QFin Trading Team Project

[$] Market Breakers [$]

May 16, 2023

By Sithum Somarathna, William Ho, Arlene Janse van Rensburg, Arshaq Siraz

Contents

1 Introduction

The 2023 QFin Trading Project required teams to create a trading strategy and algorithm that could trade a combination of 18 stocks and ETFS, including AAPL, AVGO, CRM, CVS, GOOG, GOOGL, HCA, HUM, IHF, IYW, MA, MSFT, NVDA, SPY, TXN, UNH, V, VGT. Each team could access the historical performance data of these securities from 19 April 2021 to 6 January 2023. Our final trading algorithm will be tested over the 90 day period after the historical data ended. A trading fee of 0.2% and delta limits of 1000 will be applied.

In this report, we will investigate the nature of the data provided and the relevant statistical measures that were required to identify the optimal securities to trade. We will also discuss the method we used to select our pairs and then explain how we designed our strategy and determined the spread and optimal parameters. Finally, we will discuss backtesting and how we analysed the performance of our model.

2 Background

A time series is a sequence of historical prices of a financial instrument, such as a stock or a currency pair. By analysing the patterns and trends in the time series data, traders can predict future price movements based on past data and make informed decisions on when to buy or sell an instrument.

Correlation

Correlation is used in time series data as a statistical measure of the relationship between two securities and to identify pairs of securities that move together.

The correlation between two securities is not static and may vary over longer periods of time. There may also be a temporary relationship that disappears after a short period.

The main risk associated with using correlation as a determinant of a relationship is that many actions in the markets arise from noise and randomness, which means that a seemingly statistically significant correlation could be due to chance. Therefore, we conducted additional analysis to confirm the relationships between securities.

Cointegration

Cointegration describes the long-term relationship between prices. Cointegrated asset pairs will likely have a predictable relationship over time even if the relationship diverges in the short term.

ETF and stock prices are non-stationary time series and exhibit non-random patterns over time that change the statistical properties of the data, making it difficult to analyse. However, the prices of two cointegrated assets have a constant spread over time, which is a stationary time series with stable statistical properties which are easier to model, forecast future values with reasonable accuracy and identify opportunities to make a profit from short-term fluctuations.

Cointegration analysis allowed us to create a pairs trading strategy using mean reversion which sells the overvalued asset in our cointegrated pair, and buys the undervalued asset, with the expectation that the spread would converge back to its mean value. Since the spread is stationary, we can be reasonably confident that the prices will eventually revert to their long-term relationship, making the strategy potentially profitable.

3 Pair Selection

3.1 Nature of the Data

Our data set included 4 ETFs; IHF, SPY, IYW and VGT, and 14 underlying stocks. The majority of these are in the technology sector, and over our historical data period the NASDAQ 100 Technology Sector Index (NDXT) fell over 30% in comparison to the S&P500 which fell by 75%. Technology stocks significantly under performed the market during this period, largely due to the residual impacts of the COVID-19 pandemic.

We considered the unusually high volatility and large price movements during our historical data period when we determined the open and close threshold of our trading strategy, which is further discussed later in this report.

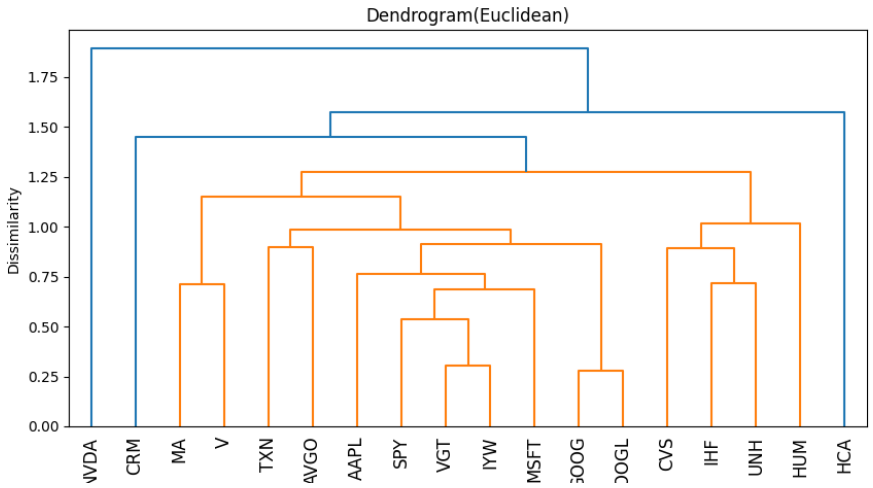
3.2 Selecting a Pair

To select the pairs of securities we would trade, we combined hierarchical clustering, correlation and cointegration statistical analysis techniques. We briefly considered using copulas to determine the dependence between securities, but due to our limited experience in this area it was beyond the scope of our project.

Hierarchical clustering allowed us to identify the pairs of securities that were most closely linked. The Linkage function from the SciPy library calculated the Euclidean distance between the normalised price of securities and clustered the securities based on their similarity.

As illustrated in the dendrogram in Figure 1, it was clear that GOOG/GOOGL and VGT/IYW were the two pairs with the closest Euclidean distance.

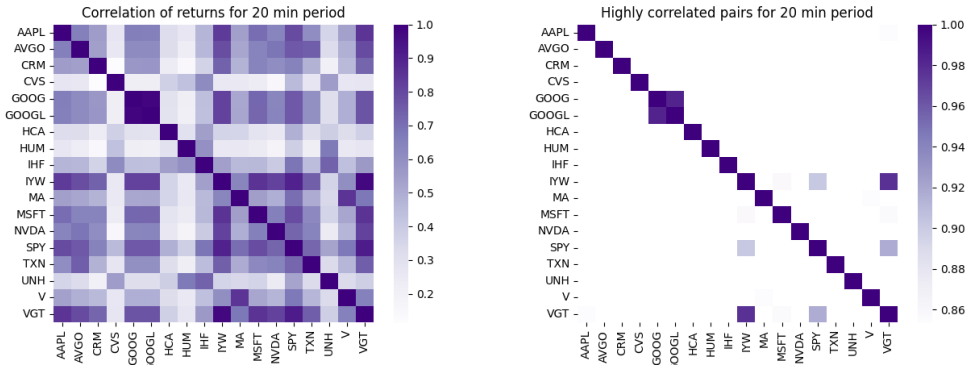
Figure 1: Euclidean Distance Between Tickers



To calculate correlation, we used the Pandas correlation function on the percentage returns across a 20 minute window for each security. As illustrated below in Figure 2, we identified two pairs with a correlation coefficient above 0.85: GOOG/GOOGL and VGT/IYW. This further confirms our findings from the Euclidean distance analysis.

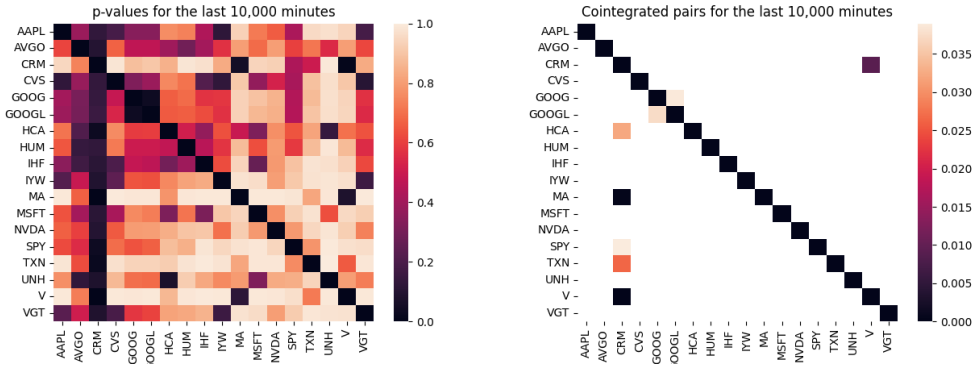
We used the cointegration function in the StatsModels library to test for cointegration in the dataset across a period of the last 10,000 minutes, shown in Figure 3. We removed some of the noise from our data, and kept only the results with a p value of less than 0.5 in the figure to the right.

Figure 2: Correlation Across the Data Set



According to these results, GOOG/GOOGL were cointegrated, but VGT/IYW were not. However, when we conducted a cointegration test between IYW and VGT over the entire data set, the cointegration p value was 0.17. Although this is not ideal, the cointegration p value of 0.17 was not high enough to disqualify the previous correlation and hierarchical clustering evidence of a relationship between VGT and IYW. Additionally, the reliability of the 10,000 minute cointegration tests is uncertain due to repeated runs of the test producing different results across different platforms and computers. Therefore, we continued developing our strategy using these two securities for pairs trading.

Figure 3: Cointegration Across the last 10,000 minutes of the Data Set



4 Strategy Design

4.1 Pairs Trading and Mean Reversion

As mentioned earlier, we aimed to use pairs trading on GOOG/GOOGL and VGT/IYW as our underlying strategy. Fundamentally, this strategy is based on the assumption that two securities that have a historically stable and predictable relationship will return to a mean spread (mean reversion). This creates arbitrage opportunities when the spread deviates significantly from its mean, allowing us to take opposing positions on members of the pair with the expectation that the prices will eventually converge back to their long-term relationship restoring the mean spread.

The idea behind a mean reversion trading strategy is that stocks tend to fluctuate around their long-term average prices, and any significant deviation from this average is likely to be temporary and provide opportunities for profitable trades.

When creating a mean reversion strategy, the main challenge is identifying suitable stock pairs that exhibit mean-reverting behaviour, defining robust trading rules that can capture these opportunities and minimizing risks. To do this, it's important to analyse historical price data, consider statistical indicators such as moving averages and standard deviations and account for macroeconomic factors that may influence the stock's price.

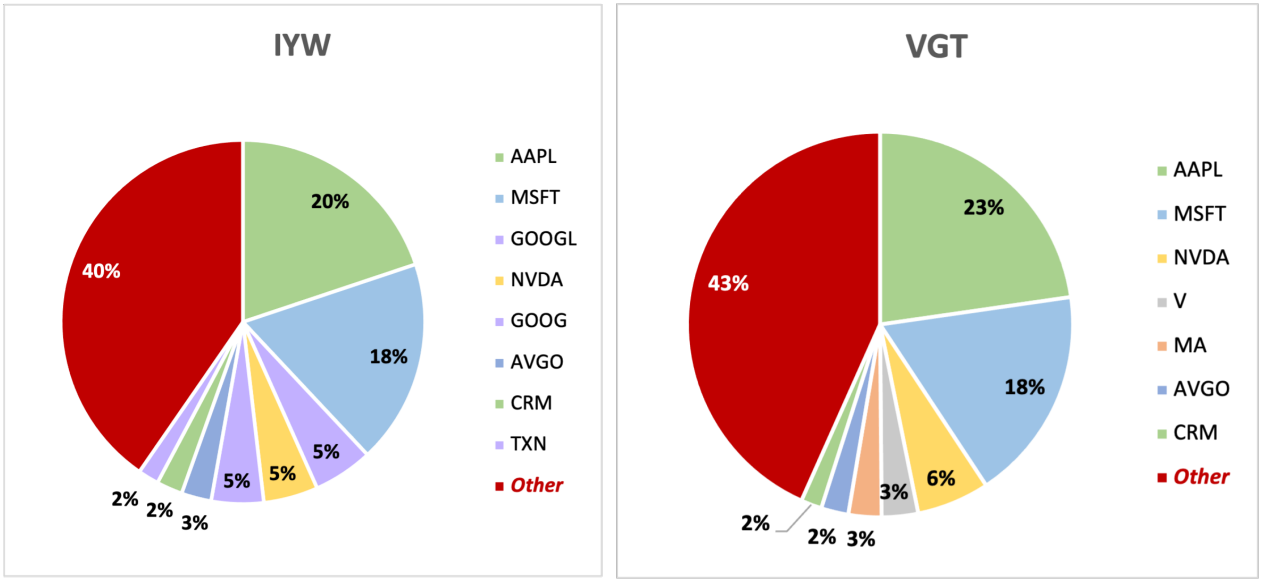
4.2 Consideration of ETF Arbitrage

Another trading strategy that we considered was ETF arbitrage, which involves taking advantage of temporary price inconsistencies between an ETF and its underlying components. Since we only had access to limited data, for this strategy to work we would have needed to firstly estimate a fair value for each ETF based on the performance of the underlying components that we had access to, and then make trades based on small discrepancies between the current ETF market price and our predicted fair value.

4.2.1 Data Limitations

However, the nature of our data made it challenging to accurately estimate the fair value of each ETF.

Figure 4: Composition from Underlying Stocks



As illustrated in the figures above, 40% and 43% of the IYW and VGT ETFs were unknown respectively. We attempted to replicate the ETFs using linear scaling, power scaling and exponential scaling of the known components to proportionally estimate the unknown components. However, for each of the replicated ETFs, our cointegration tests returned a p value of 0.8 or higher compared to the original ETFs. This showed that it was not reasonable to use any of the replicated ETFs to estimate the fair value of the original ETF with the degree of confidence required to make trading decisions for an ETF arbitrage strategy.

The risk of trading losses and missed opportunities due to incomplete data were too great, therefore we decided to solely focus on mean reversion pairs trading for our strategy instead.

4.3 Calculating Spread

When pairs trading, there are a variety of ways to define the price differences between two securities. Here are three methods that we considered:

1. Taking the absolute difference
   A naive and easy to understand approach would be to find the absolute difference between the two prices by simply subtracting the higher security from the lower security. This method provides a very intuitive metric on how the two securities compare to each other and is less sensitive to noise.

2. Taking the price ratio
   Another method is to take the price ratio by dividing the price of the higher security by the price of the lower security. This method is more sensitive and can be used to identify trading opportunities more frequently.

3. Taking the logarithmic difference
   A more complex method is to take the natural log of each security's prices and subtract them from each other (equivalently finding the natural log of their ratio). This method provides a more normalised metric of the price differences.

In order to decide which method to use, we computed the spread of VGT/IYW and GOOG/GOOGL at each timestamp for each method, and normalised this value using the distribution of spreads from the previous X days. This was done for the entire data set for X=1, X=10 and X=15. The normalised spreads were then graphed to visualise their distributions over their mean (shown on the next page).

The figures show that the spread distribution between GOOG and GOOGL closely resemble a standard normal distribution across all three methods of calculating spreads, with the choice of window size having minimal effect. This was expected because the GOOG/GOOGL pair was the most correlated. However, as VGT and IYW were not as correlated, their spread distributions showed a two-peak distribution. Although this two-peak distribution was not ideal, the ratio and logarithmic difference methods seemed to reduce their significance and approximate a single peak, especially with smaller window sizes. As the securities in each pair have not experienced any long-term growth or shrinkage within the length of the data set, the ratio and logarithmic difference methods both produced almost identical results. Hence, we decided to use the ratio method for simplicity.

Figure 5: Normalised Spreads of GOOG and GOOGL Across a 15 (top), 5 (middle) and 1 (bottom) Day Rolling Window

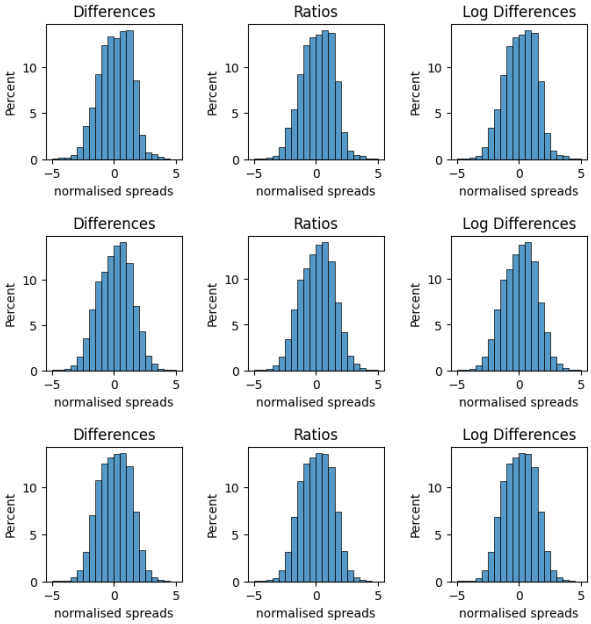
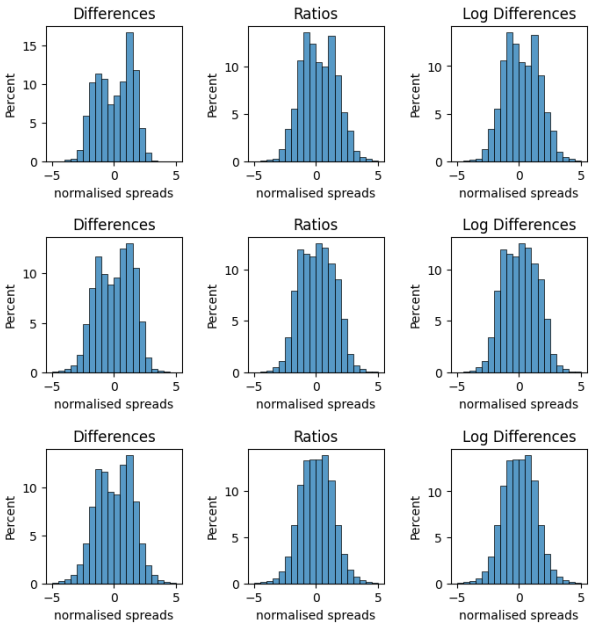


Figure 6: Normalised Spreads of VGT and IYW Across a 15 (top), 5 (middle) and 1 (bottom) Day Rolling Window



4.4 Outline of Parameters and Chosen Strategy

Having decided on our method of calculating spread, we were able to define a strategy that utilised the rolling normalised value of spread at each timestamp and performed trades based on this value (the number of standard deviations the value was away from the window's mean).

4.4.1 Rolling Window Length

In order to determine a metric for the deviation of spreads from their mean at each timestamp, we needed to compare them to spread values at previous timestamps. Since the market is constantly changing, we only wanted to use recent values to generate a distribution of spreads to measure how far our current spread deviated from this distribution. Therefore, we needed to use a rolling window that only looked at the past W days of timestamps. It should be noted that too large of a W value will contain outdated spread values while too small of a W value will be susceptible to noise.

4.4.2 Open Threshold

Next, we needed to define an entry point for trades. An entry point would be a standard deviation threshold σ_{open} chosen such that when the spread at the current timestamp was greater than σ_{open} standard deviations away from its mean, the corresponding long and short trades would be made. We needed to ensure that our value of σ_{open} was appropriately balanced, such that it was not too small and triggered by random noise and not so large that it overlooked valuable trading opportunities.

4.4.3 Close Threshold

We also needed to define when we would close our positions to lock in any profits made. This would also be a standard deviation threshold σ_{close} ($0 \leq \sigma_{close} \leq \sigma_{open}$) chosen such that when the spread of the current timestamp was within σ_{close} standard deviations from the mean, we would attempt to close any open positions we held. As the market fluctuated, the mean of the spreads was constantly changing. Therefore, we ensured that σ_{close} was not too low because this could have prolonged our exposure to the market, meaning that a significant shift in the distribution between our entry and exit times would have become more likely and could have caused major losses. Conversely, we also needed to ensure that σ_{close} was not too close to σ_{open} because this could have made our profits per trade very small and potentially insufficient to cover the trading fees.

4.4.4 Max Threshold

As the market is dependent on events that happen in the real world, it is possible that some event could cause the prices for both securities to rapidly grow or fall during a short period of time. Each security would then establish a new equilibrium rather than return to the previous equilibrium. If we traded during a period of rapid change, the movement of the securities would directly conflict with our assumption of mean reversion, which could cause major losses if we entered a position during that period. Therefore, we adjusted our standard deviation threshold σ_{max} ($\sigma_{max} >> \sigma_{open}$) to try and avoid trading during these types of events. If the standard deviation of the current timestamp surpassed σ_{max} standard deviations away from the mean, we assumed that there was a possibility of a major price shift and halted our trading until the new equilibrium is established.

4.4.5 Trading Quantity

When the opportunity arose for our strategy to make a trade ($\sigma_{open} \leq \sigma \leq \sigma_{max}$), we needed to define the quantity (Q) of each security we wished to trade. If Q is too low, we may miss out on profitable

opportunities, while if Q is too high, we may overextend our position and be less resilient against incorrect trading decisions made by our strategy.

4.4.6 Signal Boosting

As an extension, we also considered how our quantity parameter (Q) should change based on the current σ value. As the current σ moved further away from the mean, the expected profitability of trading at that time increased because there would be a larger fall/rise in overall prices once the price reverted back to the mean value. Ideally, we wanted our Q value to increase as σ increased, so we rendered Q a function of the current standard deviation, $Q(\sigma)$.

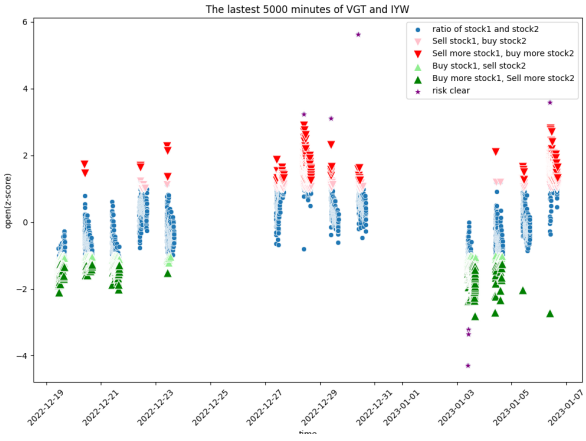
4.4.7 Strategy

Using the parameters above, our strategy at each timestamp was as follows:

1. Calculate the normalised value of spread at the current timestamp using the spreads of the most recent W timestamps as the distribution
2. If $\sigma_{open} \leq \sigma \leq \sigma_{max}$ then sell the higher security and buy the lower security with a quantity based off of $Q(\sigma)$
3. If $-\sigma_{open} \geq -\sigma \geq -\sigma_{max}$ then sell the lower security and buy the higher security with a quantity based off of $Q(\sigma)$
4. If $-\sigma_{close} \leq \sigma \leq \sigma_{close}$ then close our position
5. In all other cases, no trade takes place

4.4.8 Example of Strategy

Figure 7: Example of Strategy over last 5000 Timestamps



The diagram above shows an example of how we may run our strategy. The data points represent the normalised spread of VGT and IYW over the last 5000 timestamps of our sample data. In this example $\sigma_{open} = 1$, $\sigma_{close} = 0.8$, $\sigma_{max} = 3$. When the standard deviation of the current spread is between 1 and 3 standard deviations away from the mean, we perform trades. If it is within 0.8 standard deviations, we close our position and if it is more than 3 standard deviations away from the mean, we stay idle.

5 Backtesting and Performance of Model

Once our strategy was defined, the next step was to determine the values for our parameters that would yield the maximum profit. To achieve this, we defined a range of suitable values for each parameter and ran the backtester on all permutations of these parameter sets. As this was quite a computationally expensive operation, we limited the number of runs done on each combination of parameters to three 90-day runs and took the average profit to evaluate performance. Afterwards, we took the five highest performing combinations and tested them on twenty 90-day runs.

While our strategy produced profitable results for pairs trading done on VGT/IYW, it was unable to produce a profitable set of parameters for GOOG/GOOGL. We suspected that was due to our strategy not being effective on pairs of securities with extremely high cointegration because any profitable variation in their spread may only occur during times we deemed too uncertain to trade ($\sigma_{open} > \sigma_{max}$). However, further investigation is needed to be certain.

Therefore, our final strategy only performed pairs trading on VGT/IYW and the findings below reflect only this pair.

5.1 Parameter Optimisation

Below is the set of parameter values we found to yield the most profit:

W	σ_{open}	σ_{close}	σ_{max}	$Q(\sigma)$
15 days	1	0	1.5	$5H(\sigma - 1.25) + 5$

Where $H()$ is the Heaviside step function (explained later).

5.1.1 Signal Boosting Function

Our final signal boosting function was $5H(\sigma - 1.25) + 5$ which made use of the Heaviside step function, a function that returns 0 when the input is less than 0 and returns 1 when the input is greater than 0. This means that our strategy will trade a quantity of 5 when the standard deviation is less that 1.25, and trade a quantity of 10 when the standard deviation is greater than 1.25. We could have used infinitely many functions, but decided to keep it simple and chose a step function so that we could focus on more important areas of the parameter search.

5.1.2 Analysis of σ_{close}

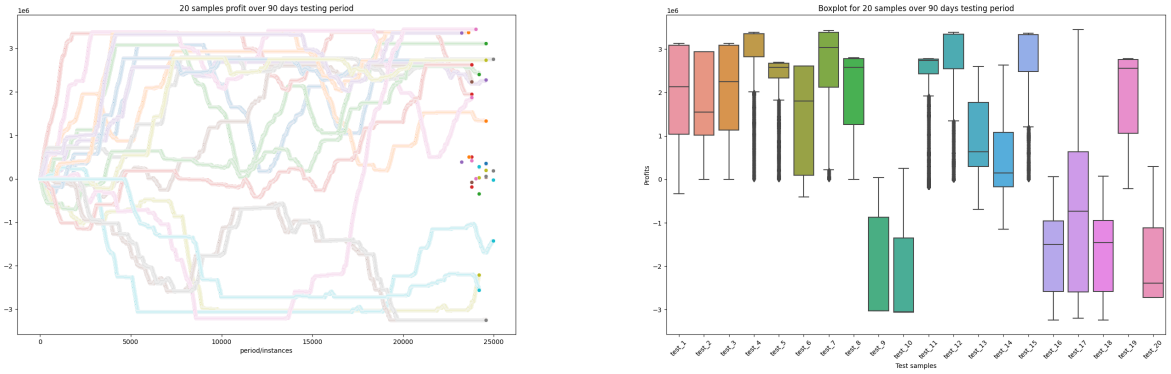
To our surprise, the optimal value for our closing threshold was 0. As this is the absolute value of the current standard deviation (the distance from the mean) that is compared to all our thresholds, a threshold of 0 would never be flagged. This means that our model ignores the closing threshold completely. Therefore, our model only relies on the spread fluctuations to alternate between exceeding the open threshold on either side of the mean so that we maintain an approximately neutral position in the long term.

5.2 Profit and Loss

Profit and loss was the key metric used to measure the effectiveness of our mean reversion strategy. The profits and losses demonstrated in our results indicated whether our strategy was able to correctly identify significant deviations and react appropriately.

These figures below show the 20 samples' profits over the 90 day period for the VGT and IYW pairs trading strategy. As illustrated on the figure on the right, the standard deviation between the 20 samples' profits varied significantly. The figure on the left shows the ongoing PnL of each run and

Figure 8: PnL Results

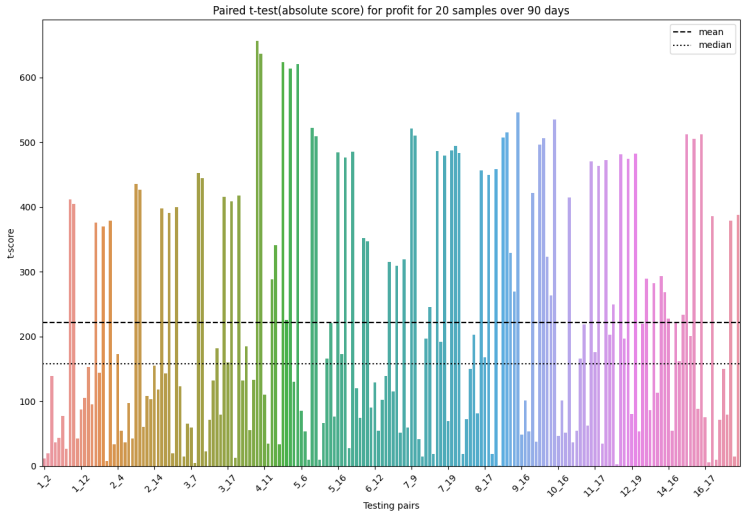


the end dots on the right hand side represent the final accumulated profits for each of the 20 samples after our model closed the positions. After executing our strategy for fifty 90-day runs, our average profit was $\$163,514.93$ with a standard deviation of $\$205,554.95$.

5.3 t-statistics

To further measure the model's consistency, we decided to use a t-test to measure how consistently our model performs. Unsurprisingly, as seen in the figure below, the t-scores in general were very large, which re-affirms that our model's performance was very unstable (which is evident in the 2 figures above). However, a t-test requires the pairs to be independent, but some of testing sample periods overlapped, which means that there is additional analysis needed to identify further ways to improve our model.

Figure 9: T-Statistics of results



6 Future Investigation and Improvements

A key opportunity to improve our strategy would be reducing the variability of our results, in other words minimising the standard deviation of our profits and losses. If we accomplish this, we could achieve greater stability in the PnL outcomes across various trials, and could potentially eliminate negative returns altogether.

In order to minimise this variability, we could explore sophisticated trading strategies that incorporate dependable loss indicators. These indicators would enable us to adapt our strategies in a dynamic way when losses are detected.

Another area for improvement would be to better utilise the additional stock data we had access to, to develop more trading strategies that could be profitable in different market conditions. For example, we could perform further analysis and identify a way to make the ETF arbitrage strategy more viable by looking into new ways to use the limited underlying stock data to extrapolate a reliable ETF fair value estimate.