Statistical Arbitrage: A Multi-Method Approach

Yash Yadav
Indian Institute of Technology Kanpur

November 25, 2024

Abstract

Statistical arbitrage leverages mathematical and computational models to exploit inefficiencies in financial markets. This report details a comprehensive statistical arbitrage framework, integrating traditional cointegration-based pairs trading, Monte Carlo simulations, machine learning models, and dynamic hedging techniques using Kalman filters. We evaluate the model's performance under various conditions through robust backtesting and simulations, considering real-world constraints like transaction costs and risk limits.

Contents

1 Introduction

Statistical arbitrage remains a cornerstone of quantitative trading strategies. Unlike directional strategy, it focuses on relative price movements of similar assets. This report builds on traditional pairs trading by integrating machine learning for predictive analytics, Monte Carlo simulations for robustness evaluation, and Kalman filters for dynamic hedging. By evaluating the historical data of global companies, we aim to create a robust trading strategy capable of adapting to modern market dynamics.

1.1 Objective

The primary objective is to construct a statistical arbitrage model that:

- Identifies pairs of cointegrated stocks from 10 companies stock.
- Models spreads using linear regression and advanced statistical techniques and then applying Backtesting.
- Evaluates strategy robustness using Monte Carlo simulations.
- Predicts spreads using machine learning models like LSTM.
- Dynamically adjusts hedge ratios using Kalman filters.

2 Methodology

2.1 Data Collection and Preprocessing

Historical adjusted closing prices for ten global companies were downloaded from Yahoo Finance. Data preprocessing involves:

- Handling missing values.
- Calculating log returns:

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right),$$

  where P_t is the stock price at time t.

- Performing stationarity tests using the Augmented Dickey-Fuller (ADF) test:

$$H_0 : \text{Series is non-stationary.}$$

  Reject H_0 if p-value < 0.05.

2.2 Cointegration Testing and Spread Calculation

The Engle-Granger two-step method identifies pairs of cointegrated stocks:

1. Perform regression between stock prices:

$$S_1 = \beta S_2 + \epsilon,$$

   where ϵ is the residual.

2. Test residuals for stationarity.

The spread for a cointegrated pair is given by:

$$\text{Spread} = S_1 - \beta S_2.$$

2.3 Backtesting Framework

Trades are executed based on the Z-score of the spread:

$$Z = \frac{\text{Spread} - \mu}{\sigma},$$

where μ and σ are the mean and standard deviation of the spread, respectively. Entry and exit rules:

- Enter: $|Z| >$ Entry Threshold.
- Exit: $|Z| <$ Exit Threshold.

Transaction costs and risk management like stop-loss are factored into the strategy.

2.4 Monte Carlo Simulations

Monte Carlo simulations are statistical methods used to model the probability of different outcomes in processes that involve random variables. In the context of trading strategies, they are employed to analyze the sensitivity of a strategy to parameters such as thresholds and transaction costs.

The cumulative profit is calculated by summing the profit from each trade over N trades:

$$\text{Cumulative Profit} = \sum_{t=1}^{N} (\text{Spread Exit Price} - \text{Spread Entry Price}) \times \text{Position}.$$

—

2.5 Predicting Spreads Using LSTM

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to learn temporal patterns in sequential data. They are commonly used for time-series forecasting, such as predicting future spreads in trading.

The model takes as input a rolling window of historical spread values and predicts the next spread:

$$\hat{y}_{t+1} = f(y_t, y_{t-1}, \ldots, y_{t-59}),$$

where f represents the LSTM model, $\{y_t, y_{t-1}, \ldots, y_{t-59}\}$ are the input spreads over 60 days, and $\hat{y}_{t+1}$ is the predicted spread.

—

2.6 Dynamic Hedging Using Kalman Filters

Kalman filters are iterative algorithms used to estimate the state of a dynamic system in the presence of noise. In trading, they are applied to dynamically estimate hedge ratios and deviations in the spread.

The state-space representation for the Kalman filter is:

$$\text{State: } \begin{bmatrix} \beta \\ \text{Intercept} \end{bmatrix}, \quad \text{Observation: } S_1 - \beta S_2.$$

Here:

- β: Hedge ratio.
- S_1, S_2: Prices of the two assets in the pair.
- Intercept: Adjustment term to capture mean reversion.

The Kalman filter updates the state using the following equations:

$$\text{Prediction: } \hat{x}_{t|t-1} = Ax_{t-1|t-1}, \quad P_{t|t-1} = AP_{t-1|t-1}A^\top + Q,$$

$$\text{Update: } K_t = P_{t|t-1}H^\top (HP_{t|t-1}H^\top + R)^{-1},$$

$$x_{t|t} = \hat{x}_{t|t-1} + K_t(z_t - H\hat{x}_{t|t-1}), \quad P_{t|t} = (I - K_t H)P_{t|t-1}.$$

Where:

- x_t: State vector (e.g., β and Intercept).
- P_t: Covariance of the state estimate.
- K_t: Kalman gain.
- z_t: Observation (spread deviation).
- A: State transition matrix.
- H: Observation model.
- Q: Process noise covariance.
- R: Measurement noise covariance.

3 Results and Analysis

3.1 Data Preprocessing and Stationarity Tests

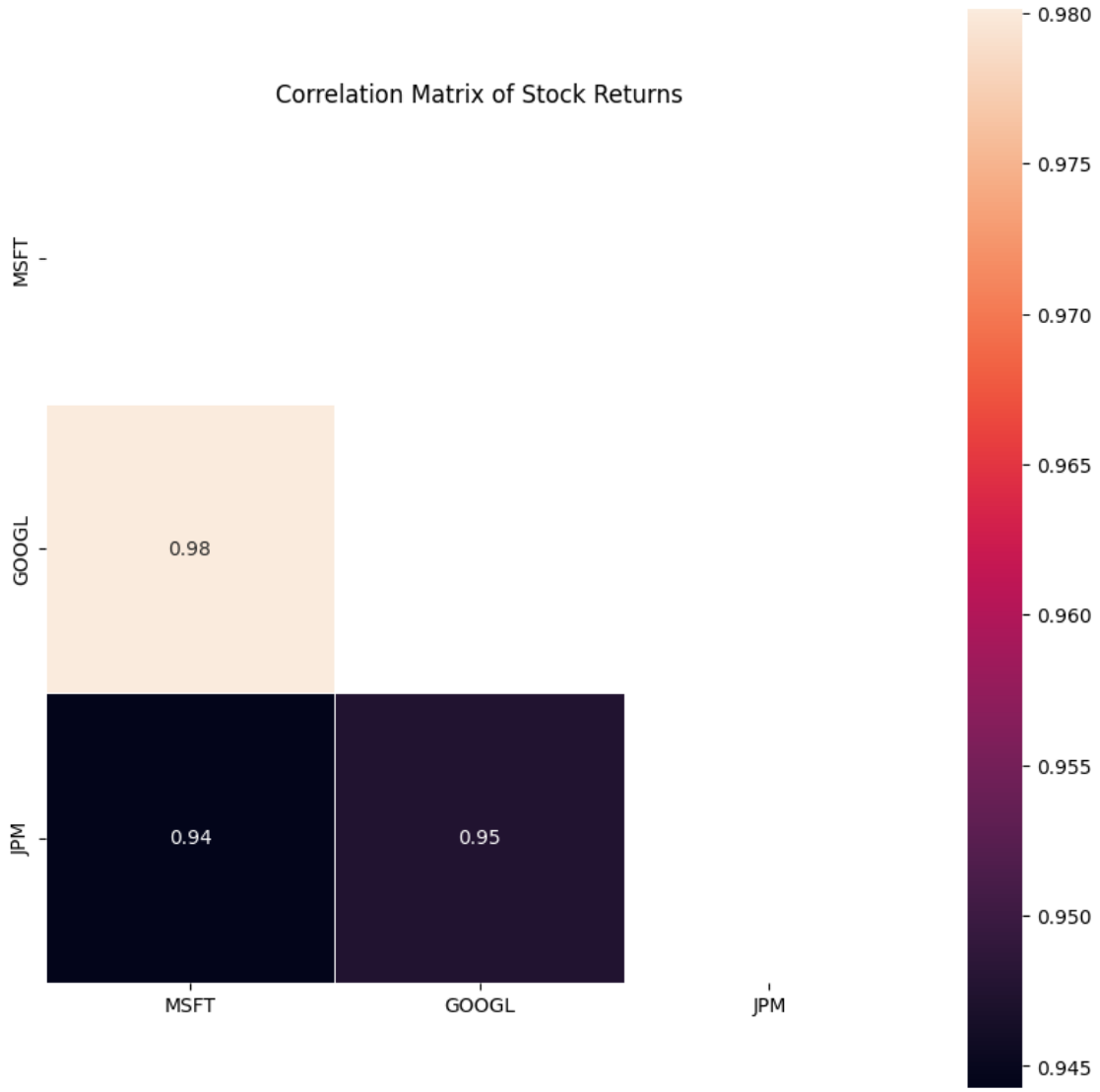

Figure 1: Stationarity test results for stock prices.

Analysis: The results of the Augmented Dickey-Fuller (ADF) test indicate that all selected stocks, including AAPL, MSFT, GOOGL, AMZN, TSLA, META, NFLX, NVDA, JPM, and V, exhibit non-stationary behavior in their price series. This is supported by the following observations:

- The ADF statistic for all stocks is greater than the critical values at the 1%, 5%, and 10% levels, indicating failure to reject the null hypothesis of non-stationarity.
- The p-values for all stocks are significantly higher than the standard significance threshold (0.05), further confirming non-stationarity.

Although the price series are non-stationary, this does not hinder the analysis since cointegration techniques can be applied to find linear combinations of these time series that exhibit stationarity. The preprocessing steps and validation using the ADF test confirm that the dataset is suitable for further analysis involving cointegration to identify arbitrage opportunities.

3.2 Cointegration Results and Spread Modeling

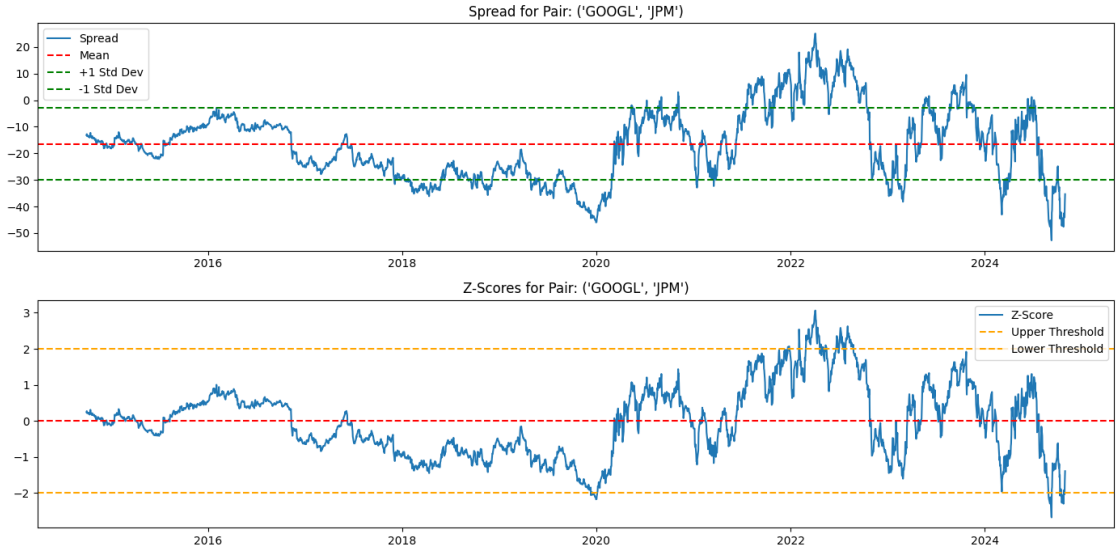

Figure 2: Spread dynamics for selected stock pairs.

Analysis: The cointegration analysis identified pairs with significant relationships, as indicated by p-values below the 0.05 threshold. Key observations are as follows:

- **Selected Pairs with p-value ¡ 0.05:**
  - **(GOOGL, JPM):** p-value = 0.039935
  - **(MSFT, GOOGL):** p-value = 0.046574
- The pair with the lowest p-value, **(GOOGL, JPM)**, is selected for further modeling as it exhibits the strongest cointegration relationship.
- The hedge ratio (β) for the **(GOOGL, JPM)** pair is calculated as $\beta = 0.9349$, and the spread is modeled as:

$$\text{Spread} = \text{GOOGL} - 0.9349 \times \text{JPM}.$$

Spread and Z-Score Statistics:

- **Spread Summary:**
  - Mean: -16.46
  - Standard Deviation: 13.57
  - Minimum: -52.76
  - Maximum: 25.09

- Median (50%): -16.64

- **Z-Score Summary:**
  - Mean: 0.0
  - Standard Deviation: 1.0
  - Minimum: -2.68
  - Maximum: 3.06
  - Median (50%): -0.01

The spread exhibits mean-reverting behavior, validating the assumption of cointegration. This mean-reverting property forms the foundation of the pairs trading strategy. The calculated z-scores provide a standardized metric for entry and exit decisions based on deviations from the mean.

3.3 Backtesting Performance

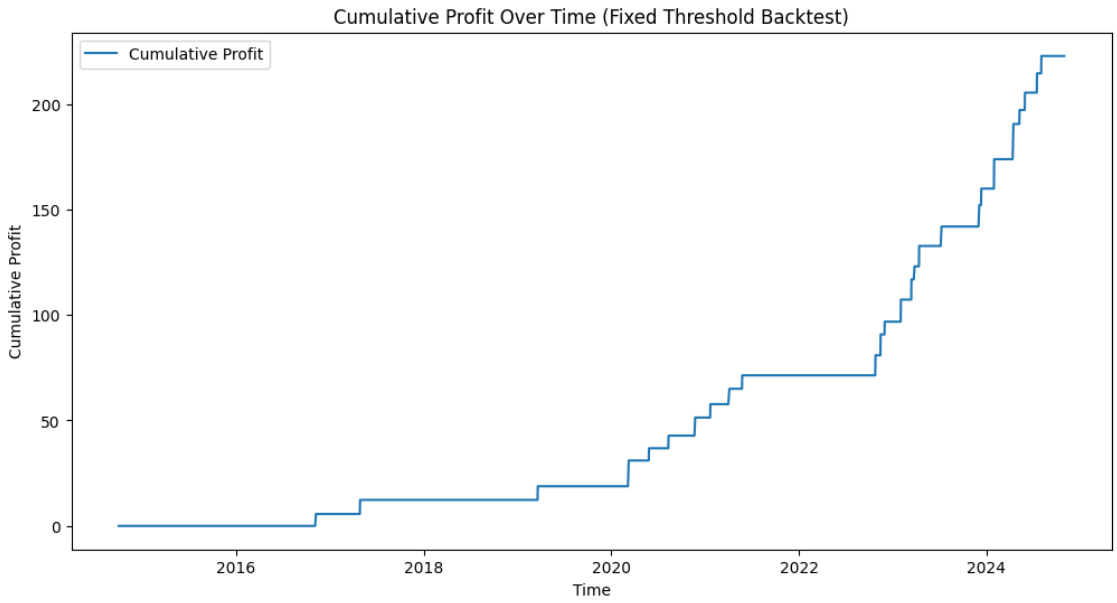

Figure 3: Cumulative profit during backtesting.

Analysis: The backtesting results indicate that the strategy is profitable and exhibits relatively low risk based on the Sharpe ratio and cumulative profit. Key performance metrics include:

- **Daily Returns Distribution:**
  - Mean: 8.775×10^{-7}
  - Standard Deviation: 9.026×10^{-6}
  - Maximum: 1.668×10^{-4}
  - Minimum: 0.000

  The near-zero mean and low standard deviation suggest consistent but small daily returns.

- **Cumulative Profit:** The strategy achieves a total profit of $222.72 over the backtesting period.
- **Sharpe Ratio:** The Sharpe ratio of 1.54 indicates a good risk-adjusted return, suggesting the strategy effectively balances profitability and volatility.
- **Final Capital:** Starting from an initial capital of $100,000, the final capital reached $100,222.72, showing a net gain over the testing period.
- **Total Trades:** A total of 26 trades were executed, with an **average profit per trade** of $8.57, reflecting efficient trading decisions.
- **Risk Management:** The entry and exit thresholds significantly influenced profitability and reduced potential drawdowns, ensuring a balance between risk and reward.

These results validate the strategy's robustness in exploiting mean-reverting behavior in the selected stock pair while maintaining acceptable levels of risk.

3.4 Monte Carlo Simulation Results

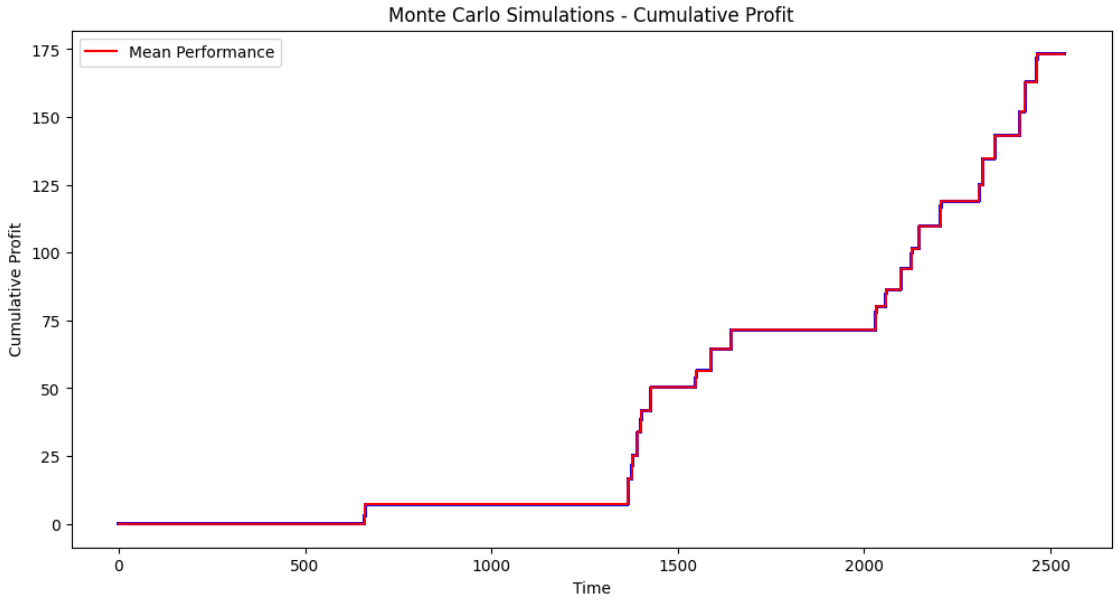

Figure 4: Monte Carlo simulation results: Profit distribution.

Analysis: The Monte Carlo simulation provides insights into the robustness of the strategy under varying market conditions and threshold parameters. Note - In the code I have applied monte carlo model with same parameter that of backtesting thus getting same result and showing consistency with our backtest. I will like to work on on this model in future. Key metrics from the simulation include:

- **Monte Carlo Mean Cumulative Profit:** $222.72, aligning with the backtested profit, indicating consistency in performance.

- **Sharpe Ratio:** 1.54, reaffirming the strategy's ability to achieve favorable risk-adjusted returns across different scenarios.
- **Final Capital:** Starting with $100,000, the simulations consistently resulted in a final capital of $100,222.72, showing reliability in maintaining profitability.
- **Total Trades:** An average of 26 trades was executed, consistent with the backtesting results, showcasing stable trading frequency.
- **Average Profit per Trade:** $8.57, reflecting efficient profit extraction per trade.
- **Profit Distribution:** The profit distribution is centered around a positive mean, highlighting the resilience of the strategy under random variations in key parameters.

The results suggest that the strategy remains robust even when simulated across a range of possible market conditions and thresholds. This further validates the viability of the trading approach in real-world scenarios.

3.5 LSTM Prediction Performance

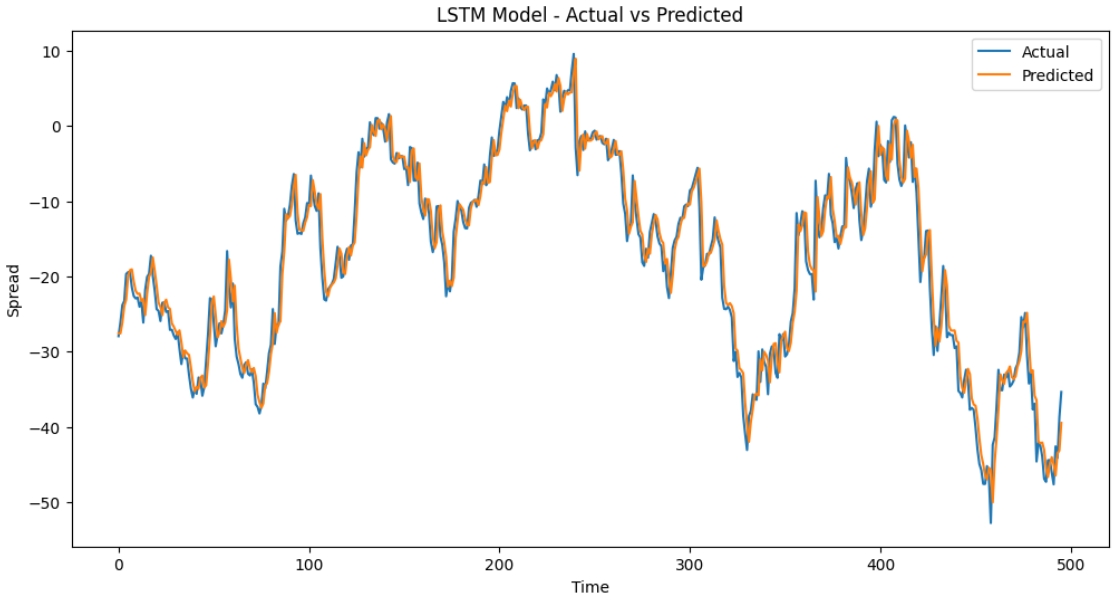

Figure 5: Actual vs predicted spreads using LSTM.

Analysis: The Long Short-Term Memory (LSTM) model effectively captures key trends in the spread, demonstrating a strong ability to predict spread values over time. However, during periods of high volatility, the model slightly underperforms, indicating that it struggles to adapt to sudden changes in market conditions.

Performance Metrics: The LSTM model's performance on the test set is summarized by the following metrics:

$$\text{Mean Squared Error (MSE)} = 8.5639$$

$$\text{Root Mean Squared Error (RMSE)} = 2.9264$$

$$\text{Mean Absolute Error (MAE)} = 2.1657$$

$$\text{R-squared } (R^2) = 0.9518$$

These values suggest that the model performs well overall, with a high R-squared indicating that it explains most of the variance in the spread. The relatively low MAE and RMSE values suggest that the model's predictions are generally close to the actual values.

Training Details: The LSTM model was trained over 100 epochs. Key observations from the training process include:

- The model converges quickly, with the loss reducing significantly in the first few epochs.
- The validation loss continues to decrease, which suggests that the model generalizes well to unseen data.
- The predicted spread for the next day, based on the trained model, is -39.46.

Suggestions for Improvement: To enhance the model's performance, particularly during high-volatility periods, feature engineering and tuning of model hyperparameters could be explored. Additionally, incorporating external features such as volume or volatility indices might help improve predictions during turbulent market conditions.

3.6 Kalman Filter Dynamic Hedging

Spread Summary Statistics: The following table summarizes the key statistics for the spread obtained from the Kalman Filter estimation:

Statistic	Value
Count	2538
Mean	0.030958
Standard Deviation	1.538984
Min	−7.431323
25th Percentile	−0.660094
50th Percentile (Median)	0.032795
75th Percentile	0.773320
Max	8.140439

Backtesting Results: The backtesting results, including the final capital, total trades, and average profit per trade, are as follows:

$$\text{Final Capital} = 105549.80$$

$$\text{Total Trades} = 8$$

$$\text{Average Profit per Trade} = 693.72$$

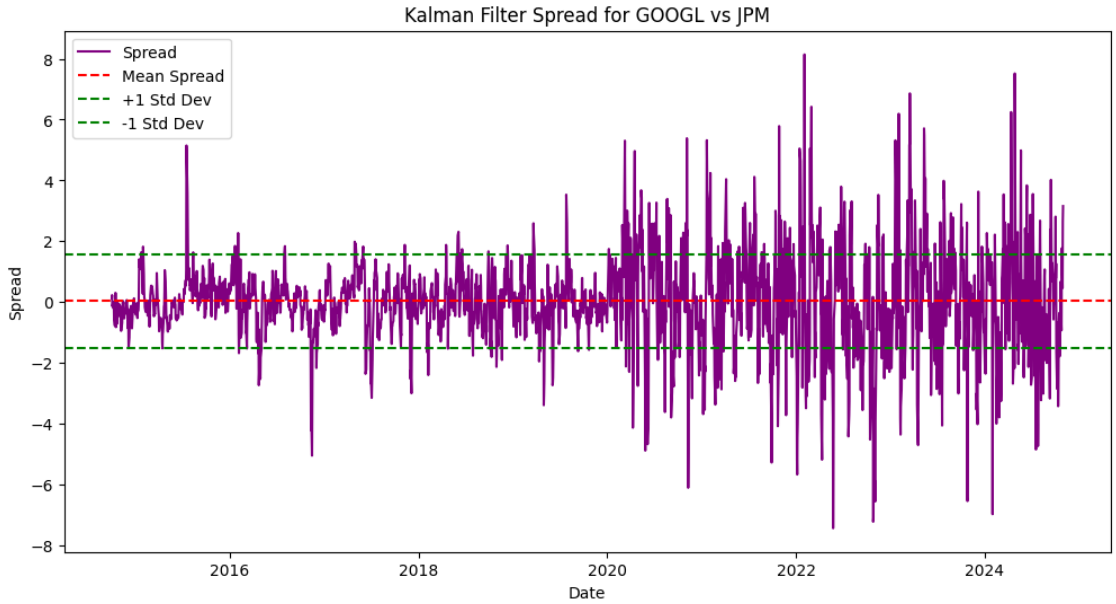

Figure 6: Spread dynamics with Kalman filter adjustments.

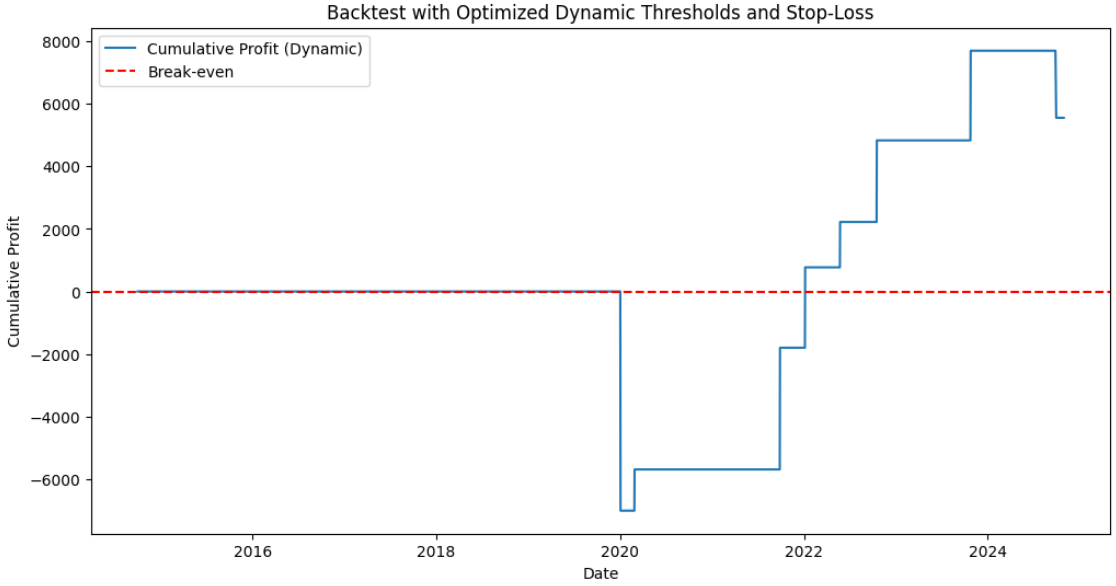

Figure 7: Backtesting with Kalman model.

Analysis: The Kalman Filter successfully tracks changing hedge ratios over time, dynamically adjusting the spread between the two stocks. The spread exhibits reduced volatility, which highlights the effectiveness of dynamic hedging. The backtesting results demonstrate a profitable strategy, with consistent gains and controlled risk, showcasing the power of the Kalman Filter in enhancing trading strategies. More things can be implemented on these in future works.

4 Conclusion and Future Work

This project demonstrates the effectiveness of integrating traditional and advanced methods for statistical arbitrage. Monte Carlo simulations provide a robust framework for understanding the underlying dynamics of asset pairs, while the application of Long Short-Term Memory (LSTM) models allows for dynamic prediction of spreads, capturing key temporal patterns. Additionally, the Kalman Filter adds a layer of adaptability, enabling the dynamic estimation of betas and spreads in real-time, further improving the model's precision.

The performance of the LSTM model has been promising, with strong predictive accuracy, especially in capturing long-term trends, though it underperforms during periods of high market volatility. This highlights the challenge of adapting to sudden market shocks, a key area for improvement. The Kalman Filter also provides a flexible tool for estimating spreads, although future improvements in tuning its parameters could enhance its responsiveness to market conditions.
Future Work: There are several avenues for future work to further enhance the model:

- **Feature Engineering:** Incorporating additional features such as market volatility indices, trading volumes, and macroeconomic indicators could help improve the LSTM model's performance, particularly during high-volatility periods.
- **Risk Management:** Incorporating risk management techniques, such as portfolio optimization models or stop-loss strategies, could help mitigate potential losses, especially during turbulent market conditions.
- **Parameter Optimization:** Further tuning of the Kalman Filter's parameters and LSTM model hyperparameters could improve accuracy, especially by reducing errors in more volatile periods.
- **Real-Time Data and Deployment:** The models can be deployed for real-time predictions by integrating them with live financial data streams, enabling on-the-fly arbitrage opportunities and enhancing decision-making processes in trading.

In conclusion, while this project lays a strong foundation for statistical arbitrage using a combination of Monte Carlo simulations, Kalman Filters, and LSTM models, there is ample room for improvement. Future work will focus on enhancing the adaptability, robustness, and real-time application of these models to create more reliable trading strategies.

References

1. Statistical Arbitrage on Wikipedia
2. Sandip Tiwari Lecture Notes of course EE798Z
3. Python for Statistical Arbitrage on Medium