Project Report

Robo Advisors & Systematic Trading

Pairs Trading Analysis

Team Members:
Ruby Wu: qw2366
Aditya Suresh: as17339

Content

Introduction

Pairs trading is a popular trading strategy used in financial markets, particularly in equities, currencies, and commodities. The strategy involves identifying two assets (stocks, currencies, or commodities) that historically exhibit a high degree of correlation in their price movements. These assets are referred to as a "pair."

The fundamental idea behind pairs trading is to take advantage of temporary divergences from the historical price relationship between the two assets. When one asset in the pair outperforms the other, the strategy involves buying the underperforming asset while simultaneously selling the outperforming asset. The expectation is that the prices of the two assets will eventually revert to their historical relationship, resulting in profits for the trader.

Problem Statement

Pairs trading is a widely-used strategy in financial markets for exploiting temporary price divergences between correlated assets. However, the success of pairs trading depends heavily on the selection of suitable asset pairs and the effectiveness of the trading strategy employed. In this project, we aim to develop an automated system for selecting the optimal pairs for pairs trading and implementing an effective trading strategy.

Objectives:

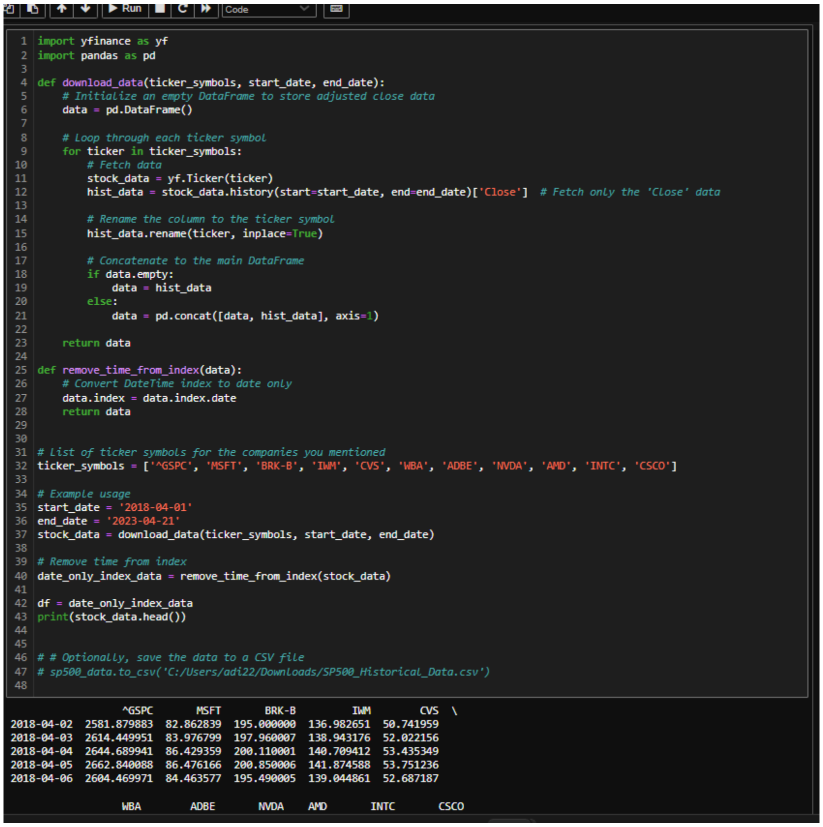
- Pair Selection: Develop algorithms to identify pairs of assets that exhibit a high degree of correlation and cointegration, indicating a potential opportunity for pairs trading.
- Historical Analysis: Conduct comprehensive historical analysis of selected pairs to establish the statistical properties of their price relationship, including mean, standard deviation, and correlation coefficient.
- Trading Strategy Development: Design and implement trading strategies for pairs trading, including entry and exit criteria based on price spread deviations and mean reversion signals.
- Backtesting and Optimization: Backtest the trading strategies using historical data to evaluate their performance and optimize parameters for maximizing risk-adjusted returns.

Expected Outcome:

The project aims to identify potential pair trading candidates, by leveraging statistical models and other python libraries. By implementing new and creative strategies, we seek to maximize profits while considering risks and potential drawdowns.

Data Collection and Preprocessing

- The project involved the development of a Python API to extract stock data from Yahoo Finance using the `yfinance` library.
- The API was designed to download the closing prices of various stocks and indices, randomly selected from a predefined list.
- The selected companies and indices included S&P 500, Microsoft, Berkshire Hathaway, Russell 2000, CVS Health, Walgreens, NVIDIA, AMD, Intel, and Cisco.
- Data was downloaded for the period from April 1st, 2018, to April 21st, 2023. The extracted data was stored in a DataFrame for further analysis and processing.
- The resulting DataFrame (`df`) contains the closing prices of the selected stocks and indices, organized by date. An excerpt of the DataFrame output, denoted as "Figure 1," is provided below:



```python
import yfinance as yf
import pandas as pd

def download_data(ticker_symbols, start_date, end_date):
    # Initialize an empty DataFrame to store adjusted close data
    data = pd.DataFrame()

    # Loop through each ticker symbol
    for ticker in ticker_symbols:
        # Fetch data
        stock_data = yf.Ticker(ticker)
        hist_data = stock_data.history(start=start_date, end=end_date)['Close']  # Fetch only the 'Close' data

        # Rename the column to the ticker symbol
        hist_data.rename(ticker, inplace=True)

        # Concatenate to the main DataFrame
        if data.empty:
            data = hist_data
        else:
            data = pd.concat([data, hist_data], axis=1)

    return data

def remove_time_from_index(data):
    # Convert DateTime index to date only
    data.index = data.index.date
    return data


# List of ticker symbols for the companies you mentioned
ticker_symbols = ['^GSPC', 'MSFT', 'BRK-B', 'IWM', 'CVS', 'WBA', 'ADBE', 'NVDA', 'AMD', 'INTC', 'CSCO']

# Example usage
start_date = '2018-04-01'
end_date = '2023-04-21'
stock_data = download_data(ticker_symbols, start_date, end_date)

# Remove time from index
date_only_index_data = remove_time_from_index(stock_data)

df = date_only_index_data
print(stock_data.head())


# # Optionally, save the data to a CSV file
# sp500_data.to_csv('C:/Users/adi22/Downloads/SP500_Historical_Data.csv')

```

```
                  ^GSPC       MSFT       BRK-B         IWM        CVS  \
2018-04-02  2581.879883  82.862839  195.000000  136.982651  50.741959
2018-04-03  2614.449951  83.976799  197.960007  138.943176  52.022156
2018-04-04  2644.689941  86.429359  200.110001  140.709412  53.435349
2018-04-05  2662.840088  86.476166  200.850006  141.874588  53.751236
2018-04-06  2604.469971  84.463577  195.490005  139.044861  52.687187

                  WBA        ADBE        NVDA    AMD        INTC        CSCO
```

(Figure 1: the code of downloading data)

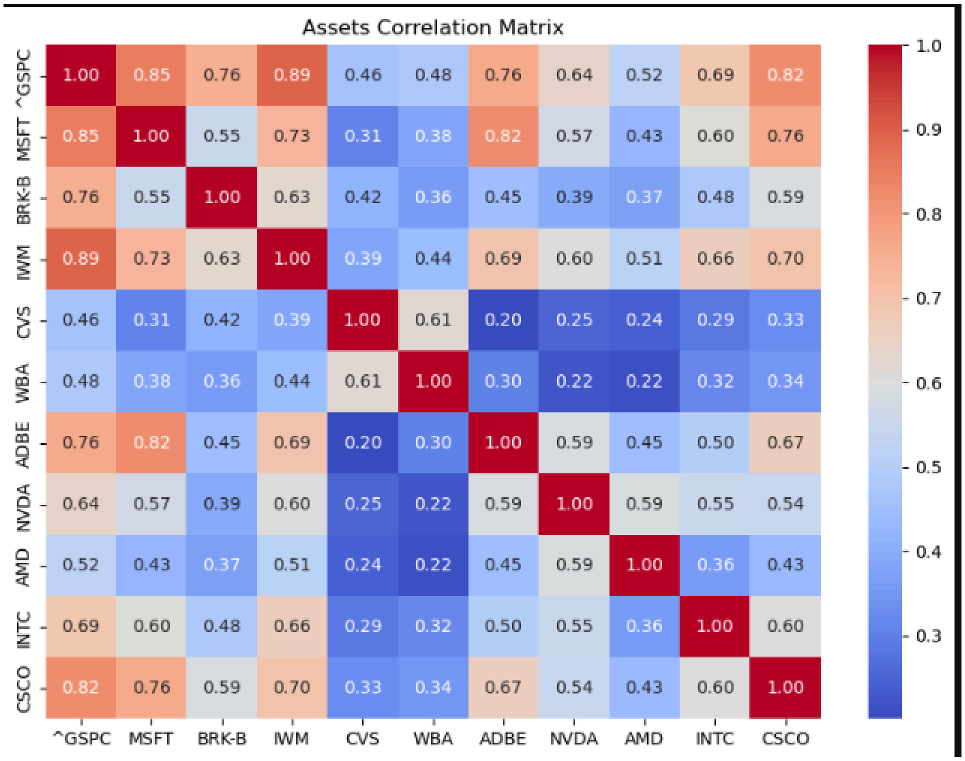
This approach facilitated the collection of historical stock price data for subsequent analysis, enabling insights into the performance of the selected stocks and indices over the specified time period.

Preprocessing:

During the initial phase of data retrieval and DataFrame creation, we observed that the downloaded data included timestamps, which were unnecessary for our analysis. As a solution, we developed a function to extract only the date component from the timestamp, effectively removing the time information. This function was applied to the index of the DataFrame, ensuring that our dataset consisted solely of date information, which was more relevant for our analysis purposes.

Pairs Selection

Correlation Matrix:



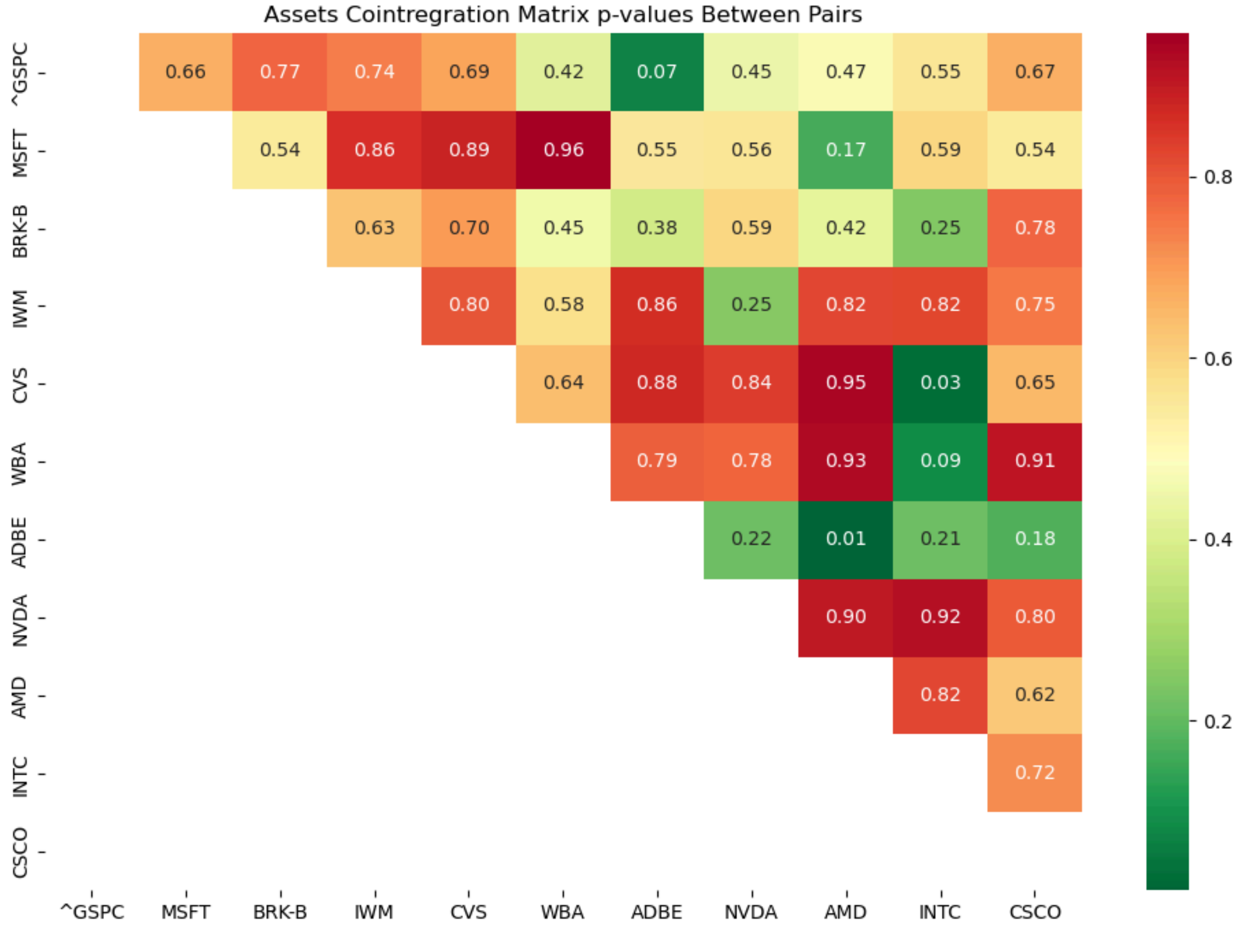
	^GSPC	MSFT	BRK-B	IWM	CVS	WBA	ADBE	NVDA	AMD	INTC	CSCO
^GSPC	1.00	0.85	0.76	0.89	0.46	0.48	0.76	0.64	0.52	0.69	0.82
MSFT	0.85	1.00	0.55	0.73	0.31	0.38	0.82	0.57	0.43	0.60	0.76
BRK-B	0.76	0.55	1.00	0.63	0.42	0.36	0.45	0.39	0.37	0.48	0.59
IWM	0.89	0.73	0.63	1.00	0.39	0.44	0.69	0.60	0.51	0.66	0.70
CVS	0.46	0.31	0.42	0.39	1.00	0.61	0.20	0.25	0.24	0.29	0.33
WBA	0.48	0.38	0.36	0.44	0.61	1.00	0.30	0.22	0.22	0.32	0.34
ADBE	0.76	0.82	0.45	0.69	0.20	0.30	1.00	0.59	0.45	0.50	0.67
NVDA	0.64	0.57	0.39	0.60	0.25	0.22	0.59	1.00	0.59	0.55	0.54
AMD	0.52	0.43	0.37	0.51	0.24	0.22	0.45	0.59	1.00	0.36	0.43
INTC	0.69	0.60	0.48	0.66	0.29	0.32	0.50	0.55	0.36	1.00	0.60
CSCO	0.82	0.76	0.59	0.70	0.33	0.34	0.67	0.54	0.43	0.60	1.00

(Figure 2: the heatmap of correlation)

We utilized a heatmap visualization technique to analyze the correlation between every asset in our dataset, as depicted in Figure 2. While this provided insights into the volatility relationships among different pairs, it alone was insufficient for identifying suitable pairs for spread-based trading strategies.

To further refine our selection process, we implemented additional tests, including cointegration and stationarity tests. These tests allowed us to evaluate the long-term relationship and stability of pairs, respectively. By analyzing the p-values obtained from these tests, we were able to identify potential assets that exhibited favorable characteristics for pairs trading strategies.

Cointegration Test:



(Figure 3: assets cointegration matrix)

We conducted a cointegration test between each asset pair and visualized the resulting p-values on a matrix, represented in Figure 3. This matrix allowed us to assess the potential suitability of asset pairs for trading.

Utilizing the heatmap visualization, we identified pairs with p-values below the threshold of 0.05, indicating significant cointegration. Based on our analysis, we identified the pairs [ADBE, AMD] and [CVS, INTC] as promising candidates for pairs trading strategies. These pairs exhibited strong potential for establishing stable and profitable trading relationships, making them favorable choices for further analysis and strategy development.

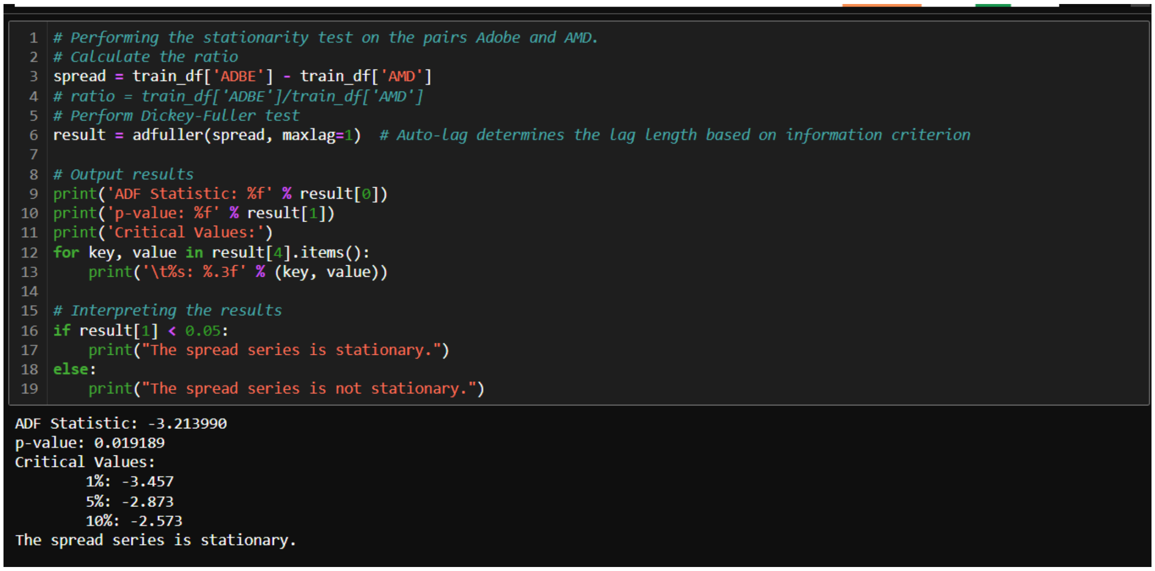
Augmented Dickey-Fuller Test:

The Augmented Dickey-Fuller (ADF) test is a statistical test used to determine whether a unit root is present in a time series dataset. A unit root indicates that a time series is non-stationary, meaning its mean and variance change over time. The ADF test is commonly employed in econometrics and finance to assess the stationarity of a time series, which is a crucial assumption in many statistical models.

The test is based on the Dickey-Fuller test, which examines the presence of a unit root in a first-differenced series. The "augmented" version of the test extends this by including lags of the differenced series as additional explanatory variables. This helps account for autocorrelation and improves the test's power.

The null hypothesis of the ADF test is that a unit root is present in the time series, indicating non-stationarity. Therefore, a p-value below a chosen significance level (e.g., 0.05) would lead to rejecting the null hypothesis, suggesting that the time series is stationary. Conversely, a p-value above the significance level would fail to reject the null hypothesis, indicating non-stationarity.

We further implemented this test to see if the pairs were mean reverting or not (Figure 4).



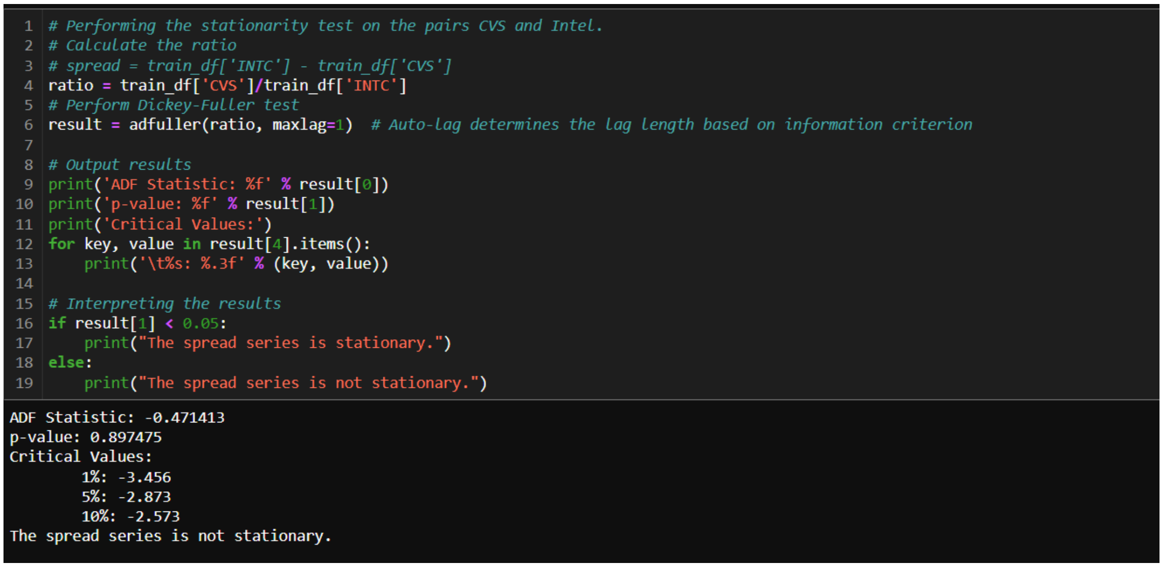
```python
# Performing the stationarity test on the pairs Adobe and AMD.
# Calculate the ratio
spread = train_df['ADBE'] - train_df['AMD']
# ratio = train_df['ADBE']/train_df['AMD']
# Perform Dickey-Fuller test
result = adfuller(spread, maxlag=1)  # Auto-lag determines the lag length based on information criterion

# Output results
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

# Interpreting the results
if result[1] < 0.05:
    print("The spread series is stationary.")
else:
    print("The spread series is not stationary.")
```

```
ADF Statistic: -3.213990
p-value: 0.019189
Critical Values:
	1%: -3.457
	5%: -2.873
	10%: -2.573
The spread series is stationary.
```

(Figure 4: the code of ADF test of ADBE and AMD)

We evaluated two trading methods: one based on the price ratio and another based on the spread. The price ratio method did not provide sufficient evidence to reject the null hypothesis. Consequently, we opted to focus on the spread between Adobe and AMD as our primary pair for pairs trading.

Additionally, we identified CVS Health and Intel as potential candidates for pairs trading based on earlier analysis. To further validate this pair, we conducted the Augmented Dickey-Fuller (ADF) test, commonly used to assess stationarity in time series data.



```python
# Performing the stationarity test on the pairs CVS and Intel.
# Calculate the ratio
# spread = train_df['INTC'] - train_df['CVS']
ratio = train_df['CVS']/train_df['INTC']
# Perform Dickey-Fuller test
result = adfuller(ratio, maxlag=1)  # Auto-lag determines the lag length based on information criterion

# Output results
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

# Interpreting the results
if result[1] < 0.05:
    print("The spread series is stationary.")
else:
    print("The spread series is not stationary.")
```

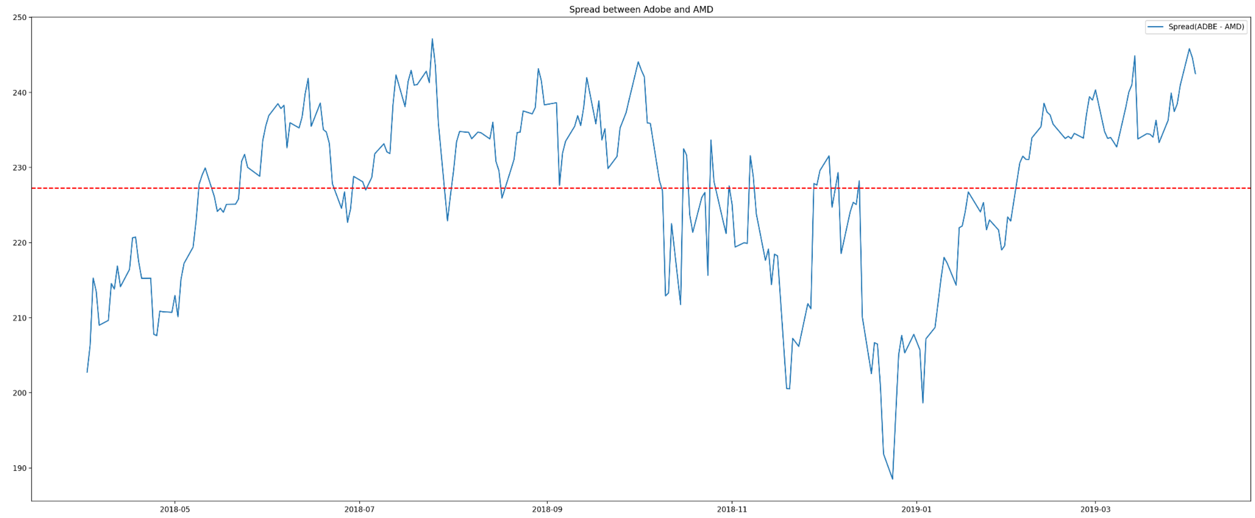
```
ADF Statistic: -0.471413
p-value: 0.897475
Critical Values:
	1%: -3.456
	5%: -2.873
	10%: -2.573
The spread series is not stationary.
```

(Figure 5: the code of ADF test of CVS and INTC)

We employed both the price ratio and the spread between the two asset stock prices as potential trading indicators. However, the stationarity test results indicated that neither method met the stationarity criteria. Consequently, we proceeded with the pair of Adobe and AMD for our analysis, as it showed more favorable characteristics for pairs trading.

After Selection of pairs:

To further visualize and see if our pairs are mean reverting, we plotted the following graph of the spread (Figure 6):



(Figure 6: the graph of the spread)

The observed spread between Adobe and AMD presents promising potential for generating favorable returns, provided that our trading strategy is appropriately structured and executed.

Trading Strategy

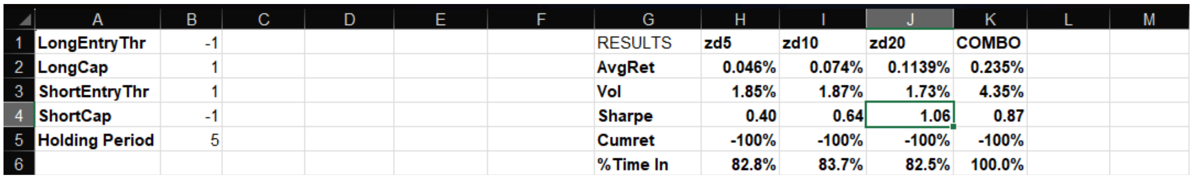
Our trading strategy incorporated the following components:

- We calculated the z-scores for each asset based on the previous 20-day returns, taking into account a trailing volatility period of 60 days.
- The spread between the two assets was determined as the difference between their respective z-scores.
- We established long and short thresholds at 1 and -1, respectively. If the spread crossed these thresholds, it signaled an opportunity to enter or exit a position.
- To manage risk, we implemented long and short caps at 1 and -1, respectively.
- We maintained a holding period of 5 days for our trades.

These parameters were carefully considered and integrated into our analysis to develop a robust and systematic trading strategy.

Analysis and Results

Baseline Analysis:



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	LongEntryThr	-1					RESULTS	zd5	zd10	zd20	COMBO		
2	LongCap	1					AvgRet	0.046%	0.074%	0.1139%	0.235%		
3	ShortEntryThr	1					Vol	1.85%	1.87%	1.73%	4.35%		
4	ShortCap	-1					Sharpe	0.40	0.64	1.06	0.87		
5	Holding Period	5					Cumret	-100%	-100%	-100%	-100%		
6							%Time In	82.8%	83.7%	82.5%	100.0%		

(Figure 7: the outcome of baseline analysis)

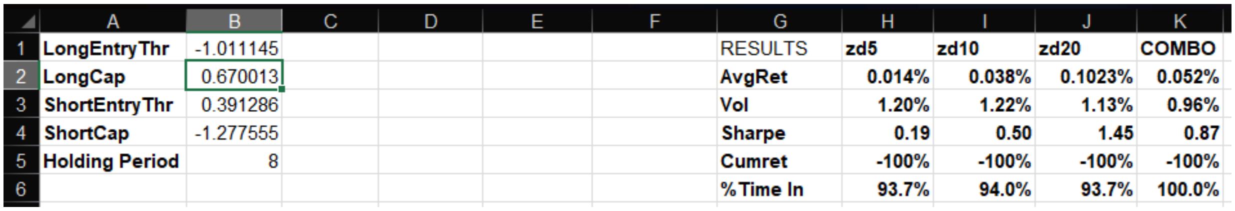
After implementing the parameter values initially set, we achieved a Sharpe ratio of 1.06 for the spread between Adobe and AMD, as illustrated in Figure 7. This served as our baseline for further optimization of the parameters.

For the optimization process, we focused on refining the zd20 strategy using the solver in Excel. The constraints applied during the optimization were as follows:

- Long Entry Threshold: Ranging from -2 to 2
- Long Cap: Ranging from -2 to 2
- Short Entry Threshold: Ranging from -2 to 2
- Short Cap: Ranging from -2 to 2
- Holding period: Ranging from 1 to 10 days

By fine-tuning these parameters within the specified constraints, we aimed to enhance the effectiveness and performance of the zd20 strategy for pairs trading.

Result:



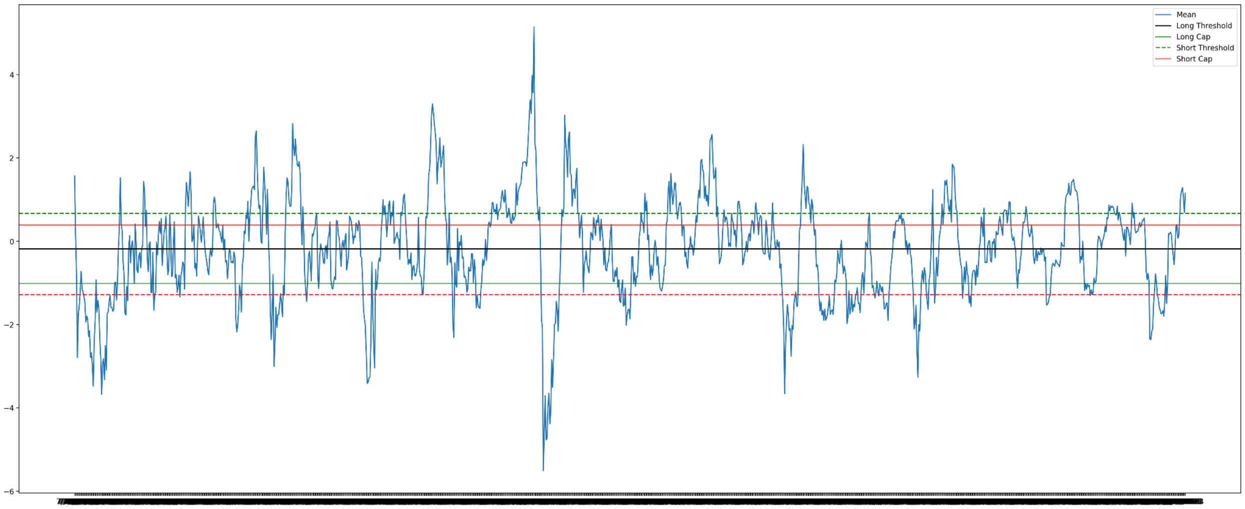
	A	B	C	D	E	F	G	H	I	J	K
1	LongEntryThr	-1.011145					RESULTS	zd5	zd10	zd20	COMBO
2	LongCap	0.670013					AvgRet	0.014%	0.038%	0.1023%	0.052%
3	ShortEntryThr	0.391286					Vol	1.20%	1.22%	1.13%	0.96%
4	ShortCap	-1.277555					Sharpe	0.19	0.50	1.45	0.87
5	Holding Period	8					Cumret	-100%	-100%	-100%	-100%
6							%Time In	93.7%	94.0%	93.7%	100.0%

(Figure 8: the outcome of parameter optimization)

Using the solver, we got the following optimum parameters that maximize the sharpe ratio to 1.45 (Figure 8).

This was a significant improvement from our baseline, and hence we decided to proceed with this strategy.

Visualization:



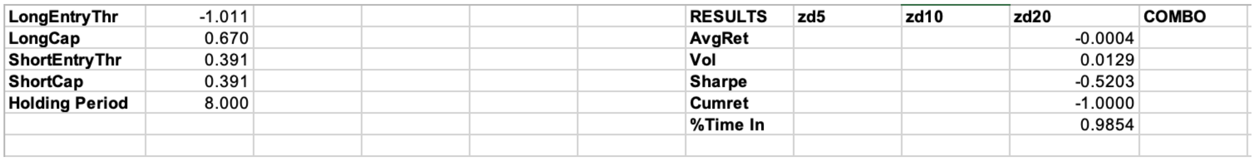
(Figure 9: the visualization graph)

The visualization clearly depicts the set thresholds for our strategy (Figure 9). When the zdiff20 value surpasses a specific threshold, we initiate a long or short position on the spread, depending on which threshold it crosses. Additionally, we only execute trades on the spread when the value falls within the range defined by the long threshold and long cap, or the short threshold and short cap.

Analysis of one-year withheld data:

As mentioned before, we only used the data from April 1st 2018 to April 21st 2023. We still had more than a year of untouched data.
This untouched data served as our deployment stage, where we would test our strategy in the real world.

We observed the following results:



LongEntryThr	-1.011					**RESULTS**	**zd5**	**zd10**	**zd20**	**COMBO**
LongCap	0.670					**AvgRet**			-0.0004	
ShortEntryThr	0.391					**Vol**			0.0129	
ShortCap	0.391					**Sharpe**			-0.5203	
Holding Period	8.000					**Cumret**			-1.0000	
						%Time In			0.9854	

(Figure 10: the outcome in unseen data)

Our analysis of the untouched data highlights the complexity inherent in predicting future market movements solely based on long and short thresholds and caps. Despite our initial optimism, the application of our strategy to previously unseen data resulted in a negative Sharpe ratio. This outcome underscores the challenges of extrapolating from historical data to make accurate predictions about future market behavior.

This experience underscores the importance of incorporating additional indicators and refining our strategy to enhance its predictive power and resilience. Moving forward, we recognize the need to explore and integrate more sophisticated indicators and risk management techniques into our trading strategy. By doing so, we aim to develop a more robust and reliable approach capable of navigating the complexities and uncertainties of the financial markets.

Conclusion

Our initial statistical analysis provided compelling evidence for the suitability of Adobe and AMD as assets for spread-based trading. However, translating our strategy from theory to practice revealed the challenges of real-world application in a dynamic market environment.

While our initial analysis using the chosen strategy yielded satisfactory results, the real-world testing exposed limitations and underscored the need for a more comprehensive approach. By integrating additional factors such as market indicators and implementing robust risk management strategies, we believe that the spread between Adobe and AMD holds significant potential for generating substantial returns.

Despite the initial setbacks, we remain optimistic about the prospects of refining our strategy to capitalize on the opportunities presented by the Adobe and AMD spread. This underscores the iterative nature of trading strategy development and the importance of adaptability in navigating the complexities of financial markets.

Future Development

The current phase of our project remains primarily theoretical and not yet suitable for practical implementation. However, this exploration has deepened our understanding of pair trading strategies while also highlighting numerous areas for optimization and refinement.

To transition our theoretical framework into a practical strategy, several key elements must be addressed:

- Initial Capital Investment: Establishing a specific initial investment amount is essential for accurate profit and loss calculations and ensures clarity in strategy implementation.

- Additional Indicators: While our focus on the spread of normalized returns over N days is foundational, incorporating supplementary indicators is necessary to enhance the strategy's robustness and effectiveness.
- Risk Management: Developing comprehensive risk management protocols, such as implementing stop-loss and profit target mechanisms, is critical to safeguard against potential losses and maintain the strategy's sustainability.

By integrating these components, we aim to evolve our theoretical framework into a practical and effective trading strategy capable of navigating the complexities of real-world financial markets.

Links and References

Link to project:

- GitHub: https://github.com/adi-suresh01/Pairs_Trading_Strategy_1

References:

- https://www.youtube.com/watch?v=q_HS4s1L8UI
- https://www.youtube.com/watch?v=f73ItMWO4z8
- https://youtu.be/JTucMRYMOyY?si=lgPf64U0xu5bajnT
- https://youtu.be/fqltiq5EahU?si=nUDrTlxQEykKgMfo