Pairs Trading with Kalman Filter(UID:15)

Ayushmaan Dhanai(23b0397)

February 13, 2025

1 Introduction

Pairs trading is a market-neutral strategy that identifies cointegrated stock pairs for statistical arbitrage. This report presents a step-by-step development from static cointegration analysis to an adaptive **Kalman Filter-based pairs trading system** with **risk management** and **backtesting**.

2 Assignment 1: Cointegration Analysis

2.1 Objective

Assignment 1 primarily focuses on cointegration analysis within financial time series, particularly examining stock price relationships. The core objective is to identify stock pairs that exhibit cointegration, a statistical property that suggests a stable long-term relationship between two financial assets. Cointegration analysis is critical in statistical arbitrage, where traders exploit short-term deviations from the mean to generate profits. By identifying stock pairs that move together over time, traders can develop a robust strategy for pairs trading, assuming that any divergence between them will eventually revert to the mean. This assignment sets the foundation for more advanced trading strategies by selecting asset pairs with statistically significant relationships.

2.2 Methodology

- **Importing Libraries** : To conduct this analysis, essential libraries such as numpy, pandas, statsmodels, yfinance, matplotlib, and seaborn were imported. These libraries facilitate numerical computations, time series analysis, data visualization, and statistical hypothesis testing. Additionally, warnings were suppressed to enhance readability and ensure a cleaner output during execution.
- **Fetching Stock Data**: Financial data was retrieved using pandas_datareader and yfinance, both of which provide seamless access to stock market data. A dataset consisting of stock prices for 11 major companies—Apple

(AAPL), Adobe (ADBE), Oracle (ORCL), eBay (EBAY), Microsoft (MSFT), Qualcomm (QCOM), HP (HPQ), Juniper Networks (JNPR), AMD, IBM, and SPY (an ETF tracking the S&P 500 index)—was compiled for the period between 2010 and 2019. This historical data provides a broad timeframe to assess potential long-term relationships between stock pairs.

- **Cointegration Testing** : A function, find_cointegrated_pairs(), was developed to test stock pairs for cointegration using the Engle-Granger Cointegration Test. This function computed cointegration scores and p-values, storing them in matrices. Stock pairs with p-values below a threshold of 0.05 were considered cointegrated. Identifying such pairs is crucial because it enables the construction of statistical arbitrage strategies, where deviations from historical price relationships can be used to execute profitable trades.

- **Visualization of Cointegration Results** : To enhance interpretability, a heatmap of p-values was generated using seaborn. This visualization illustrated which stock pairs exhibited statistically significant cointegration relationships. Additionally, the list of identified cointegrated stock pairs was printed for reference. The heatmap provided a clear and intuitive representation of cointegration strength across different stock pairs, making it easier to select candidate pairs for further analysis.

2.3 Results

The cointegration analysis successfully identified stock pairs likely to maintain a stable long-term relationship. This information is valuable in developing pairs trading strategies, where one stock is bought while the other is sold short when a temporary divergence occurs. The heatmap representation provided an effective way to visualize and interpret the results, reinforcing the significance of cointegration testing in financial time series analysis. Overall, this work established a foundational understanding of statistical arbitrage strategies and prepared the groundwork for more sophisticated trading models in subsequent assignments.

3 Assignment 2: Kalman Filter for Dynamic Hedge Ratio

3.1 Objective

Building upon the foundation established in Assignment 1, Assignment 2 explores the use of Kalman Filter Regression to dynamically model the hedge ratio between two stocks—Adobe (ADBE) and Microsoft (MSFT). The Kalman Filter is a powerful tool that allows for adaptive estimation of parameters in a time series. Unlike the static cointegration approach in Assignment 1, this method refines the pairs trading strategy by dynamically adjusting the hedge ratio over time. This approach enhances trading performance by accounting for changing market conditions and improving decision-making in statistical arbitrage strategies.

3.2 Methodology

- **Importing Libraries**: The assignment made use of specialized libraries such as pykalman for implementing Kalman Filters. Additionally, standard financial and statistical libraries like pandas, yfinance, statsmodels, numpy, and sklearn were utilized for data handling, time series modeling, and machine learning integration. Some machine learning tools, including RandomForestRegressor and train_test_split, were also imported but were not heavily used in this assignment.

- **Fetching Stock Data**: Stock price data for Adobe (ADBE) and Microsoft (MSFT) was fetched from Yahoo Finance, covering the period from 2013 to 2019. This dataset was chosen based on findings from Assignment 1, where these stocks demonstrated a significant cointegration relationship. By limiting the dataset to a refined stock pair, the assignment aimed to implement a more focused and sophisticated trading strategy.

- **Kalman Filter Regression**: Defined the hedge ratio dynamically using:

$$Y_t = \beta_t X_t + \alpha_t + \epsilon_t \tag{1}$$

  where Y_t is MSFT, X_t is ADBE, and β_t (hedge ratio) evolves over time.

- **Kalman Filter Regression for Hedge Ratio**: A function, kalman_filter_regression(), was defined to estimate a dynamic hedge ratio using Kalman Filters. This function modeled stock X (ADBE) as the independent variable and stock Y (MSFT) as the dependent variable, allowing for real-time estimation of their relationship. By implementing a state-space representation, the function provided time-varying hedge ratios, which were more responsive to market conditions compared to the static approach in Assignment 1.

- **Trading Strategy & Profit Calculation**: A spread-based trading strategy was implemented using the dynamically estimated hedge ratio. The strategy generated trading signals based on the spread's position relative to statistical thresholds:

  A **buy signal** was triggered when the spread dropped below its mean by two standard deviations.

  A **sell signal** was triggered when the spread exceeded its mean by two standard deviations.

  Positions were closed when the spread crossed back to the mean.

  Portfolio performance was evaluated using cumulative returns, accounting for transaction costs (0.05%) to ensure realistic profitability estimation.

3.3 Results

The use of the Kalman Filter enabled a more adaptive approach to pairs trading by dynamically adjusting the hedge ratio. This refinement improved trading accuracy and mitigated risks associated with static models. The spread-based strategy provided systematic entry and exit points, reinforcing the practical application of statistical arbitrage. Overall, Assignment 2 demonstrated the advantages of incorporating dynamic state-space models into financial trading strategies.

4 Final Submission: Backtesting with Risk Management

4.1 Objective

The final submission built upon the progress made in Assignments 1 and 2 by integrating risk management mechanisms, a backtesting framework, and enhanced visualization techniques. The objective was to create a fully functional trading system capable of executing and evaluating pairs trading strategies under real-world conditions. The final phase built a full backtesting framework, incorporating:

- Stop-loss Mechanism
- Volatility-Based Position Sizing
- Trade Execution and Cumulative PnL Calculation

4.2 Methodology

- **Spread Calculation**: Using Kalman Filter regression:

$$S_t = |X_t - \beta_t Y_t - \alpha_t| \tag{2}$$

- **Stop-Loss Implementation**: Automatically closed positions when the spread exceeded a predefined threshold. Closed positions when:

$$S_t > \text{Stop Threshold} \tag{3}$$

- **Volatility-Based Position Sizing**: Adjusted trade sizes dynamically based on a 30-day rolling standard deviation. Adjusted trade sizes using:

$$\text{Position Size} = \frac{\text{Risk Factor}}{\sigma_{30}(S_t)} \tag{4}$$

  where $\sigma_{30}(S_t)$ is the rolling 30-day standard deviation.

- **Volatility Filter**: Prevented trading in highly volatile market conditions to reduce risk.

4.3 Backtesting Results

A comprehensive backtesting framework, run_backtest(), was developed to evaluate the trading strategy. This function:

- Strategy was tested on MSFT and AAPL (2013-2019).
- Generated buy/sell signals.
- Applied risk management techniques.
- Computed cumulative profit and loss (PnL) and PnL curve was generated.
- Visualized results, including stock price movements, trade signals, and cumulative returns.

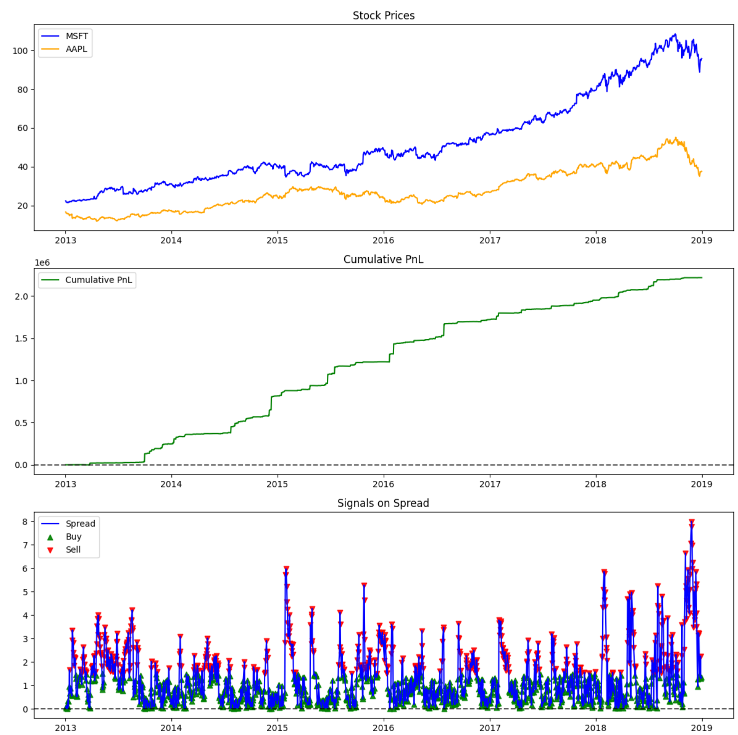

Figure 1: Trading Strategy Performance: Cumulative PnL and Trading Signals

5 Conclusion

This project demonstrated a significant evolution from basic cointegration analysis to a fully implemented and backtested pairs trading strategy. The final submission introduced crucial elements such as dynamic hedge ratios, stop-loss mechanisms, and backtesting frameworks, ensuring that the strategy was both practical and robust. By refining statistical arbitrage methodologies, this work provided a structured approach to financial trading that balances profitability with risk management. Key takeaways include:

- Cointegration Analysis helps identify trading opportunities.
- Kalman Filters improve trade execution by dynamically adjusting hedge ratios.
- Risk Management (stop-loss and volatility filters) enhances strategy robustness.
- Backtesting provides insights into profitability and risk.