0.200→0.200	0.813	12.842	1480.491
12	600	21	611.842	0.200→0.300	0.751	11.842	1477.017
13	600	21	611.842	0.200→0.230	0.751	11.842	1477.017
14	600	21	611.842	0.200→0.280	0.751	11.842	1477.017
15	600	21	611.842	0.200→0.240	0.751	11.842	1477.017
16	600	21	611.842	0.200→0.250	0.751	11.842	1477.017
17	600	21	611.842	0.200→0.220	0.751	11.842	1477.017
18	600	21	611.842	0.200→0.290	0.751	11.842	1477.017
19	600	21	611.842	0.200→0.200	0.751	11.842	1477.017
20	600	21	611.842	0.200→0.270	0.751	11.842	1477.017

The best-performing trades are characterized by:

- OTM, but not deep OTM, initial strikes
- Selling immediately after the anticipated event occurs (indicating short term options benefit most from IV expansion)
- Large underlying movements in increased stock price
- Small to moderate changes in IV showing that even moderate increases in implied volatility produce extreme returns for OTM calls with favorable underlying price movement
- Low initial and high final prices

Note: I'm not sure why there is this big gap under the following table, I could not get it to go away :(

Table 6: Bottom 20 Negative Return Scenarios with IV Jump

Rank	Strike	DTE	Final S	IV Buy → IV Sell	Price Buy	Price Sell	Return (%)
1	506	23	506.160	0.200→0.280	51.692	0.160	-99.690
2	506	23	506.160	0.200→0.290	51.692	0.160	-99.690
3	506	23	506.160	0.200→0.240	51.692	0.160	-99.690
4	506	23	506.160	0.200→0.260	51.692	0.160	-99.690
5	506	23	506.160	0.200→0.220	51.692	0.160	-99.690
6	506	23	506.160	0.200→0.250	51.692	0.160	-99.690
7	506	23	506.160	0.200→0.300	51.692	0.160	-99.690
8	506	23	506.160	0.200→0.210	51.692	0.160	-99.690
9	506	23	506.160	0.200→0.270	51.692	0.160	-99.690
10	506	23	506.160	0.200→0.200	51.692	0.160	-99.690
11	506	23	506.160	0.200→0.230	51.692	0.160	-99.690

Rank	Strike	DTE	Final S	IV Buy → IV Sell	Price Buy	Price Sell	Return (%)
12	506	22	506.160	0.200→0.220	51.607	0.160	-99.690
13	506	22	506.160	0.200→0.210	51.607	0.160	-99.690
14	506	22	506.160	0.200→0.240	51.607	0.160	-99.690
15	506	22	506.160	0.200→0.260	51.607	0.160	-99.690
16	506	22	506.160	0.200→0.290	51.607	0.160	-99.690
17	506	22	506.160	0.200→0.250	51.607	0.160	-99.690
18	506	22	506.160	0.200→0.300	51.607	0.160	-99.690
19	506	22	506.160	0.200→0.280	51.607	0.160	-99.690
20	506	22	506.160	0.200→0.230	51.607	0.160	-99.690

The worst performing trades are characterized by:

- ITM initial strikes
- Selling at the end of the holding period passed the event
- Large underlying price decrease
- No to small change in IV (because the worry of the stock has gone away days after the event occurred)
- High initial price and low final price

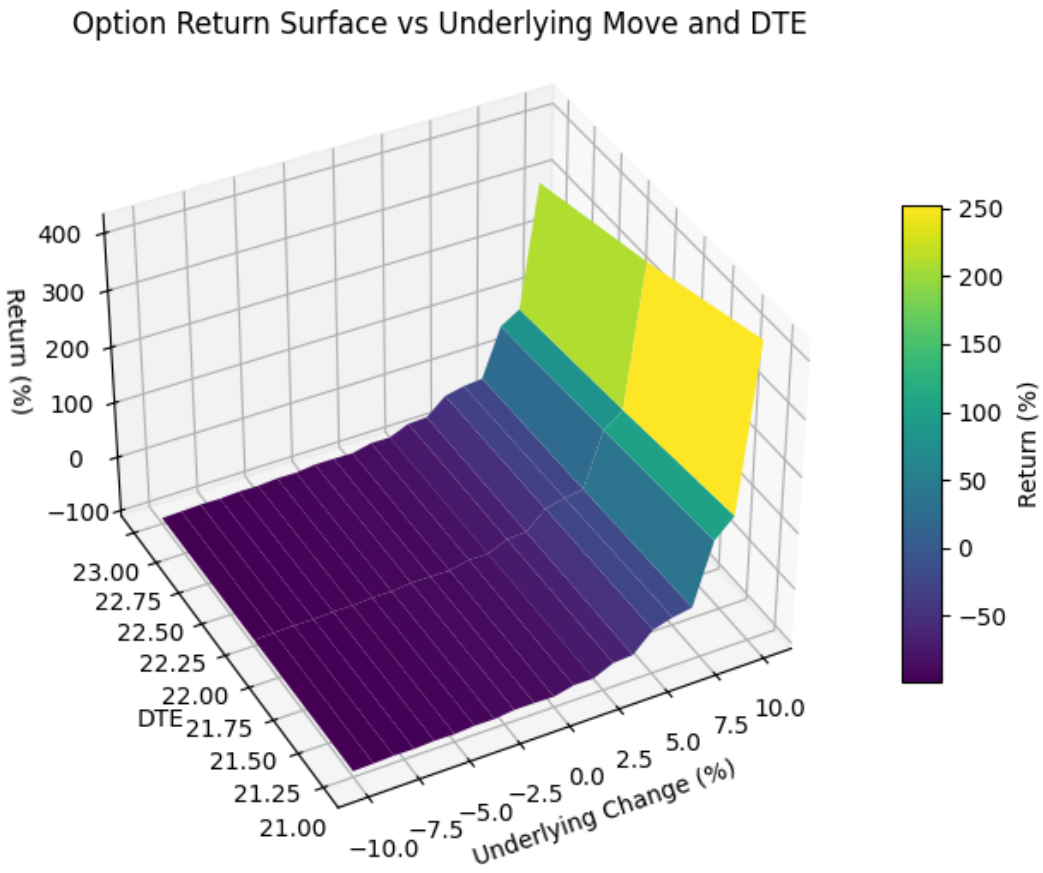

FIGURE 4.1. Option return surface as a function of underlying price movement and DTE.

The figure (Figure 4.1) provides evidence for the key findings regarding optimal call option trading characteristics. It supports the analysis that the best returns occur with high positive underlying movements and selling immediately after the anticipated event occurs (indicating short-term options benefit most from IV expansion). Conversely, the most negative returns occurred with high negative underlying movements and selling at the end of the holding period of the event.

The surface shows that the highest returns (green/yellow regions reaching 25-250%) occur in a specific "sweet spot" where underlying changes are positive (roughly 4.5-10%) and DTE is in the middle range (around 22 days). The sharp decline in returns (deep blue/purple regions showing -50% to -100% losses) when underlying prices decrease significantly validates the finding that “large underlying price decreases" characterize the worst performing trades. Additionally, the surface

shows that very short DTE options (left edge) and very long DTE options (right edge) both underperform compared to the middle range, confirming that optimal strike selection involves finding the right balance between being out-of-the-money enough to benefit from leverage while not being so far OTM or having time decay characteristics.

To view the complete results (∼40,000+ results), please see the file called **variable_iv_analysis_results.txt**

5. Option Calculator for Rolling Positions and Legging Strategies

This analysis relates to the file called **option_calculator_roll.py**

5.1. **Scenario.** This file serves to act as a calculator that answers the question, "At what future underlying price will a new option position (with shorter maturity and possibly a different strike) have the same value as my current option?" This is essentially the mechanics of an option roll (closing one option and opening another, usually further out in time or at a different strike) or trying to leg into a position by purchasing an additional position.

5.2. **Experimentation and Example.** The code is designed to be run from the command line, with the user able to input their current option's initial underlying value (s0), initial strike (k0), strike of the target (rolled) option, days to expiry of the current option (dte0), number of days rolled forward i.e., how much closer to expiry the new option is (rwsize), and current implied volatility (iv). It's considered a theoretical underlying price calculator because the model assumes a constant implied volatility across both strikes and maturities, which doesn't reflect real market conditions where IV varies (skew/term structure).

An example of calling the function would be

```
python option_calculator_roll.py --s0 100 --k0 100 --k1 110 --dte 90 --iv 0.25 --rwsize 30
```

In this scenario our initial position was bought when the underlying was at $100 and was purchased ATM with a strike of 100. There was 90 days remaining until the option expired and the IV was 25%. We want to determine what the underlying should be for a new contract with a strike at 110 and with 60 days to expiry (90 - 30) such that the value of the option is equal to or greater than the initial position.

```
Initial Value (V0): 5.4033
Required Underlying (S1): 111.1024
Final Value (V1): 5.4033

V0 at S1 (K0 strike): 13.2759
V1 at S1 (K1 strike): 5.4033
Spread (V1 - V0): -7.8726
```

Our initial position ($S_0 = 100$, $K = 100$, DTE = 90, IV = 25%) had a value (V_0) of 5.40 according to the Black-Scholes pricing model. For our new position of $K = 110$ and a DTE of 60, to be worth the same ($V_1 = 5.403$), the underlying stock would have to move to $111.10 (an 11% increase). If the stock were actually $111.10, the original contract with the 100 strike would now be worth $13.28. This shows that rolling up and shortening the time to expiration is not value-neutral: while the new 110-strike option reaches the initial value at a higher stock price, the original option has gained substantially in value. The resulting spread of -7.87 indicates that the roll would forfeit almost $7.87 in potential gains, highlighting that such a roll sacrifices intrinsic and time value in exchange for a cheaper, farther OTM position.

6. Identifying Local Extrema in Time Windows

This analysis relates to the file called **local_min_max.py**

6.1. **Scenario.** This tool is designed to identify local minima and maxima of a stock price over time. This is important for analysis as it can be helpful to detect significant local turning points, which may be used for analysis, trend detection, or mean reversion.

Local extrema are calculated using a sliding window approach, where the code checks each price point against a surrounding window of length wsize. A point is considered a local minimum if it is the lowest value within its window, and a local maximum if it is the highest value within its window. This ensures that consecutive identical extrema do not appear (in the same window), preventing the graph from marking multiple adjacent points as minima or maxima when they belong to the same trend. The analysis can be performed over multiple window sizes simultaneously, allowing the detection of short-term, medium-term, and long-term extrema.

The analysis can handle variable window sizes (wsize) to capture extrema at different time sizes. Smaller windows highlight short-term fluctuations, while larger windows capture broader trend reversals. Users can also restrict the analysis to a specific time range using the --train argument, allowing for studies of selected periods.

6.2. **Experimentation.** The function `find_local_extrema_in_windows()` begins by taking a ticker to be analyzed, and the optional parameters of --train and --wsize. The function identifies local minima and maxima in a time series of closing prices. For a given window size, the function divides the price array into consecutive segments of that length and examines each segment independently. Within each segment, the minimum and maximum prices are located, and their indices are converted to positions relative to the full dataset. To prevent multiple consecutive extrema of the same type, the function records only one extremum per window, ensuring that local minima or maxima are not duplicated in adjacent time steps. This mechanism allows for clear and meaningful identification of turning points in price trends.

The output of the function consists of two arrays: the indices of extrema and their corresponding types (either `min` or `max`). These arrays are then passed to a plotting function, which visualizes each extrema with distinct markers on the price chart.

6.3. **Analysis and Results.** In experimentation, I used the ticker symbol `MSFT`, a date range restriction using `-train 2023-01-01,now`, and multiple window sizes with `-wsize 10 20 50`. The `-train` argument slices the dataset to include only prices from January 1, 2023, to the present and the `-wsize` calculates local extrema over window lengths of 10, 20, and 50 days separately.

This is how I ran the program with the parameters explained above.

```
python local_min_max.py MSFT --train 2023-01-01,now --wsize 10 20 50
```

Below is some sample output of the program:

Table 7: Sample 7: Instance Output – 10 Day Window Extrema

Rank	Date	Price	Type
1	2022-01-06	313.88	min
2	2022-01-12	318.27	max
3	2022-01-21	296.03	min
4	2022-01-25	288.49	min
5	2022-02-02	313.46	max

Rank	Date	Price	Type
6	2022-02-09	311.21	max
7	2022-02-18	287.93	min

Table 8: Sample 7: Instance Output – 20 Day Window Extrema

Rank	Date	Price	Type
1	2022-01-21	296.03	min
2	2022-02-02	313.46	max
3	2022-02-18	287.93	min
4	2022-03-08	275.85	min
5	2022-03-18	300.43	max
6	2022-03-29	315.41	max
7	2022-04-14	279.83	min

Table 9: Sample: 7 Instance Output – 50 Day Window Extrema

Rank	Date	Price	Type
1	2022-01-25	288.49	min
2	2022-03-08	275.85	min
3	2022-03-29	315.41	max
4	2022-05-04	289.98	max
5	2022-06-13	242.26	min
6	2022-07-26	251.90	min
7	2022-08-15	293.47	max

To see the full output, please see the file called **local_min_max.txt**
Below are the labeled graphs with local extrema indicated in red and green colors.

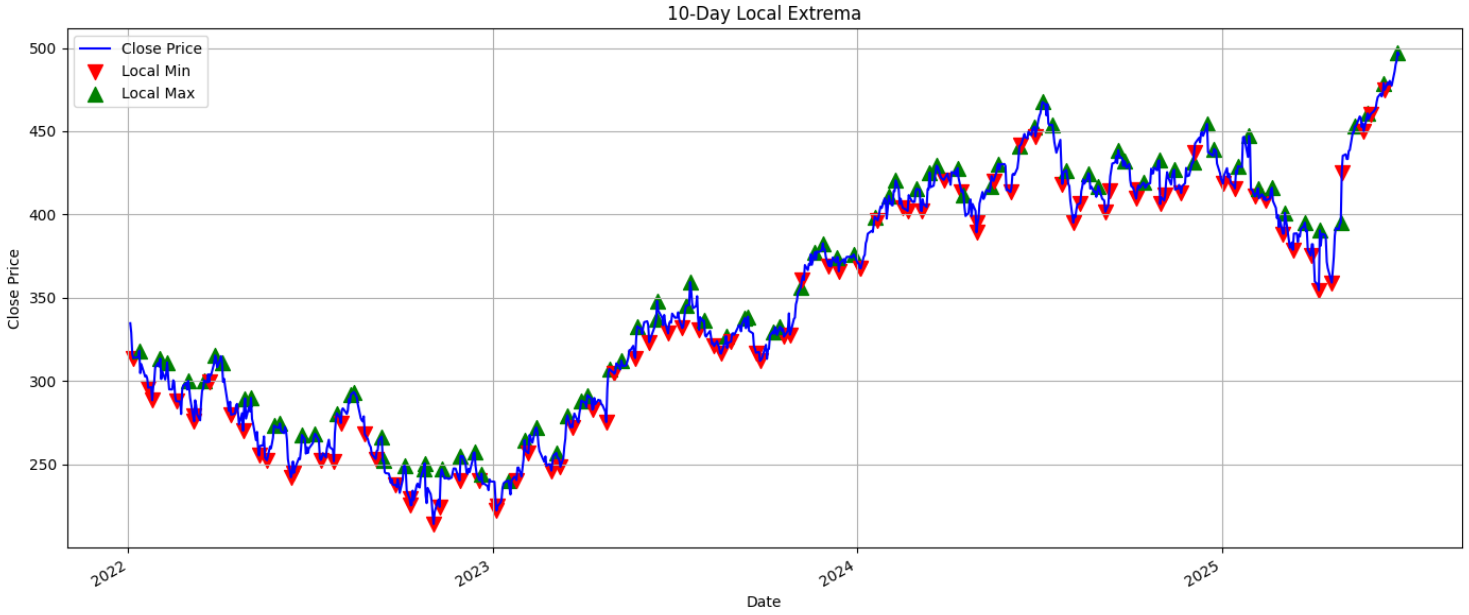

FIGURE 6.1. MSFT 10 Day Window Extrema

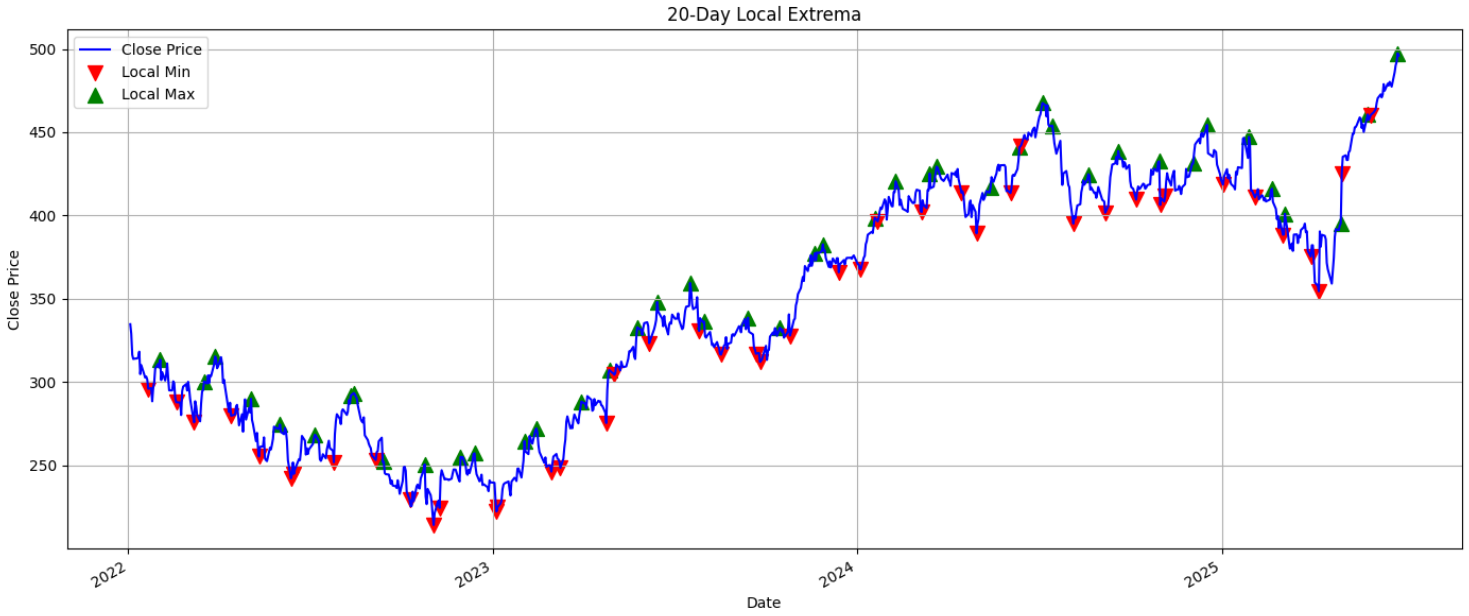

FIGURE 6.2. MSFT 20 Day Window Extrema

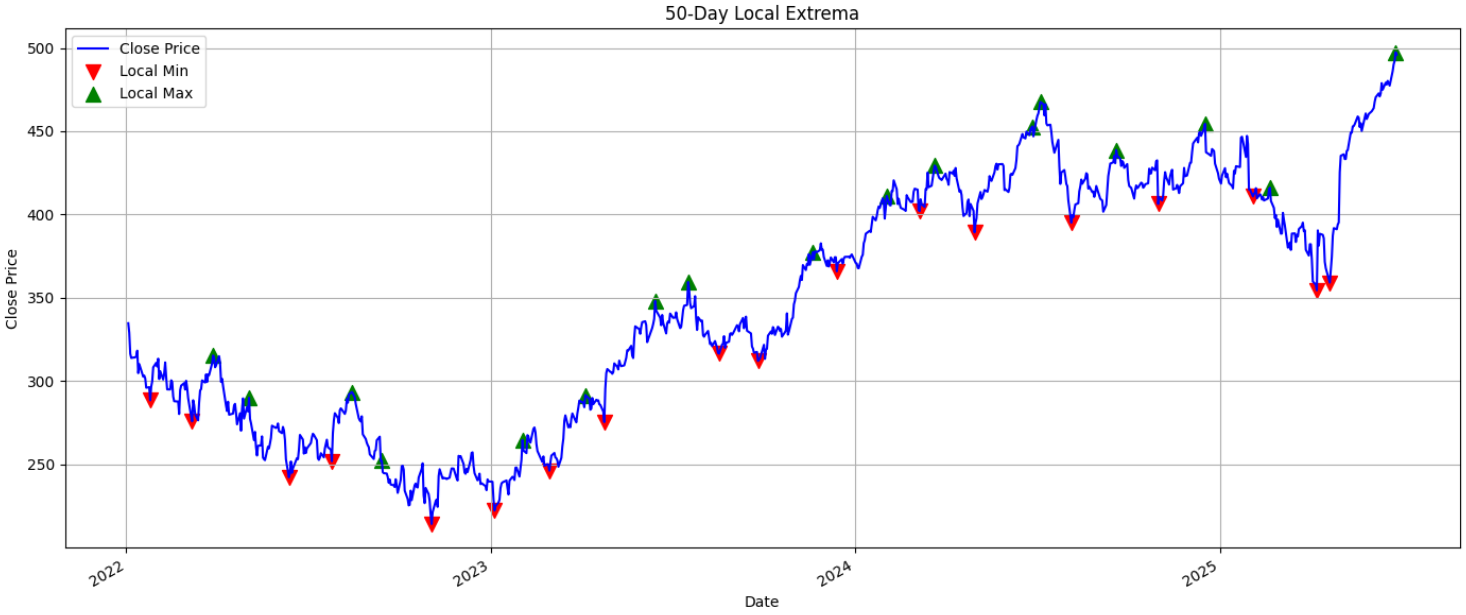

FIGURE 6.3. MSFT 50 Day Window Extrema

We can use these points for various analysis such as trend detection or mean reversion.

7. Rolling Beta and Momentum Analysis Using Least Squares Regression and the Capital Asset Pricing Model

This analysis relates to the file called **least_square_fitting.py**

7.1. **Scenario.** This tool performs rolling linear regression analysis to calculate beta coefficients and momentum factors for individual stocks relative to a benchmark ticker. The analysis implements a two-stage regression approach: first calculating Capital Asset Pricing Model (CAPM) beta coefficients through rolling windows, then extending the analysis to capture momentum effects by examining how stock residuals respond to lagged market returns.

The primary objective is to quantify how individual stock sensitivities to market movements change over time, and to identify momentum patterns that may persist beyond the standard market beta relationship. This analysis is particularly valuable for portfolio construction, risk management, and identifying stocks that exhibit momentum or mean-reversion characteristics relative to the benchmark.

The code uses SPY (SPDR S&P 500 ETF) as the market benchmark (however it can be changed) and performs rolling 25-day window regressions to capture the relationship between individual securities and the market. The momentum analysis extends beta calculations by examining whether stocks that outperform or under perform the market in one period tend to continue that pattern in subsequent periods.

7.2. **Experimentation.** For this experimentation, SPY was used as the market benchmark and three major stocks where picked (Apple, Microsoft, Tesla). Below is how to run the program:

```
python least_square_fitting.py SPY AAPL MSFT TSLA
```

The function `linear_fit()` performs the initial part of the program by computing rolling linear regressions between individual stock log returns and market log returns. For each 25-day window, the function fits the model:

$$r_{i,t} = \alpha_i + \beta_i \cdot r_{m,t} + \varepsilon_{i,t}$$

where $r_{i,t}$represents the log return of stock i at time t, $r_{m,t}$ represents the market return, α_i is the stock's alpha (excess return), and β_i is the stock's beta (market sensitivity).

The function `momentum_way()`implements the second part of the analysis by analyzing residuals from the initial regression. It fits the model:

$$\varepsilon_{i,t} = \gamma_i \cdot r_{m,t-1} + \eta_{i,t}$$

where $\varepsilon_{i,t}$ are the residuals from the first regression and γ_i represents the momentum coefficient, capturing how current period residuals respond to previous period market returns.

The analysis processes multiple stocks simultaneously, creating time series of rolling alpha, beta, and momentum coefficients for each ticker input. This approach allows for identification of periods when individual stocks become more or less sensitive to market movements, and where momentum or mean-reversion patterns could exist.

The analysis produces three output arrays:

- Alphas array: Shape (N_stocks, N_windows) containing rolling alpha coefficients
- Betas array: Shape (N_stocks, N_windows) containing rolling beta coefficients
- Momentum array: Shape (N_stocks, N_windows) containing momentum coefficients

These arrays will be used for visuals and analysis.

7.3. **Analysis.** The rolling regression analysis of AAPL, MSFT, and TSLA against SPY over the period 2022-2025 reveals patterns in market sensitivity and momentum characteristics for each stock.

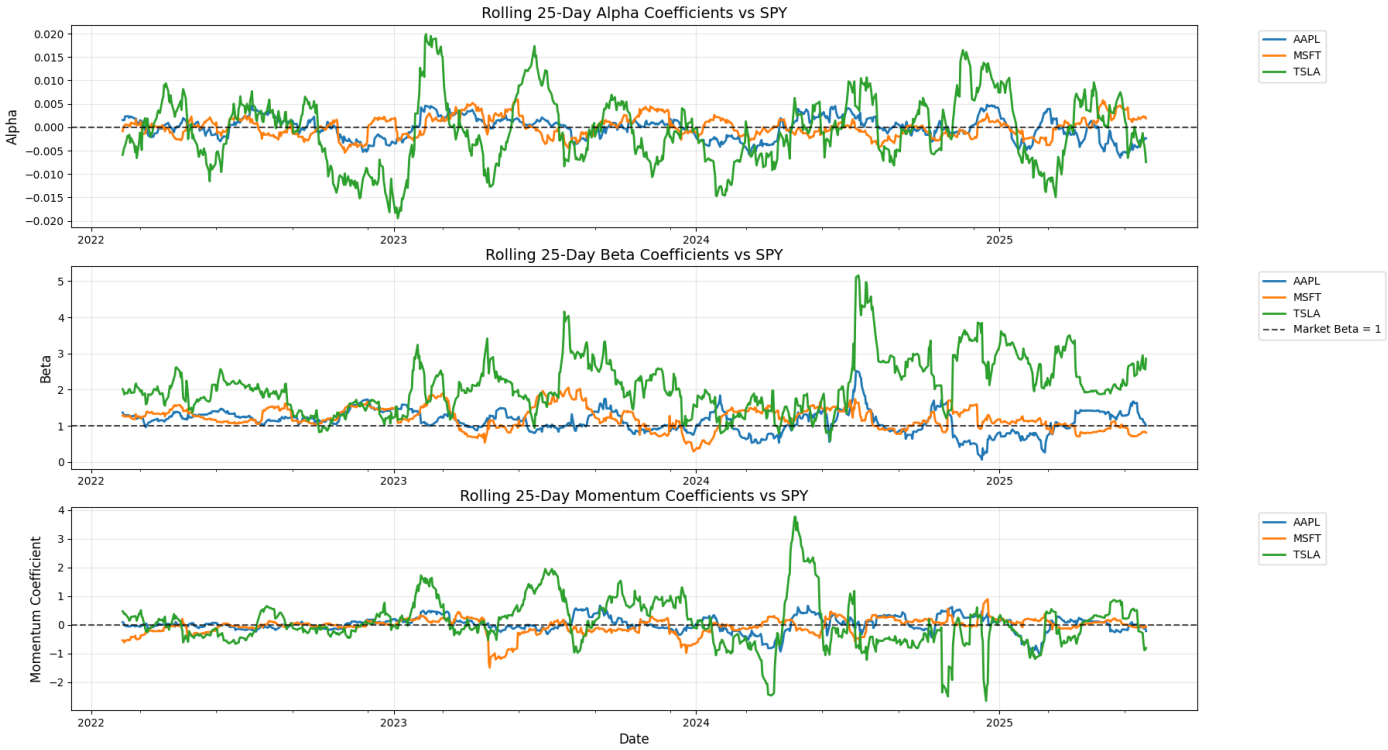

FIGURE 7.1. 25 Day Rolling Window Alphas, Betas, Momentums

In Figure 7.1, we see 25 day rolling window alphas, betas, momentums for the 3 tickers.

- Alpha Coefficients (Top Panel)
  - As a reminder, alpha refers to a stocks excess return when fitting with the equation. The rolling 25 day alpha coefficients reveal that all three stocks maintain excess returns close to zero relative to SPY, which aligns with efficient market expectations. TSLA (green line) exhibits the most volatile alpha pattern, with dramatic swings between approximately +0.02 and -0.02, particularly notable during 2023 and onwards. This is somewhat expected as Tesla tends to be one of the most volatile stocks and movements can change heavily based on what is said on social media. Additionally, these large alpha deviations suggest periods where TSLA significantly outperformed or underperformed its CAPM predicted returns. AAPL (blue line) and MSFT (orange line) show much more stable alpha patterns, rarely deviating beyond ± 0.005, indicating these more mature technology companies maintain more predictable relationships with the market and outperforming it from time to time.
- Beta Coefficients (Middle Panel)

  - As a reminder, beta refers to the stock's market sensitivity. The beta analysis shows major differences in market sensitivity across the three stocks. TSLA demonstrates extreme beta volatility, with values ranging from below 1.0 to peaks exceeding 5.0, particularly during 2023 and mid to late 2024. This indicates periods where TSLA was dramatically more sensitive to market movements than the average stock. This is some what expected as Tesla is one of the most volatile stocks and movements can change heavily based on what is said on social media. AAPL maintains a relatively stable beta around 1.2-1.5, with some fluctuation but generally predictable market sensitivity. MSFT exhibits the most stable beta profile, consistently hovering around 1.0 throughout the entire period, suggesting it moves nearly in perfect correlation with the broader market. Values larger than 1.0 can indicate unpredictability but this also means returns can be much higher or much lower than the broader market.
- Momentum Coefficients (Bottom Panel)
  - As a reminder, momentum captures how current period residuals respond to previous period market returns. Additionally, the momentum analysis reveals how residual returns respond to lagged market movements. TSLA shows the most extreme momentum effects, with coefficients ranging from -2.5 to +4.0, indicating strong persistence in residual returns. The large positive momentum spike around mid-2024 suggests a period where TSLA's deviations from expected returns showed strong positive correlation. AAPL and MSFT display more moderate and stable momentum patterns, generally oscillating between -1.0 and +1.0, with frequent zero crossings indicating less persistent momentum effects.

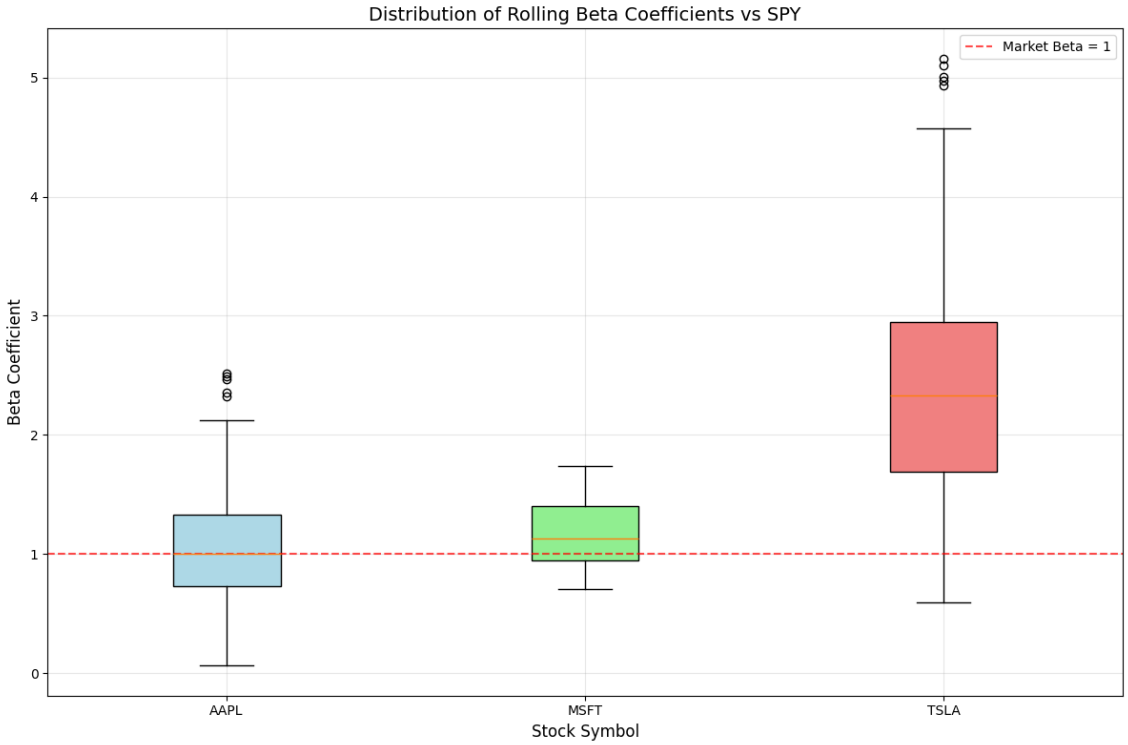

FIGURE 7.2. Beta Distribution Box Plots

Figure 7.2 provides summary statistics for the rolling beta coefficients across the entire time period. TSLA shows the highest median beta (approximately 2.0) with an extremely wide distribution, as seen by the large interquartile range extending from roughly 1.7 to 3.0, and numerous outliers reaching up to 8.0. This confirms TSLA's highly variable and elevated market sensitivity as mentioned in the previous part.

AAPL displays a more concentrated distribution with a median beta around 1.0-1.1, showing some outliers but generally stable behavior. The relatively tight interquartile range suggests consistent market sensitivity over time.

MSFT exhibits the most stable beta distribution, with a median very close to 1.0 and the tightest interquartile range of all three stocks. The few outliers present are modest compared to the other stocks.

The red dashed line at $\beta = 1.0$ (market beta) shows that MSFT trades closest to market sensitivity, AAPL slightly above market sensitivity, and TSLA significantly above market sensitivity with high variability.

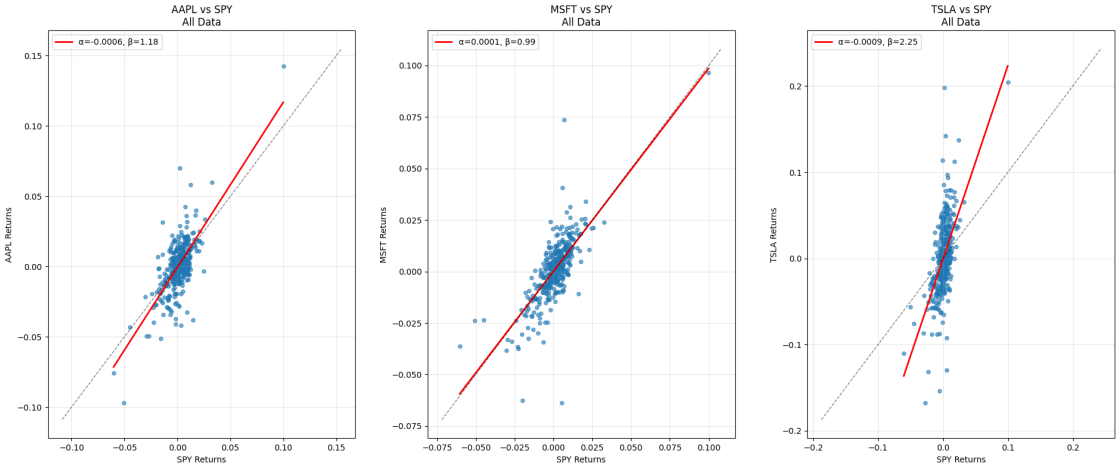

FIGURE 7.3. Relationship Between SPY Returns and Individual Returns

- AAPL vs SPY (Left Panel)
  - The scatter plot shows a clear positive linear relationship with α = -0.0006 and β = 1.18. The data points cluster relatively tightly around the regression line, indicating a stable relationship. The slight negative alpha suggests AAPL slightly under performed its CAMP-predicted returns on average. However the higher than market beta suggests AAPL can yield larger (more positive and negative) returns.
- MSFT vs SPY (Middle Panel)
  - MSFT displays the strongest linear relationship with SPY, as seen by the tight clustering of points around the regression line. With α = 0.0001 and β = 0.99, MSFT essentially mirrors market movements with minimal excess return. The regression line nearly overlaps with the theoretical market line (dashed).
- TSLA vs SPY (Right Panel)
  - TSLA shows the highest beta (β = 2.25) and positive alpha (α = 0.0009), indicating both higher market sensitivity and slight out performance. However, the scatter shows much wider dispersion around the regression line, reflecting TSLA's higher volatility. Several extreme outliers are visible, particularly in positive return scenarios, which is consistent with TSLA's reputation for dramatic price movements.

This analysis is useful to understand how to deal with stable, moderate, and highly volatile relationships with the broader market which can provide insights for portfolio construction and risk management strategies.

To see the full results with all beta, alpha, and momentum values, please see the file called: **rolling_regression_results.txt**

8. Stock Correlation Analysis

This analysis relates to the file called **correlation_and_de.py**

8.1. **Scenario.** There are two components to the code file. This section will discuss the first component, which uses correlation.

This code analyzes stock correlations by computing Pearson correlation coefficients between a reference stock and a broader universe of stocks using their log returns. The analysis focuses on identifying stocks that exhibit strong positive correlation, strong negative correlation, or neutral correlation relative to the chosen benchmark stock. Log returns are calculated as the difference between consecutive logarithmic prices, providing a measure of percentage price changes that are approximately normally distributed and additive over time.

The correlation analysis can help identify stocks that move similarly to a benchmark for portfolio diversification (uncorrelated stocks, in theory, would have opposite price movements) or concentration strategies (correlated stocks, in theory, move together), as well as find potential hedging candidates through negatively correlated assets and provide insight into market sector relationships. The methodology implements multiple correlation calculation approaches for validation, including manual computation of Pearson coefficients and NumPy's built-in correlation function; however, for the analysis, only NumPy's implementation will be used.

8.2. **Experimentation.** The analysis is designed to be run from the command line with parameters including an input dataset (-i), a reference stock for correlation analysis (--r), a correlation threshold (--t) for classification, and a reference date (–train) indicating when the data analysis should begin. For example, running `python correlation_and_de.py -i sp500 -r AVGO -t 0.7 -train 2023-01-01,now`would analyze correlations within the S&P 500 using AVGO as the reference stock with a correlation threshold of 0.7 from 2023 to the present.

The function begins by loading aligned stock price series and converting them to log returns using the `log_returns()` function, which computes `log(price[t]) - log(price[t-1])` for each consecutive time period. The correlation computation is done through the `using_corrcoef()` the function that wraps numpy's `corrcoef()` to generate the full correlation matrix between all stock's log returns.

For pairwise analysis with the reference stock, the `extract_correlation_pairs2()` function extracts the reference stock's correlation row from the full matrix and categorizes each relationship based on the threshold parameter `t`. Stocks with correlations above `t` are classified as highly correlated, those below `-t` as negatively correlated, and those between `-t` and `t` as having moderate correlation. This classification system allows for systematic identification of stocks with different relationship patterns to the benchmark.

8.3. **Analysis.** The correlation analysis using AVGO (Broadcom) as the reference stock reveals patterns within the S&P 500 over the period from January 2023 to June 2025. Using a correlation threshold of 0.7, the analysis identified 8 stocks with high positive correlation to AVGO and notably zero stocks with strong negative correlation (below -0.7), indicating that AVGO's price movements align directly with the broader market during this period.

The highly correlated stocks where from a diverse set of industries, suggesting that AVGO's correlation patterns reflect broader market dynamics rather than sector-specific movements. The strongest correlations include EQT (0.930), UHS (0.892), ETN (0.887), and CRL (0.843), which are in the energy, healthcare, industrial, and research sectors respectively. This diversification across sectors while maintaining high correlation suggests these stocks may be responding to similar

macroeconomic factors such as interest rate changes, growth expectations, or market sentiment rather than industry-specific drivers.

The absence of strongly negatively correlated stocks (none below -0.7) is particularly noteworthy, as it indicates that during this 2.5-year period, no major S&P 500 members exhibited consistent reverse behavior relative to AVGO. This finding suggests either a period of broad market synchronization or that AVGO's semiconductor exposure made it representative of general technology and growth stock movements.

The majority of stocks (454 out of 462, or 98%) fell into the moderate correlation category (-0.7 to 0.7), with correlations ranging from very weak positive relationships to moderately strong positive relationships. Notable observations include traditionally defensive stocks like utilities and consumer staples showing moderate positive correlations rather than negative correlations, suggesting that during this period, sector rotation was less pronounced and growth/technology factors dominated market movements across sectors.

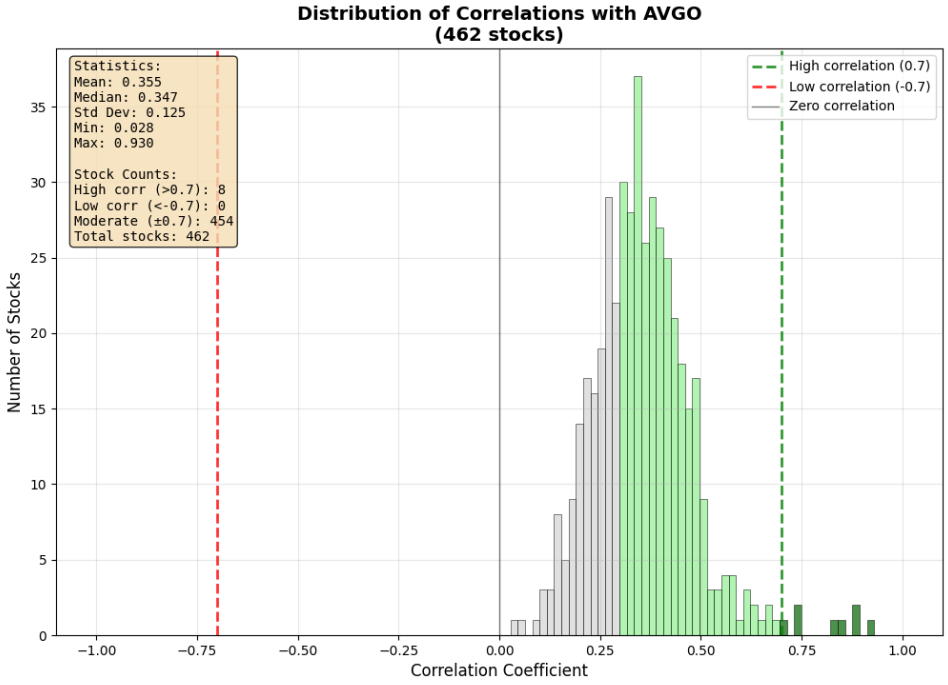

FIGURE 8.1. Distribution of Correlations with AVGO

The correlation distribution histogram (Figure 8.1) reveals several insights about AVGO's relationship with the broader S&P 500 market structure. The distribution exhibits a slight right skew with mean correlation of 0.355 exceeding the median of 0.347, indicating that AVGO tends to move positively with most market constituents rather than providing diversification benefits. The standard deviation of 0.125 suggests relatively homogeneous correlation patterns across the index,

while the absence of stocks with correlation below -0.7 could limit hedge opportunities within large-cap US equities. This narrow correlation range from 0.028 to 0.930, combined with 98% of stocks (454/462) falling in the moderate correlation category, indicates that AVGO exhibits systematic risk characteristics rather than idiosyncratic patterns, potentially requiring diversification or derivative strategies for effective risk management.

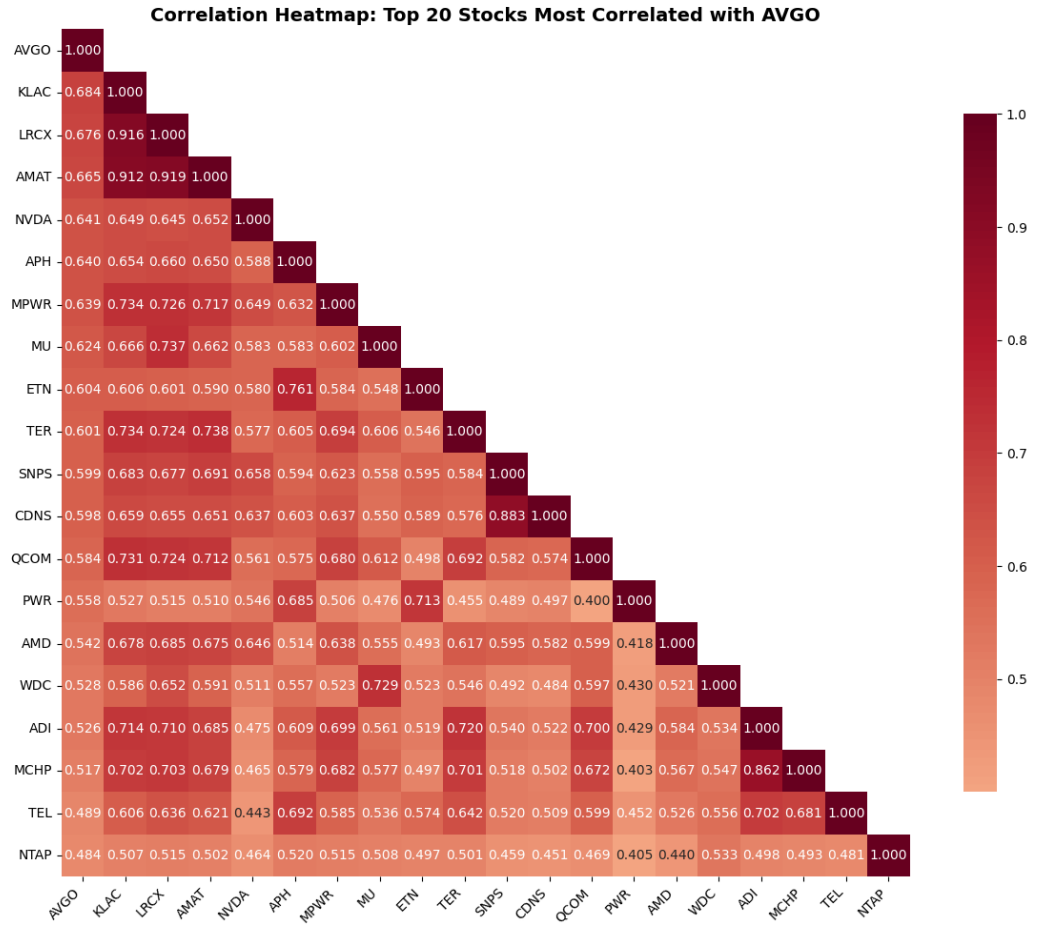

FIGURE 8.2. Correlation Heatmap with AVGO

The correlation heat map (Figure 8.2) provides more insights into the relationships among the top 20 stocks most correlated with AVGO, revealing clustering patterns that could have significant portfolio construction implications. The semiconductor equipment cluster consisting of KLAC (0.684), LRCX (0.676), and AMAT (0.665) demonstrates mutual correlations, confirming these stocks trade as a cohesive sector and validating AVGO's supply chain inter dependencies within the semiconductor ecosystem. More intriguingly, the cross-sector correlation patterns show industrial stocks like ETN (0.604) and TER (0.601) exhibiting significant correlations despite operating

outside core technology, suggesting that growth overwhelms traditional sector based differentiation during this analysis period. These high mutual correlations among the selected stocks create significant concentration risk for portfolio construction, as selecting multiple stocks from this matrix would not provide effective diversification. The correlation structure implies these 20 stocks share common exposure to growth factors, interest rate sensitivity, and momentum effects, with the absence of negative correlations and tight positive clustering indicating the 2023-2025 analysis period represents a unified market characterized by technology/AI investment themes, low volatility environment, and growth across sector boundaries, making this correlation matrix valuable for constructing portfolios, testing factor model assumptions, and calibrating risk models in portfolio management systems.

To view all correlation values, please see the file called: **correlation.txt**

9. Differential Evolution for Selecting Related Stocks

This analysis relates to the file called **correlation_and_de.py**

9.1. **Scenario.** There are two components to the code file. This section will discuss the second component, which uses Differential Evolution (DE) which is an optimization algorithm.

In this analysis, DE is used to select a group of stocks that are most strongly correlated with each other. Given a correlation matrix of all stocks, the algorithm searches for exactly `k` stocks whose pairwise correlations maximize the overall similarity within the group. Each candidate solution is treated as a binary vector (values above 0.5 indicate selection), and the objective function evaluates how well the chosen group maximizes total correlation.

In the code, DE evaluates each candidate group by summing the correlations between all selected stocks. If the group does not contain exactly `k` stocks, a penalty is applied. The algorithm generates new candidate groups by combining and mutating existing ones, keeping the groups that produce higher total correlations. Over many iterations, DE converges on a set of \textttk stocks that are most mutually correlated.

9.2. **Experimentation.** The DE algorithm is implemented in the `select_topk_de()` function, which works with the following parameters:

- **C**: Correlation matrix ($n \times n$) containing pairwise correlations between all stocks
- **symbols_v**: Vector of stock symbols corresponding to matrix rows/columns
- **k**: Target number of stocks to select ($k = 15$ in this experiment)
- **seed**: Random seed

The algorithm works as follows:

- **Initialization**: Creates a random population of 15 candidate solutions (vectors of length 462)
- **Evolution Loop**: For each generation:
  - **Mutation**: Creates mutant vectors by combining existing solutions
  - **Crossover**: Mixes mutant with target vectors based on recombination rate
  - **Selection**: Keeps better solutions based on objective function value
- **Convergence**: Continues until improvement stops or maximum iterations are reached
- **Post-processing**: Converts the final continuous solution to a binary selection via thresholding

As mentioned in Section 8.1, there is an objective function to evaluate how well the chosen group of `k` stocks maximizes total correlation. The DE algorithm maximizes the objective function

$$f(\mathbf{x}) = \mathbf{x}_{\text{bin}}^T \mathbf{C} \mathbf{x}_{\text{bin}}$$

subject to

$$\sum_{i=1}^{n} x_{\text{bin},i} = k,$$

where $\mathbf{x}_{\text{bin}}$ is the binary selection vector, $\mathbf{C}$ is the correlation matrix, and $k = 15$ is the target number of stocks.

Additionally, the DE algorithm was executed with the following parameters:

- Maximum iterations: 1000 (to shorten computation time)
- Mutation range: 0.5 to 1 (controls the step size for creating new candidate solutions)

- Recombination rate: 0.7 (probability of mixing components between mutant and target vectors during crossover)
- Strategy: `best1bin` (uses the best individual as a base for creating new candidates)

9.3. **Results and Analysis.** The DE algorithm identified the following 15 stocks as the most mutually correlated group:

DE Objective Value: 120.7126	
Index	**Symbol**
39	APTV
47	AXP
67	BX
141	EMN
207	HST
213	IEX
239	KIM
293	MTB
338	PLD
358	RF
397	TEL
419	UDR
423	UNP
438	WAT
452	WY

The Differential Evolution algorithm successfully converged, identifying a group of 15 stocks that maximizes the sum of their pairwise correlations. The final objective value for this selected group was 120.7126. This value represents the sum of all elements in the 15×15 correlation sub matrix corresponding to these stocks. To better understand the strength of this relationship, we can calculate the average pairwise correlation within the group. The sum of the diagonal elements of this sub matrix is always k=15 (since each stock's correlation with itself is 1). The remaining value, $120.7126 - 15 = 105.7126$, is the sum of the $15 \times 14 = 210$ off-diagonal elements. Therefore, the average pairwise correlation between any two stocks in this group is approximately $105.7126/210 \approx 0.503$. This indicates a consistently strong positive relationship across the entire selected portfolio, confirming the algorithm's effectiveness in identifying a cohesive cluster of assets.

A qualitative review of the selected stocks reveals a clear thematic connection, which validates the quantitative findings. The list is heavily weighted towards the Financials, Real Estate (specifically REITs), and Industrial sectors. For example, AXP (American Express), BX (Blackstone), MTB (M&T Bank), and RF (Regions Financial) represent financial services, while HST (Host Hotels & Resorts), KIM (Kimco Realty), PLD (Prologis), UDR (UDR, Inc.), and WY (Weyerhaeuser) are all REITs or in the real estate sector. This clustering is financially intuitive, as these sectors are often highly sensitive to the same macroeconomic factors, such as interest rates, credit cycles, job numbers and overall economic growth. From a practical standpoint, this analysis has direct applications in portfolio management. For risk management, this group represents a source of concentrated systematic risk; a portfolio heavily invested in these assets would lack diversification. Conversely, for a trading strategy, this highly correlated basket could be a prime candidate for

arbitrage or pairs trading, where one would look for temporary price divergences from the group's aggregate behavior.

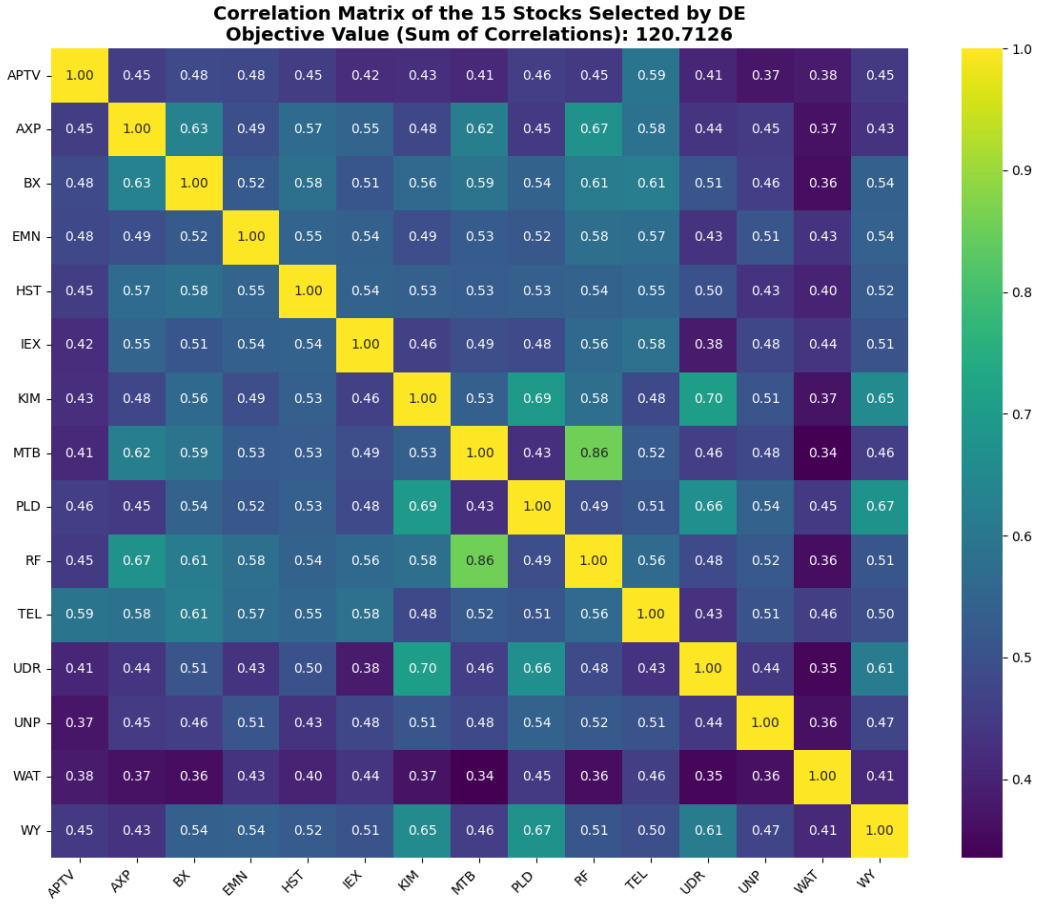

FIGURE 9.1. Correlation heatmap of the DE-optimized stock portfolio

To visually validate the output of the Differential Evolution algorithm, the correlation matrix of the 15 selected stocks is presented in the heat map above (Figure 9.1). This visualization serves to move beyond the single aggregate objective score (120.7126) and to provide a view of the individual pairwise relationships that the algorithm optimized for. The objective function, $f(\mathbf{x}) = \mathbf{x}_{\text{bin}}^T \mathbf{C} \mathbf{x}_{\text{bin}}$, is the sum of every value in this matrix. The uniformly warm color pattern and the fact that all correlation values are greater than zero together confirm the group's cohesiveness.

This visual evidence directly supports the quantitative analysis. The average pairwise correlation of approximately 0.503 is reflected in the consistent blue-green shading across the off-diagonal elements. Furthermore, the heat map reinforces the analysis regarding sector-based clustering. For example, we can observe particularly strong relationships (brighter blue-green cells) between stocks within the financial and real estate sectors, such as the 0.86 correlation between MTB and

RF. This experiment demonstrates that the Differential Evolution algorithm did not merely find a mathematically optimal solution but identified a genuinely and consistently inter-correlated group of assets, which is an important task in quantitative strategies like risk factor modeling.

To view the DE output, please see the file called: **correlation.txt**

10. The Relationship Between Volatility (Risk) and Returns (Reward) in Financial Markets

This analysis relates to the file called **risk_vs_reward.py**

10.1. **Scenario.** This analysis examines the relationship between market volatility (risk) and returns (reward) in financial assets. A sliding window methodology is applied to test whether stocks with higher volatility deliver proportionally higher returns—a common assumption when creating portfolios.

The core hypothesis under investigation is that the risk return trade off holds in practice: assets exhibiting greater historical volatility should compensate investors with higher future returns. However, markets are not always perfectly efficient. Factors such as changing market conditions, liquidity constraints, geopolitics, news, and investor behavior can distort or break this relationship.

Understanding when and where the risk return trade off holds or breaks down is critical for: portfolio optimization, asset pricing models, which rely on risk premiums for valuation, and investment strategy.

10.2. **Experimentation.** The analysis comes from the function `calculate_risk_reward()`. In this experimentation, the analysis implements a sliding window approach to capture the evolving risk vs. reward relationship over time.

Risk is determined as the standard deviation of logarithmic returns over historical window of `swsize` days, where `swsize` represents the sliding window size for standard deviation calculation (here I used 50 days of historical price data). Standard deviation of logarithmic returns serves as an excellent risk measure because it captures price volatility in a statistically robust manner as logarithmic returns are approximately normally distributed, making standard deviation a meaningful measure of dispersion. Additionally, the log transformation ensures that returns are additive over time and symmetric around zero. Lastly, standard deviation quantifies the uncertainty or unpredictability of returns, which directly corresponds to investment risk from both upside and downside perspectives. The volatility measure is then scaled by $\sqrt{\texttt{rwsize}}$, where `rwsize` is the the return window size for future performance measurement. This is done to ensure statistical comparability across different time windows.

Reward is determined by the percent return over a forward looking window of `rwsize` days, where `rwsize` represents the return window size extending into the future from each calculation point, calculated as $\frac{\text{Price}_{t+\text{rwsize}} - \text{Price}_t}{\text{Price}_t}$. Percentage returns provide the most intuitive and practical measure of reward because they directly represent the actual percentage gain or loss that an investor would experience, making them easy to interpret for portfolio performance evaluation (such as a 100% gain doubles an investment while a -50% loss halves it). The forward looking nature of this measurement tests whether historical volatility contains predictive information about future price movements over the specified `rwsize` period, establishing a relationship between past risk and future reward.

The analysis implements a sliding window approach controlled by the `xwsize` parameter, where `xwsize` represents the extended window size that determines the number of calculation points extending forward from a reference date. The reference date serves as the temporal anchor point for the entire analysis, representing a specific moment in time from which all risk and reward calculations are performed. The sliding window enables the analysis to generates multiple risk vs. reward observations per stock by shifting the calculation window over time from this reference date, significantly increasing the number of scenarios analyzed.

The mentioned parameters, `rwsize, swize, refdate, and ticker` can be called from the command line. Two examples of how to run the program, which this analysis will be based on are the following:

```
1) python risk_vs_reward.py -i sp500 --refdate 2023-04-03 --swsize 50 --rwsize 5 --xwsize 1 --show
2) python risk_vs_reward.py AAPL --refdate 2025-04-01 --swsize 30 --rwsize 5 --xwsize 10 --show
```

Therefore in this analysis, in scenario (1), the SP500 will be analyzed, the reference date of April 3rd, 2023, `swsize` of 50 days, `rwsize` of 5 days, and `xwsize` of 1 day will be used. `-show` is an optional command line flag to choose whether to display the visuals or not. In scenario (2), Apple will be analyzed, with a reference date of April 1st, 2025, `swsize` of 30 days, `rwsize` of 5 days, and `xwsize` of 1 day will be used.

10.3. **Analysis.** Both figures have reference lines of $y = \pm x, y = \pm 2x$, and $y = \pm 3x$ to show theoretical risk-reward returns (100%, 200%, 300% returns).

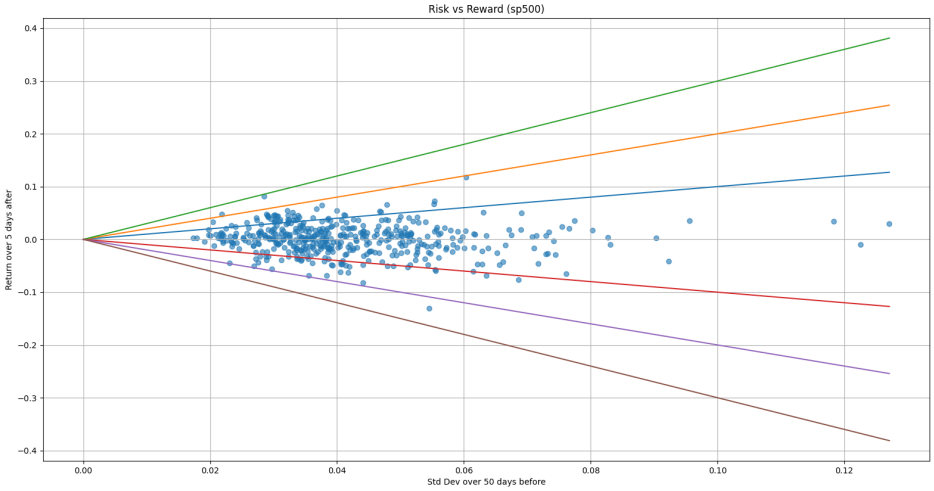

FIGURE 10.1. Risk vs Reward Analysis for SP500 stocks (April 3, 2023)

The S&P 500 analysis, conducted with a single observation point (`xwsize=1`) on April 3rd, 2023, using 50 days of historical volatility to predict 5 day forward returns, shows a clustering of observations between $y = \pm x$ returns regardless of risk level. However, there are more points above $y = 0$, suggesting the S&P 500, in general, rewards those who take the risk by investing there.

The reference lines ($y = x, y = 2x, y = 3x$) in the S&P 500 plot serve as benchmarks for risk compensation, yet not many observations fall above even the most conservative 1:1 risk-reward line. The few positive outliers achieving returns above the line appear randomly distributed across the volatility spectrum, indicating that short-term out performance was driven by news factors or random market movements rather than risk characteristics. The concentration of observations near

the zero-return horizontal line, combined with the wide dispersion of volatility measures, provides evidence that 5 day price movements followed a random walk pattern during this time period.

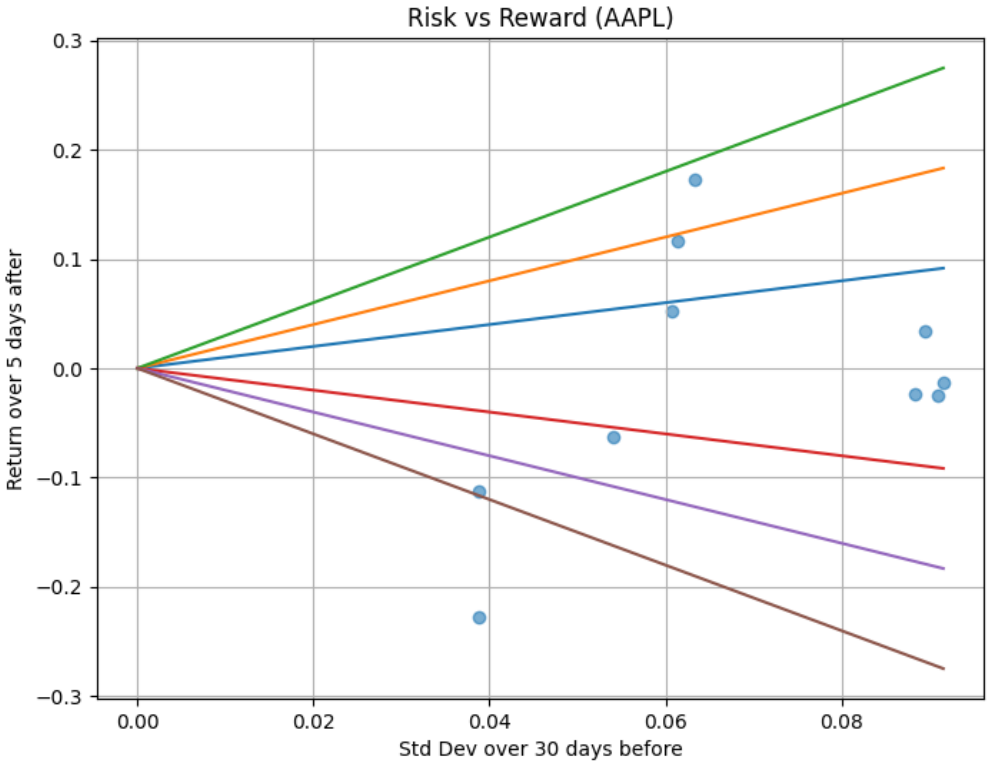

FIGURE 10.2. Risk vs Reward Analysis for SP500 stocks (April 1, 2025)

AAPL's individual stock analysis, conducted with more recent data (April 1st, 2025 reference date) using 30 days of historical volatility across 10 sliding window observations (xwsize=10), reveals a different but equally challenging risk-reward profile. The 10 observations (which are notably few to draw conclusions) span a volatility range from approximately 0.03 to 0.09, with returns varying dramatically from -0.25 to +0.17, demonstrating the higher variability inherent in single stock analysis compared to diversified portfolio effects. However, it must also be noted that the date was intentionally picked here for AAPL, as that is when there was severe volatility in the market due to tariff announcements. Apple was chosen for this reason as much of their manufacturing is done in affected countries like China and India. Critically, several observations fall substantially below the negative reference lines, indicating periods where AAPL experienced significant losses despite only moderate volatility levels (0.04-0.06), suggesting that individual stock risk encompasses factors beyond price volatility.

The structure in AAPL's 10 observation sliding window analysis provides insights into the instability of risk-reward relationships over time. The presence of both extreme positive and negative outcomes at similar volatility levels indicates that the risk-reward relationship for individual securities is highly time dependent and subject to company-specific events, market changes, or macroeconomic shifts that are not captured by historical volatility alone. Unlike the S&P 500's consistent horizontal clustering, AAPL's observations show greater dispersion both vertically and horizontally, suggesting that individual stock analysis captures risks and opportunities that portfolio diversification typically eliminates. The few observations approaching or exceeding positive reference lines occur at moderate volatility levels (0.06-0.08), indicating that AAPL's risk-reward profile may be more favorable during specific market conditions, but this relationship is neither consistent nor predictable based solely on historical volatility measures.

These findings demonstrate that the classical risk-return paradigm fails to hold empirically over short time horizons (5 days) in both diversified and concentrated positions. The S&P 500 results suggest that modern financial markets have achieved sufficient efficiency to eliminate systematic risk based arbitrage opportunities, while the AAPL analysis reveals that individual stock selection has additional layers of complexity and unpredictability that make risk-reward relationships even more difficult. These are implications for portfolio management as investors cannot reliably expect higher returns from higher volatility assets over short-term periods, and risk management strategies based purely on historical volatility measures may provide inadequate protection against outcomes.

11. Analyzing Seasonal Stock Returns Using Kernel Density Estimation

This analysis relates to the following file: **density_of_returns.py**

11.1. **Scenario.** The purpose of this analysis is to examine if seasonal patterns exist in stock market returns by reorganizing historical price data according to calendar days rather than chronological time. This approach reveals whether certain calendar dates or periods during the year consistently exhibit higher or lower returns across multiple years, potentially indicating market seasonality effects. Many institutional investors and quantitative funds analyze such calendar based patterns to identify recurring market behaviors for risk management purposes.

The analysis uses Kernel Density Estimation (KDE) to compute smoothed return distributions from historical price data. Rather than using raw daily returns, which can be noisy and obscure patterns, KDE applies Gaussian kernels to create continuous probability density functions of returns. This approach transforms discrete return observations into smooth density curves that better reveal the underlying characteristics and seasonal patterns.

The return computation is calculated as $\frac{price_t}{price_{t-wsize}} - 1$, where `wsize` represents a look back window. These returns are then processed through a Gaussian convolution operation that weights nearby observations according to a normal distribution kernel. The resulting density estimates provide a cleaner signal for identifying consistent seasonal patterns while filtering out random market noise.

11.2. **Experimentation.** The experimentation process transforms raw stock price data into smooth seasonal return distributions. Each stage is implemented in the functions:

- Return Computation — `compute_returns()` Computes rolling returns using

$$r_t = \frac{P_t}{P_{t-wsize}} - 1$$

  where `wsize` controls the lookback window.
- Gaussian Kernel Construction — `gaussian_kernel()` Builds a normalized Gaussian weighting function:

$$K(x) = \frac{1}{\texttt{bandwidth} \cdot \sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x}{\texttt{bandwidth}}\right)^2\right)$$

  which smooths returns.
- Kernel Density Convolution — `compute_kde_convolution()` Integrates the above steps: returns are passed through the Gaussian kernel via Numpy convolution, yielding continuous density estimates that suppress noise and highlight patterns.
- Seasonal Reorganization — `reorganize_by_calendar_day()` Groups smoothed returns by calendar date (e.g., all January 15 prices grouped together across years). This transformation converts time series returns into a calendar based view, which is helpful in finding seasonal patterns independent of long term market drift.

This program takes command line arguments to specify the stock symbol, the training period, and whether to show the figures. For example, the following command runs the program for Microsoft (MSFT), trains it using data from January 1, 2020 until now, and displays the output:

```
python desnity_of_returns.py MSFT --train 2020-01-01,now --show
```

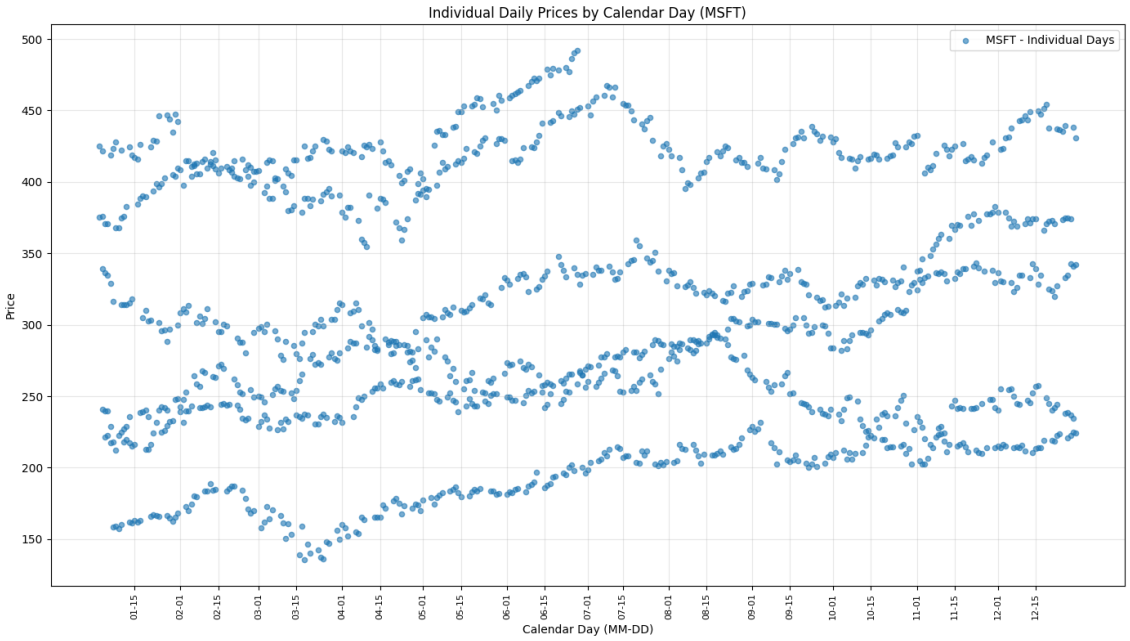

FIGURE 11.1. MSFT Individual Daily Prices by Calendar Day (Jan 2020 - July 2025)

11.3. **Analysis and Results.** This scatter plot demonstrates what reorganizing financial data by calendar date rather than chronological sequence looks like and helps to reveal hidden seasonal patterns. The visualization shows distinct vertical bands that indicate seasonal behavior. Notable patterns emerge where early February consistently exhibits higher price clusters as well as the end of spring months to summer of April to July. The clear vertical banding patterns indicate that market participants exhibit recurring behavioral patterns tied to calendar events, earnings seasons, and re-balancing cycles that repeat annually regardless of the underlying chronological progression. Another useful insight into why February, April/May, July, and October tend to have higher prices is that is when MSFT historically has earnings.

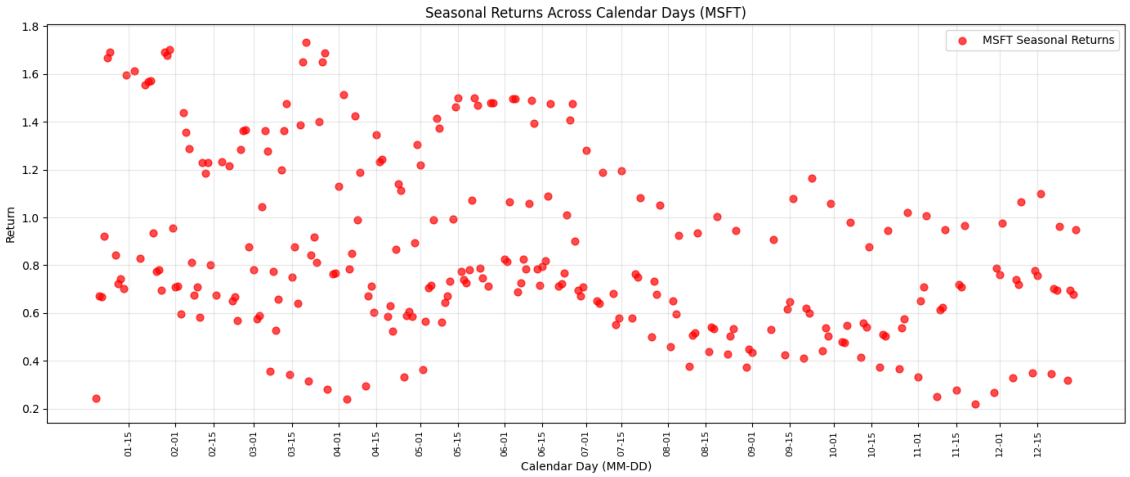

FIGURE 11.2. MSFT Seasonal Returns Across Calendar Days - Kernel Density Estimated (Jan 2020 - July 2025)

The return data reorganized by calendar day reveals seasonal variations that directly correlate with corporate earnings cycles, demonstrating a key insight from seasonal analysis . The chart shows high return spikes during January-February (1.6-1.7x), April-May (1.4-1.5x), July-August (1.2x), and October periods (1.1-1.2x) - aligning with quarterly earnings announcement windows when companies typically report financial results. This pattern indicates that seasonal analysis can effectively identify earnings-driven volatility cycles that create predictable return distributions throughout the year. The kernel density estimation approach successfully captures these earnings related seasonal effects by smoothing out daily noise and preserving the underlying quarterly reporting cycles. Between earnings periods, returns are in lower ranges (0.4-0.8x), suggesting that seasonal analysis can distinguish between high volatility periods driven by corporate events versus lower volatility intervals, providing insights for timing strategies and risk management.

12. Analysis of Economic Indicators: TSA Passenger Volumes

This analysis relates to the following files: **extract_tsa.py** and **tsa_data_analysis.py**

12.1. **Scenario.** The purpose of this analysis is to analyze passenger travel volumes from the TSA (passenger volumes in the USA) to assess economic sentiment and indicators. If more people are traveling now than in recent years, then ordinary people may be less worried about the economy and have more disposable income. Many large firms and government agencies analyze similar alternative economic indicators for their analysis of the market or economy. Examples include passenger volumes, number of new car purchases, mortgage applications, credit card debt, number of ships in ports, number of new jobs, etc.

12.2. **Experimentation.** The analysis of TSA passenger volumes was conducted using historical travel data from the TSA's publicly available datasets on their website (TSA Passenger Volumes). There is currently data available from 2019 - 2025 and it is updated every weekday.

Data extraction was performed using web scraping techniques implemented in Python, utilizing the Beautiful Soup library to parse HTML tables containing passenger volume information. The extraction script (**extract_tsa.py**) was designed to collect all available data. To ensure reliable data retrieval and avoid being blocked by the website's security measures, the script employed a standard user agent string mimicking a legitimate browser request.

Travel
- Security Screening
- TSA Cares
- TSA PreCheck®
- **Passenger Volumes**
  - 2024
  - 2023
  - 2022
  - 2021
  - 2020
  - 2019
- Travel Tips
- FAQ

TSA checkpoint travel numbers

Passenger travel numbers are updated Monday through Friday by 9 a.m. Travel numbers during holiday weeks though may be slightly delayed.

Date	Numbers
9/4/2025	2,305,711
9/3/2025	1,965,877
9/2/2025	2,276,198
9/1/2025	2,835,928
8/31/2025	2,343,037
8/30/2025	2,223,164
8/29/2025	2,971,217
8/28/2025	2,833,372

Figure 12.1. Example of Data on TSA's Website

The raw data underwent preprocessing to standardize date formats and remove formatting characters such as commas from numerical values. All passenger counts were converted to integers and dates were formatted into a consistent MM/DD/YYYY format for sorting and analysis.

The analysis was built around the concept of centered rolling averages to smooth out daily fluctuations and reveal trends in passenger volumes. A command line configurable window size was implemented, allowing analysis at different scales. For example, a 10 day window uses data from 5 days before and 5 days after each date to calculate the average, reducing noise while preserving seasonal patterns.

Here is an example of how to run the program, where the `-days` parameter specifies the window size described above:

```
python tsa_data_analysis.py --days 10
```

The analysis script (**tsa_data_analysis.py**) processes data by converting calendar dates to day of-year indices (1-366) to facilitate year-over-year comparisons of seasonal patterns. This approach enables direct comparison of specific calendar periods across different years, accounting for leap years and ensuring consistent alignment.

Two separate analyses were conducted: one including 2020 data and another excluding it. This distinction was necessary because 2020 represents an extraordinary outlier due to the COVID-19 pandemic's dramatic impact on air travel, with passenger volumes dropping by as much as 95% during peak lock down periods. When 2020 was excluded, comparisons were made directly between 2019 and 2021. By examining both cases, the study highlights the pandemic's immediate impact as well as underlying pre- and post-pandemic travel trends.

The main analysis function was `centered_rolling_mean()` for smoothing daily fluctuations. Two things where calculated: percent change which was calculated using the standard formula ($\frac{current-previous}{previous} \times 100$) and raw number differences from the same time last year. These metrics provide perspectives on year-over-year growth patterns, with ratios and percent changes providing standard growth rate interpretations. It is worth noting, that at the time of the graph creation it is August, 2025, and as such the 2025 graph is incomplete.

12.3. **Results and Analysis.** The main analysis file produces tabled outputs as well as graphs. Below is an example of the tabled output which highlights the 10 day windows around July 4th, and December 25th, which are some of the highest travel days in the USA. To view the full output, please see the file called **tsa_data_analysis.txt**

Day (±5 days)	Year	Passengers	Gap from previous year	% Change from previous year
Jul 02	2019	2,476,866	-	-
	2020	670,373	-1,806,493	-72.9%
	2021	1,987,706	+1,317,333	+196.5%
	2022	2,234,622	+246,916	+12.4%
	2023	2,510,617	+275,995	+12.4%
	2024	2,700,648	+190,031	+7.6%
	2025	2,678,875	-21,774	-0.8%
Jul 03	2019	2,456,534	-	-
	2020	678,964	-1,777,570	-72.4%
	2021	1,995,369	+1,316,405	+193.9%
	2022	2,234,519	+239,150	+12.0%
	2023	2,520,184	+285,665	+12.8%

Day (±5 days)	Year	Passengers	Gap from previous year	% Change from previous year
	2024	2,688,288	+168,103	+6.7%
	2025	2,672,492	-15,796	-0.6%
Jul 04	2019	2,469,903	-	-
	2020	686,991	-1,782,911	-72.2%
	2021	2,011,032	+1,324,040	+192.7%
	2022	2,253,238	+242,206	+12.0%
	2023	2,513,741	+260,503	+11.6%
	2024	2,655,725	+141,984	+5.6%
	2025	2,664,685	+8,959	+0.3%
Jul 05	2019	2,467,737	-	-
	2020	695,596	-1,772,141	-71.8%
	2021	2,035,927	+1,340,330	+192.7%
	2022	2,240,780	+204,853	+10.1%
	2023	2,468,036	+227,257	+10.1%
	2024	2,653,840	+185,803	+7.5%
	2025	2,674,548	+20,708	+0.8%
Jul 06	2019	2,491,489	-	-
	2020	710,670	-1,780,819	-71.5%
	2021	2,031,213	+1,320,542	+185.8%
	2022	2,201,777	+170,564	+8.4%
	2023	2,456,222	+254,445	+11.6%
	2024	2,654,224	+198,002	+8.1%
	2025	2,691,491	+37,267	+1.4%
Jul 07	2019	2,488,254	-	-
	2020	723,082	-1,765,172	-70.9%
	2021	1,998,067	+1,274,985	+176.3%
	2022	2,203,519	+205,452	+10.3%
	2023	2,474,067	+270,547	+12.3%
	2024	2,660,986	+186,919	+7.6%
	2025	2,717,688	+56,702	+2.1%
Jul 08	2019	2,499,329	-	-
	2020	717,494	-1,781,835	-71.3%
	2021	1,998,151	+1,280,657	+178.5%
	2022	2,230,756	+232,605	+11.6%
	2023	2,514,959	+284,203	+12.7%
	2024	2,665,142	+150,183	+6.0%
	2025	2,712,435	+47,293	+1.8%
Dec 21	2019	2,421,207	-	-
	2020	997,994	-1,423,213	-58.8%
	2021	2,025,341	+1,027,347	+102.9%
	2022	2,181,418	+156,077	+7.7%
	2023	2,457,199	+275,781	+12.6%
	2024	2,505,150	+47,951	+2.0%
	2025	-	-	-
Dec 22	2019	2,463,394	-	-

Day (±5 days)	Year	Passengers	Gap from previous year	% Change from previous year
	2020	1,057,495	-1,405,899	-57.1%
	2021	2,004,006	+946,511	+89.5%
	2022	2,174,129	+170,124	+8.5%
	2023	2,469,716	+295,587	+13.6%
	2024	2,545,779	+76,063	+3.1%
	2025	-	-	-
Dec 23	2019	2,494,542	-	-
	2020	1,081,091	-1,413,451	-56.7%
	2021	2,002,957	+921,866	+85.3%
	2022	2,157,888	+154,931	+7.7%
	2023	2,494,291	+336,403	+15.6%
	2024	2,602,128	+107,837	+4.3%
	2025	-	-	-
Dec 24	2019	2,497,057	-	-
	2020	1,076,356	-1,420,701	-56.9%
	2021	1,992,458	+916,102	+85.1%
	2022	2,168,675	+176,218	+8.8%
	2023	2,519,064	+350,389	+16.2%
	2024	2,643,407	+124,343	+4.9%
	2025	-	-	-
Dec 25	2019	2,441,785	-	-
	2020	1,084,862	-1,356,924	-55.6%
	2021	1,952,322	+867,460	+80.0%
	2022	2,143,134	+190,812	+9.8%
	2023	2,482,932	+339,798	+15.9%
	2024	2,653,799	+170,867	+6.9%
	2025	-	-	-

The July 4th holiday period represents one of the heaviest travel times of the year, making it an excellent case study for economic sentiment analysis. Passenger volumes from July 2–8 highlight both the disruption and bounce back of U.S. air travel. 2019 levels averaged 2.47–2.50M daily passengers. The pandemic collapse in 2020 cut volumes by ˜72%, but recovery was swift: 2021 rebounded to ˜2.0M daily (still ˜18% below 2019). Growth in 2022–2023 brought volumes back above pre-pandemic levels, showing a full recovery and expansion. By 2024, travel hit record highs (2.65–2.70M, ˜7–8% above 2019). In 2025, growth plateaued: some days dipped slightly while others slightly higher, suggesting stabilization at historically strong levels and normal growth increases.

Christmas holidays are another peak travel period marked by high discretionary spending. December 21–25 volumes mirror summer trends but with a faster, and perhaps more emotional, recovery. 2019 set a strong baseline at 2.42–2.50M daily passengers. In 2020, travel fell to ˜1.0–1.1M (-55–59%), but by 2021 volumes nearly doubled, as many likely prioritized family reunions after prolonged isolation due to the pandemic. Recovery continued in 2022, still slightly below pre-pandemic levels. Then breakthrough came in 2023, when volumes exceeded 2019 for the first time with 13–16% growth. By 2024, passenger counts reached 2.51–2.65M, 3–9% above pre-pandemic records, showing normalized family travel and strong consumer spending power during this holiday period.

Here is the percent changes over the entire year visually displayed:

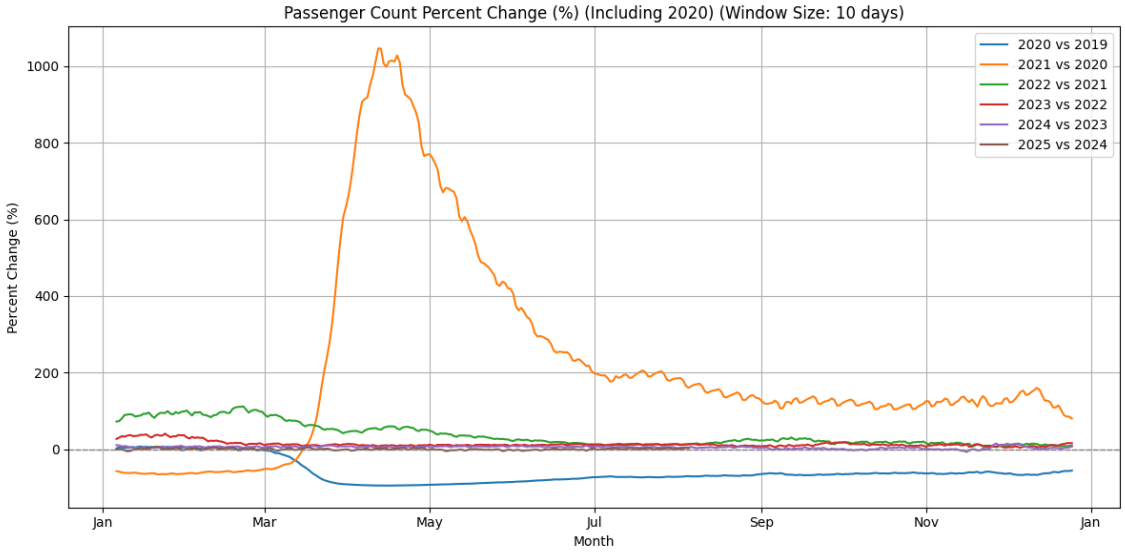

FIGURE 12.2. Passenger Count % Change Compared to Previous Year Including 2020

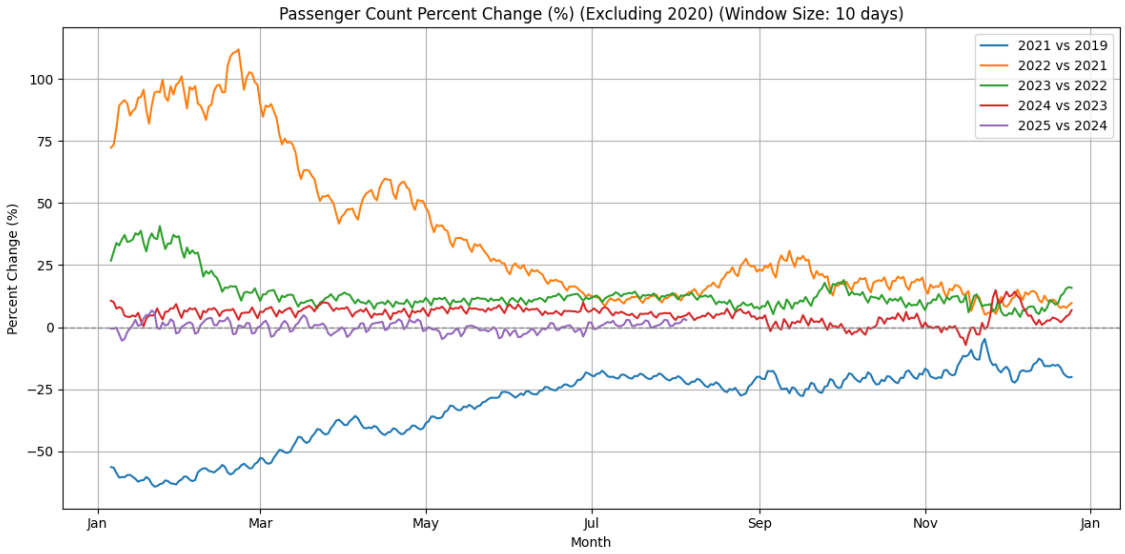

FIGURE 12.3. Passenger Count % Change Compared to Previous Year Excluding 2020

The first visual (Figure 12.2) reveals the dramatic impact of the COVID-19 pandemic on air travel . The most striking feature is the orange line (2021 vs 2020), which shows extraordinary

percent changes reaching above 1100% during the spring months (March-April). This represents the period when travel began recovering from the complete collapse experienced in 2020. The peak around April corresponds to the anniversary of the initial pandemic lock downs, when 2020 passenger volumes hit their absolute lowest points.

The 2020 vs 2019 comparison (blue line) shows the inverse relationship, dropping to near-zero ratios during the peak lock down period, illustrating how passenger volumes fell by over 95% compared to pre-pandemic levels. This creates the effect that makes subsequent recovery appear dramatically amplified in the percent change calculations.

Other year comparisons (2022-2025) cluster much closer to the baseline, indicating more normal year-over-year growth patterns once the industry stabilized post-pandemic. The seasonal variations visible in these lines reflect typical travel patterns, with slight increases during summer months and holiday periods.

The percent change view excluding 2020 (Figure 12.3) provides more insights when accounting for "normal" times. The 2022 vs 2021 growth (orange line) shows sustained 40-100% growth rates early in the year, moderating to 10-20% by year-end. This pattern reflects the strong but decelerating recovery as travel demand approached pre-pandemic levels.

The 2021 vs 2019 comparison demonstrates that recovery was incomplete, with passenger volumes running 20-55% below pre-pandemic levels throughout 2021. The gradual improvement throughout the year indicates steady recovery and growing consumer confidence to travel.

Recent years show more typical economic growth patterns: 2023 vs 2022 growth rates of 5-15%, and 2024 vs 2023 showing modest positive growth of 0-15%. These normal growth rates suggest the travel industry has returned to pre-pandemic economic cycles, with growth rates consistent with broader economic expansion rather than recovery dynamics.

The seasonal patterns visible across all years reflect traditional travel behaviors, with higher growth rates during summer months and holiday periods, indicating that consumer discretionary spending patterns have fully returned to normal.

Travel patterns highlight TSA passenger volumes as a powerful real-time economic indicator. Both July 4th and Christmas periods show that demand for costly holiday travel not only recovered from the pandemic shock but has since surpassed pre-2019 levels, signaling robust consumer confidence and spending power. Americans' willingness to prioritize travel during high-expense periods points to economic strength that reflects growth, not just recovery. This showcases the value of alternative, real-time data sources: travel acts as a natural experiment in consumer behavior, offering faster and more revealing insights into economic health than traditional lagged indicators. Other potential indicators include credit card spending data, hotel occupancy rates, restaurant reservations, number of new car purchases, mortgage applications, number of ships in ports, etc., which together can provide a broader picture of discretionary consumption and economic sentiment.

To view the complete output of the table, please see the file called: **tsa_data_analysis.txt**

13. Final Remarks