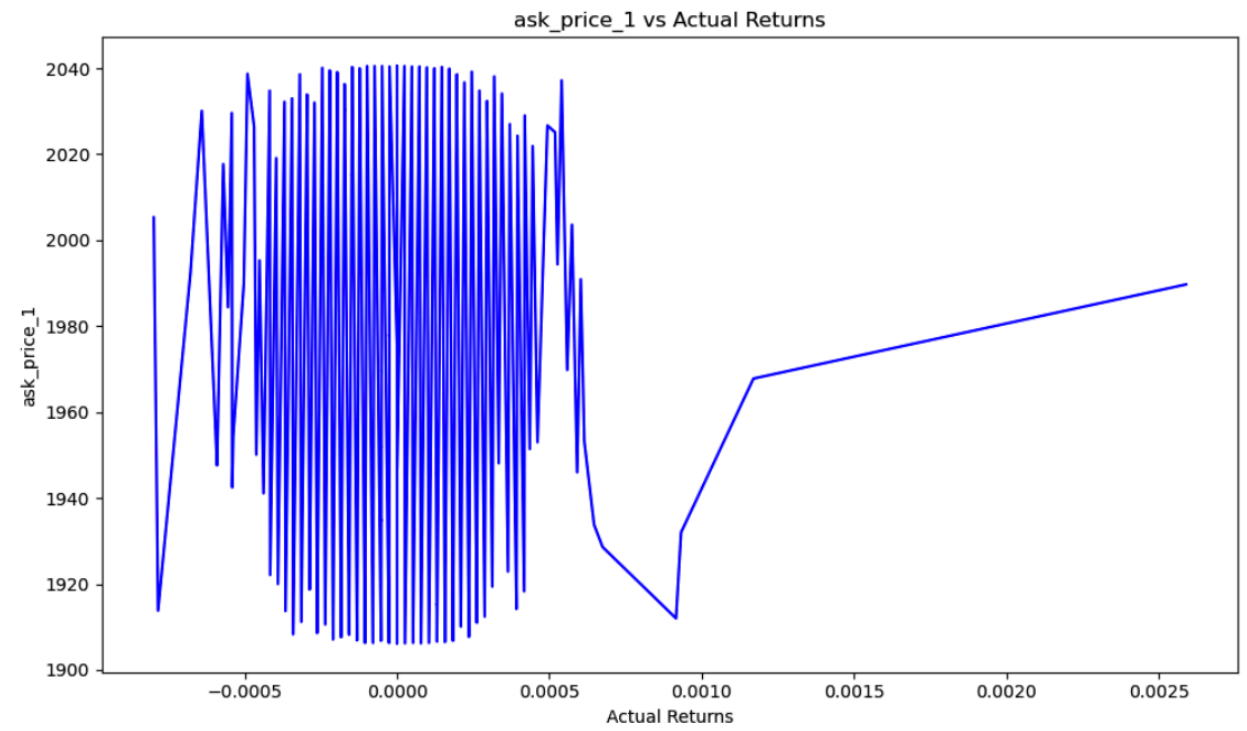
PROBLEM STATEMENT 2

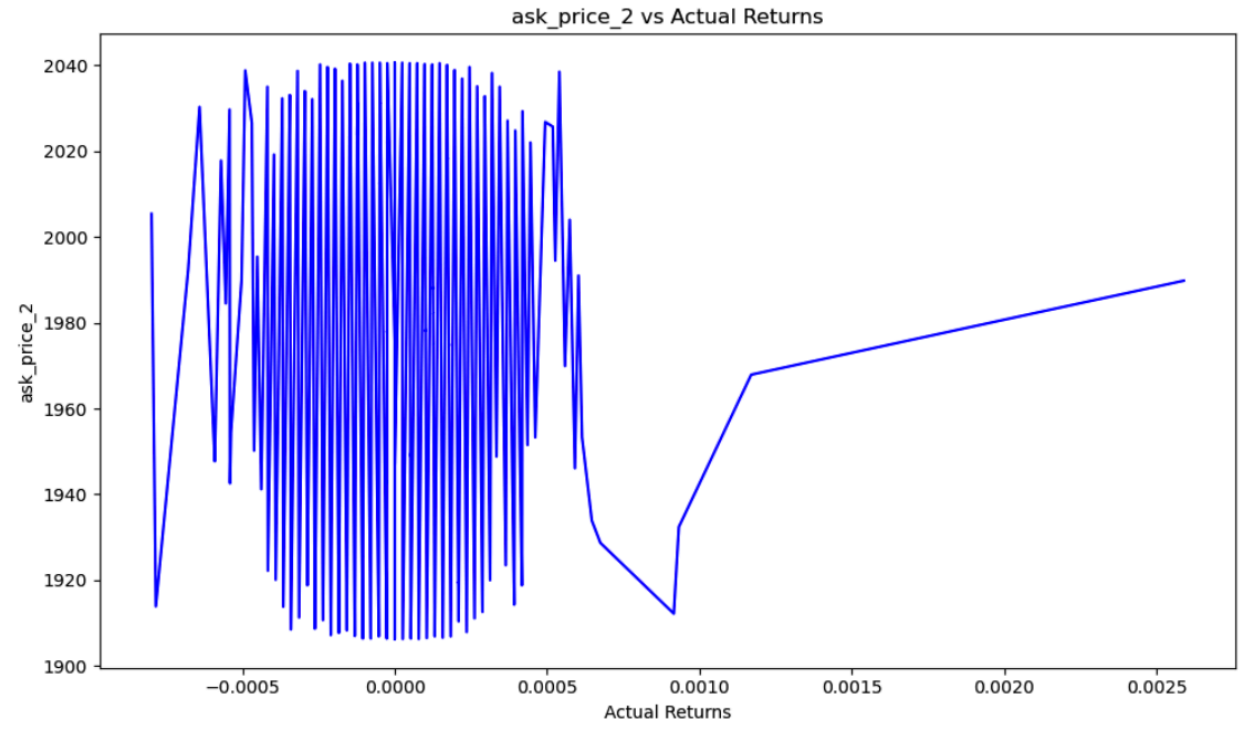
The problem wanted us to find alphas on the situation in hand. There was a limit order book given, with quantities like the aforementioned, in the attached graphs below.

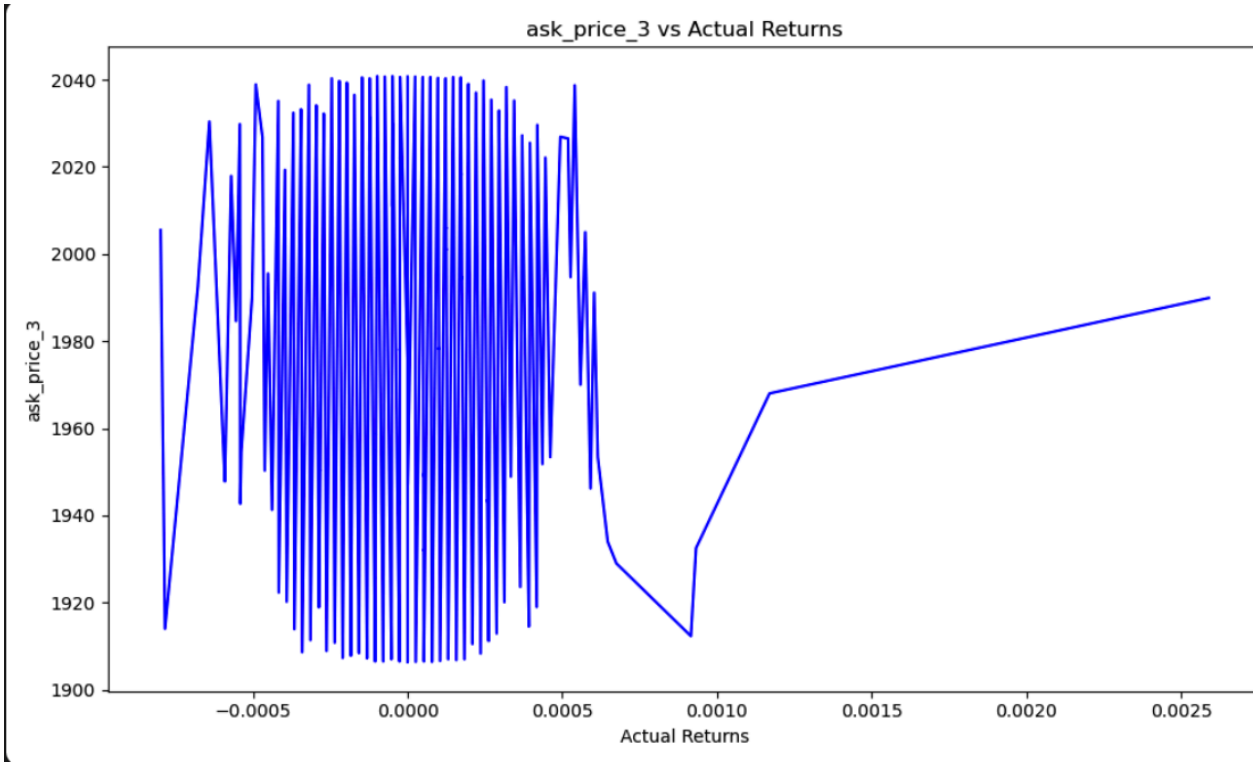
The idea was to be able to find alphas to fit the corresponding train data accurately and be able to test them on the test data, and get correlation relations as close to 1 as possible.

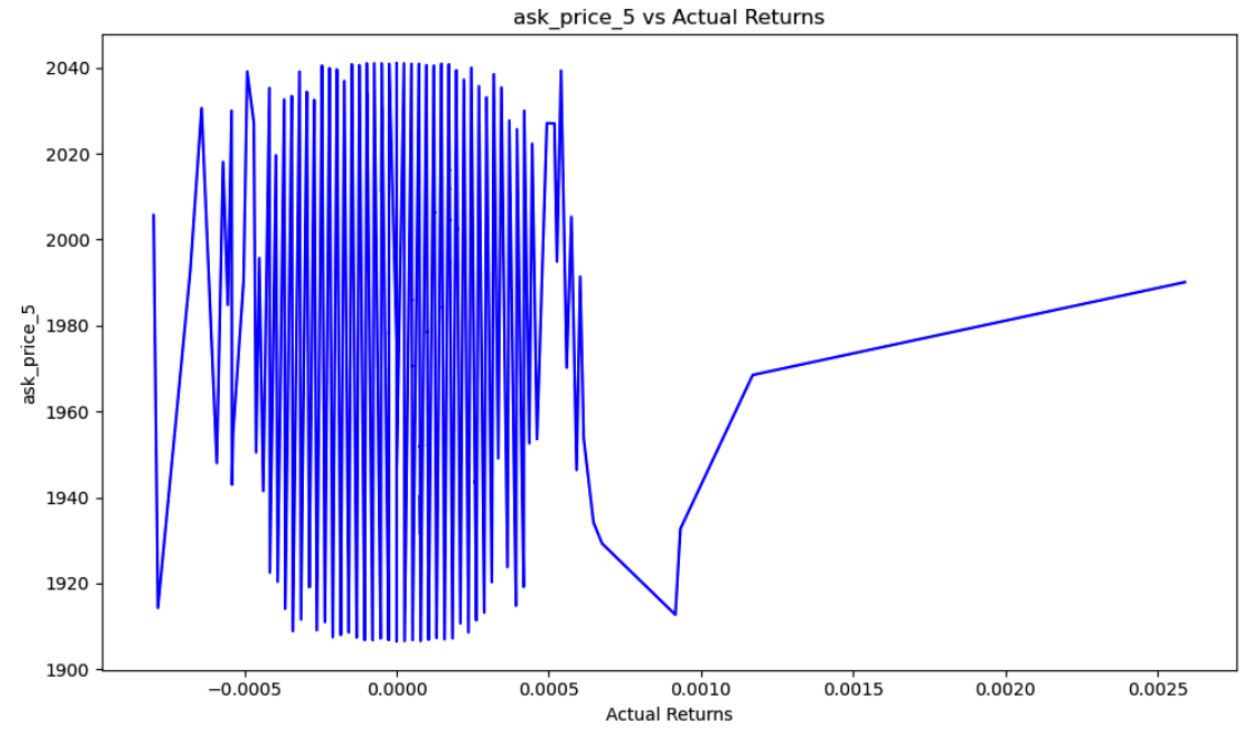
So, the first step, went through the 101 WorldQuant Alphas book, which seemed fascinating the way they came about with so many different alphas, with mathematical complexities associated with them.

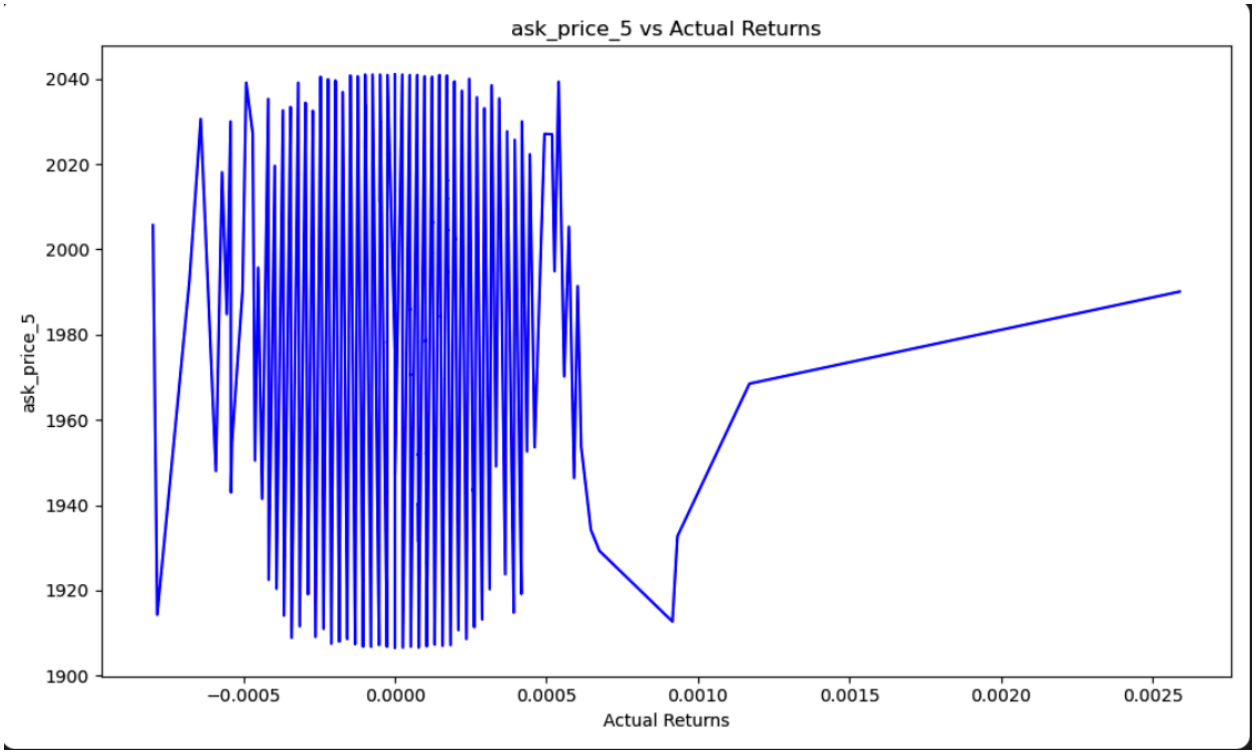
Then, plotted the graphs of the parameters gave interesting visual patterns on being plotted,

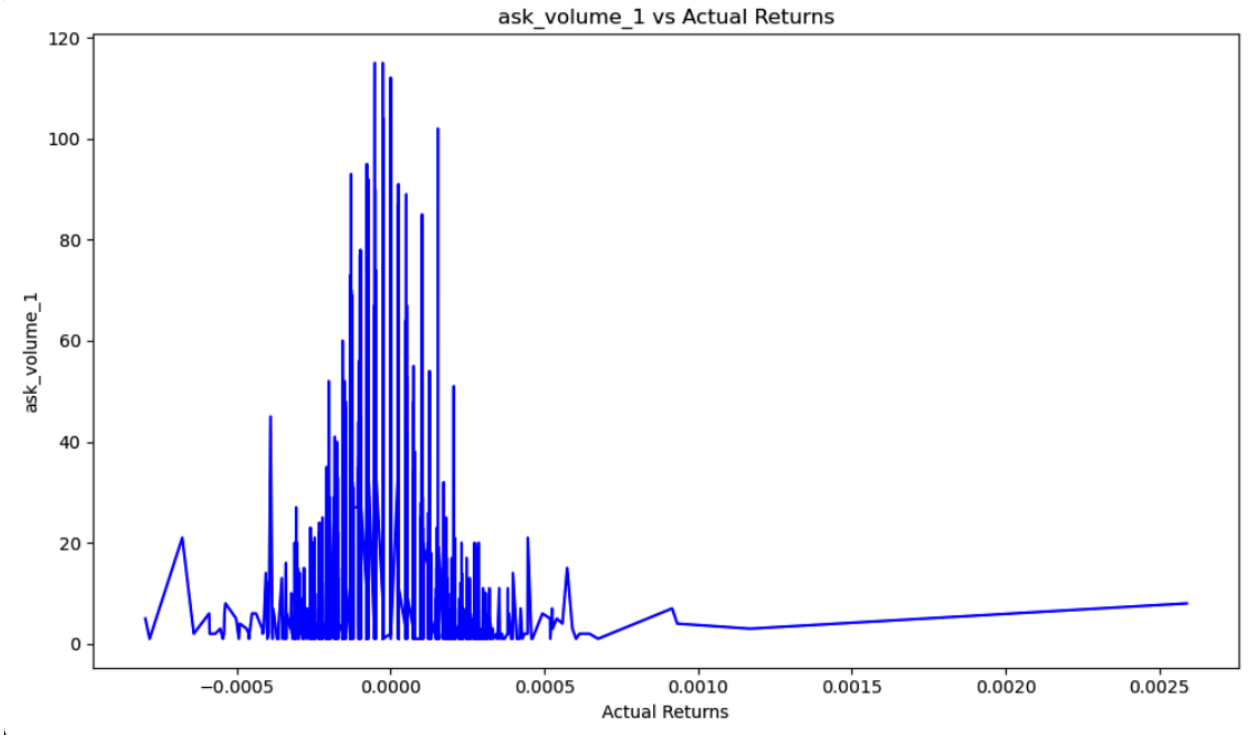

ask_price_1 vs Actual Returns
ask_price_1
Actual Returns
2040
2020
2000
1980
1960
1940
1920
1900
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

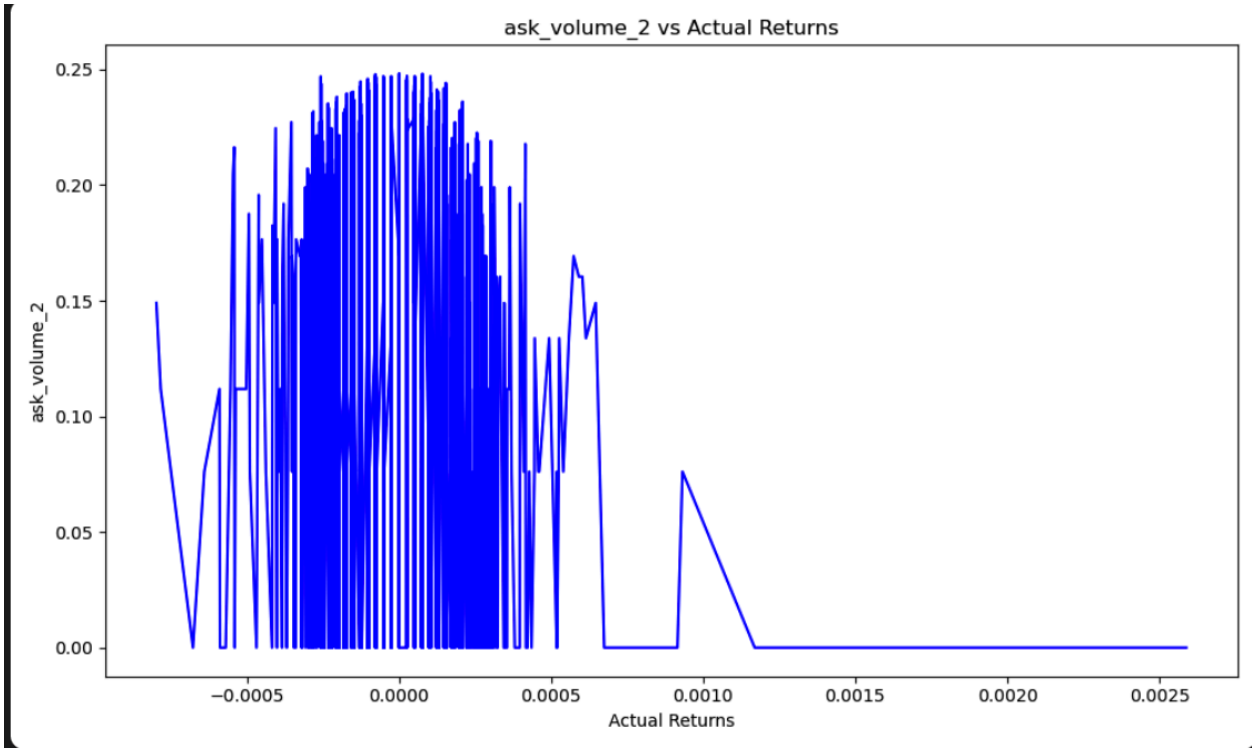

ask_price_2 vs Actual Returns
ask_price_2
Actual Returns
2040
2020
2000
1980
1960
1940
1920
1900
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

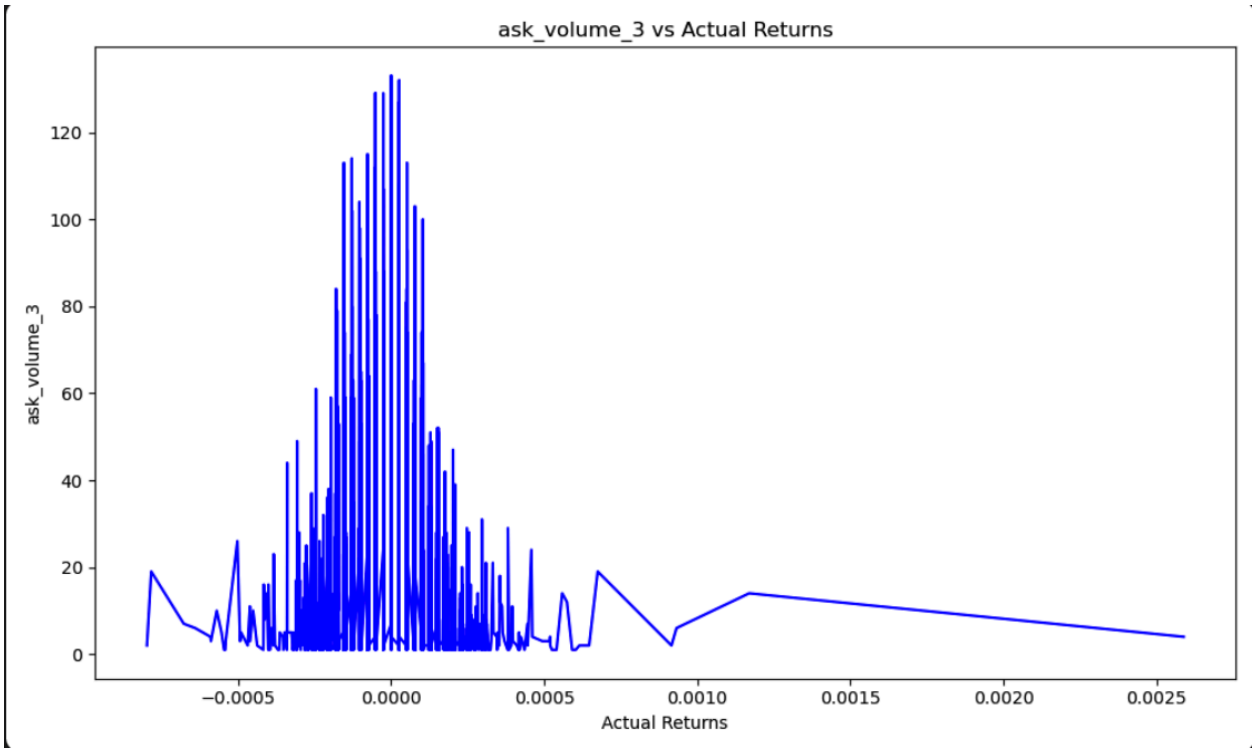

ask_price_3 vs Actual Returns
ask_price_3
Actual Returns
2040
2020
2000
1980
1960
1940
1920
1900
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

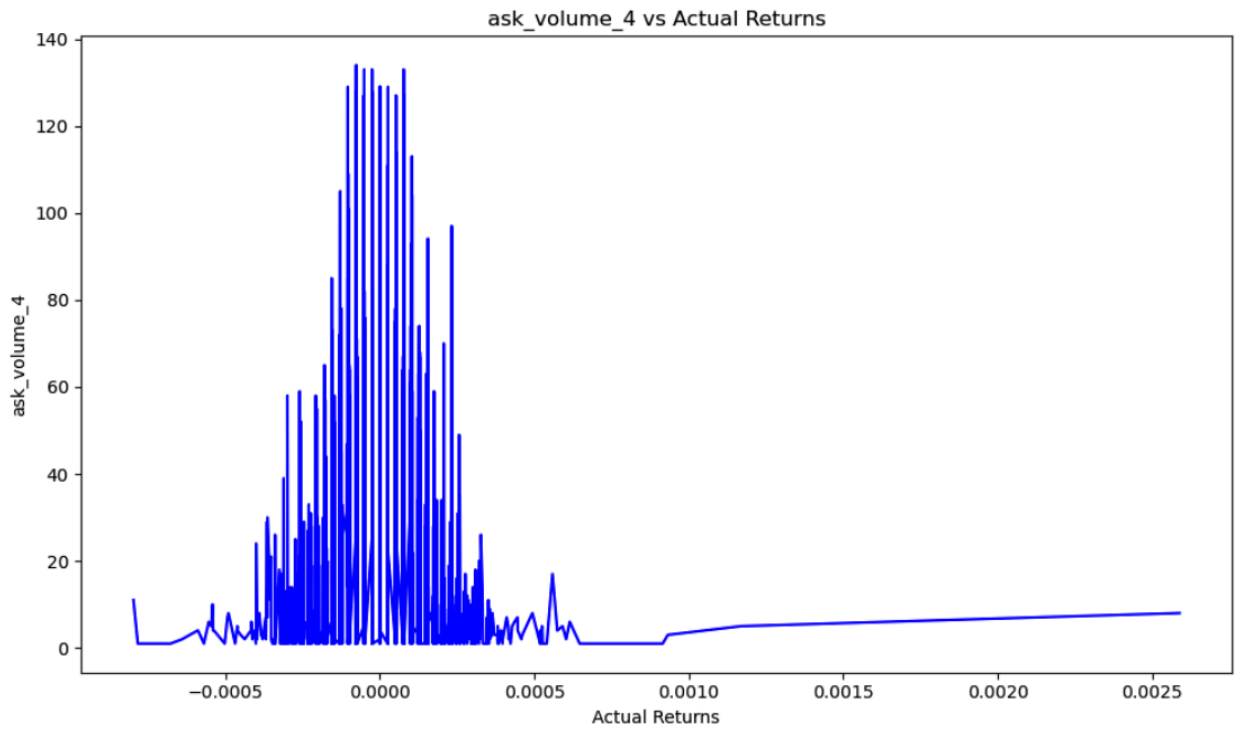

ask_price_5 vs Actual Returns
ask_price_5
Actual Returns
2040
2020
2000
1980
1960
1940
1920
1900
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

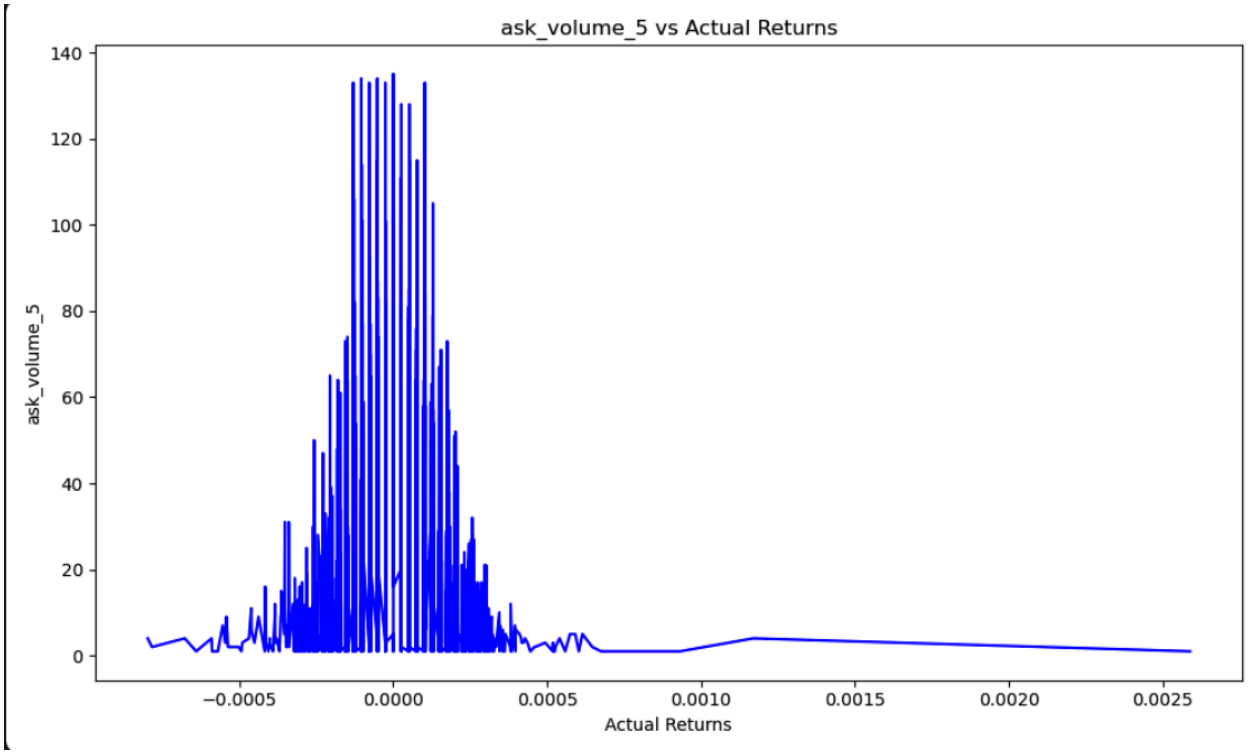

ask_price_5 vs Actual Returns
ask_price_5
Actual Returns
2040
2020
2000
1980
1960
1940
1920
1900
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

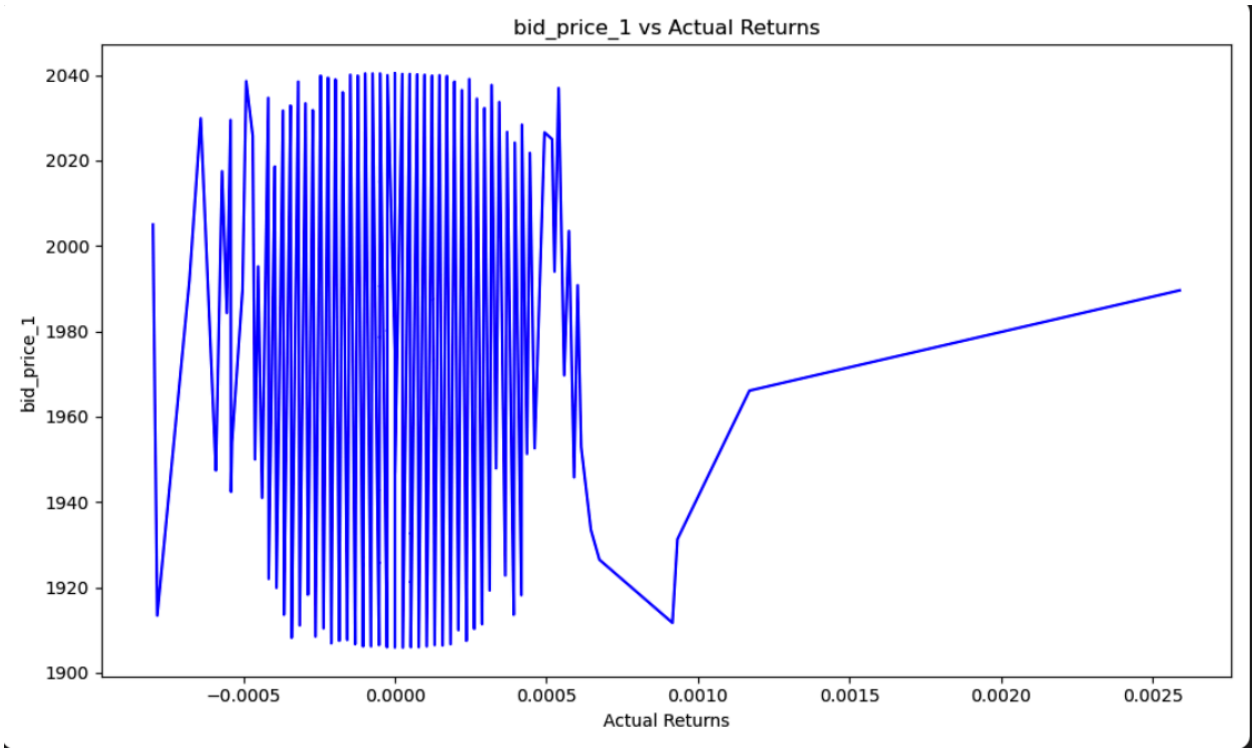

ask_volume_1 vs Actual Returns
ask_volume_1
Actual Returns
120
100
80
60
40
20
0
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

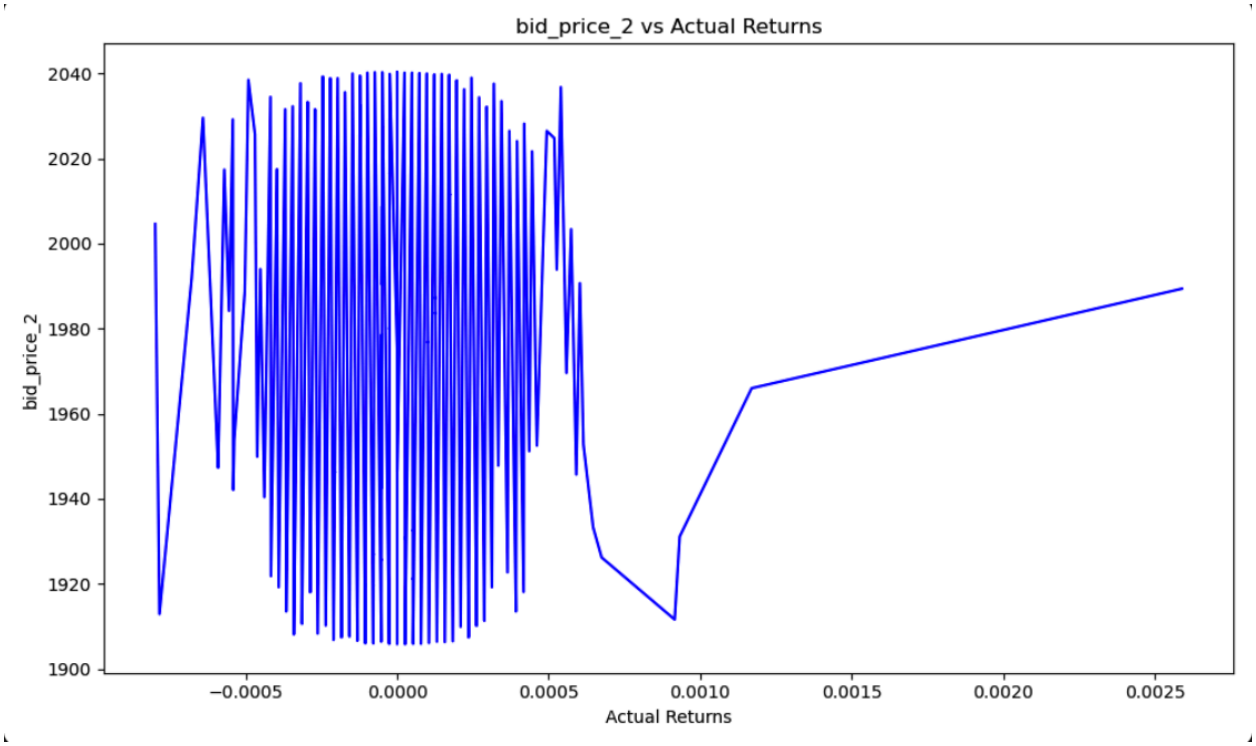

ask_volume_2 vs Actual Returns
ask_volume_2
0.25
0.20
0.15
0.10
0.05
0.00
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025
Actual Returns

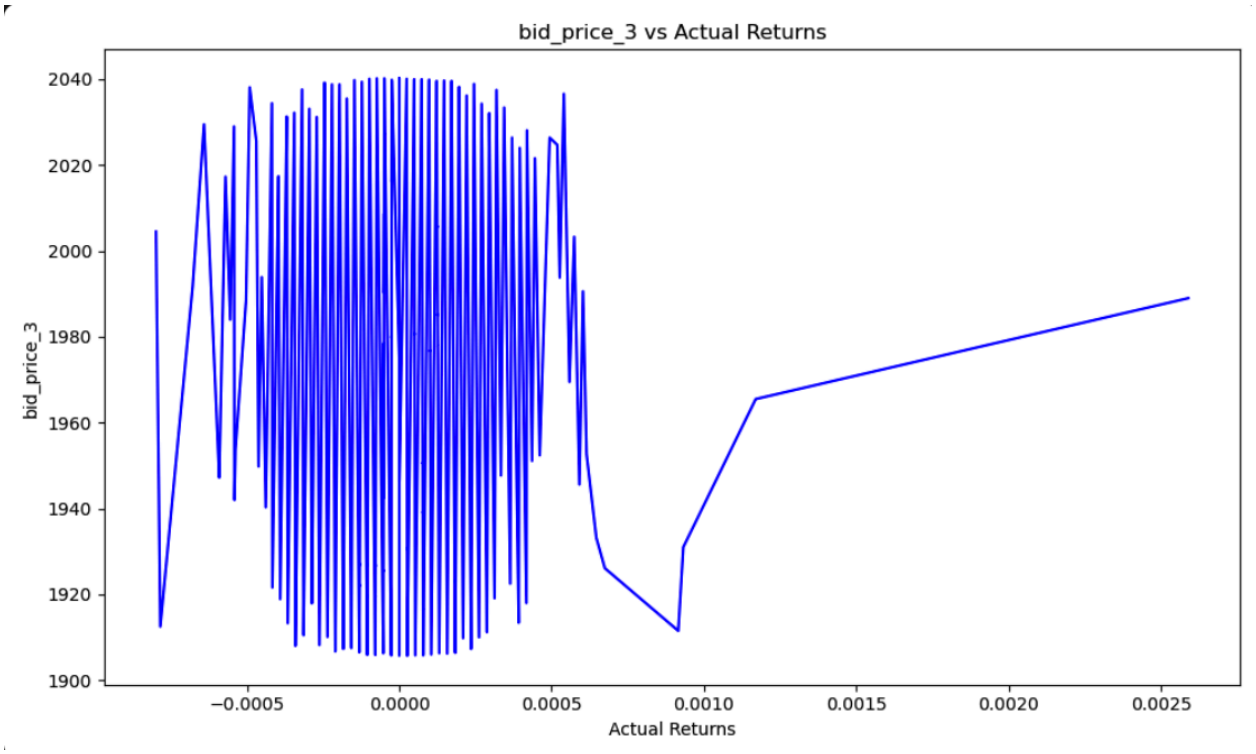

ask_volume_3 vs Actual Returns
ask_volume_3
120
100
80
60
40
20
0
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025
Actual Returns

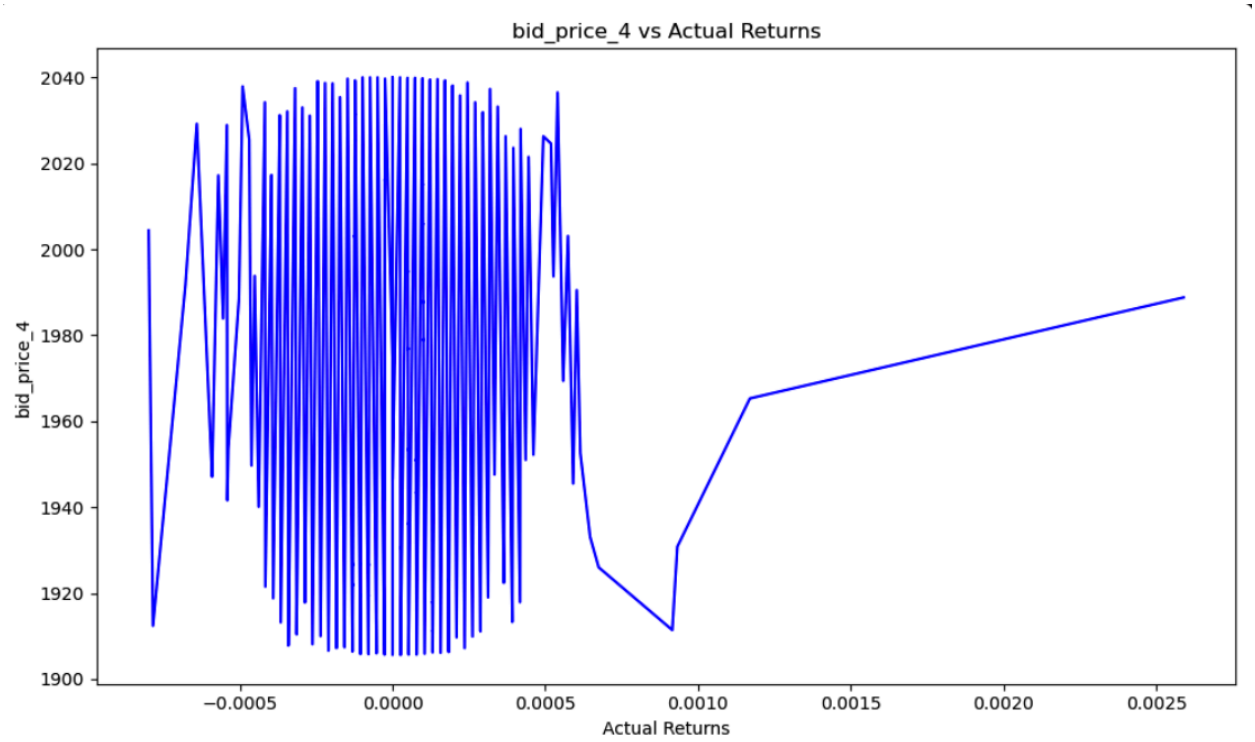

ask_volume_4 vs Actual Returns
ask_volume_4
Actual Returns
140
120
100
80
60
40
20
0
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

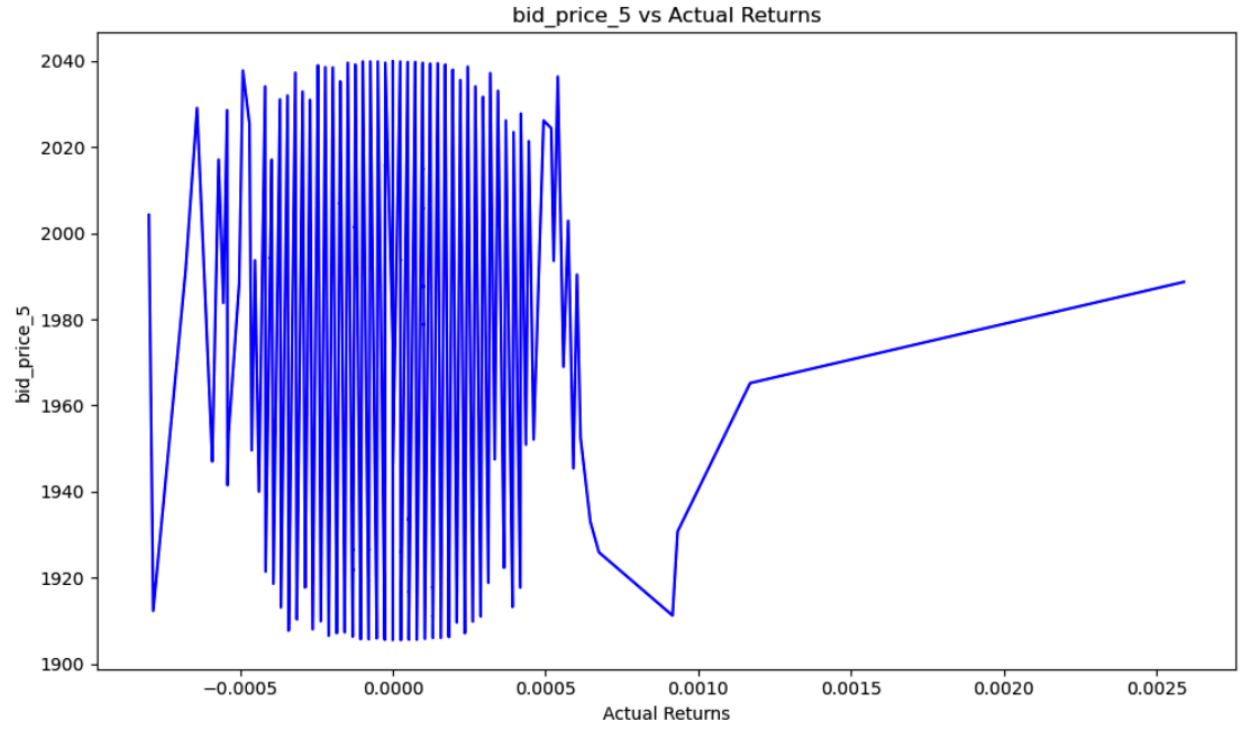

ask_volume_5 vs Actual Returns
ask_volume_5
Actual Returns
140
120
100
80
60
40
20
0
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

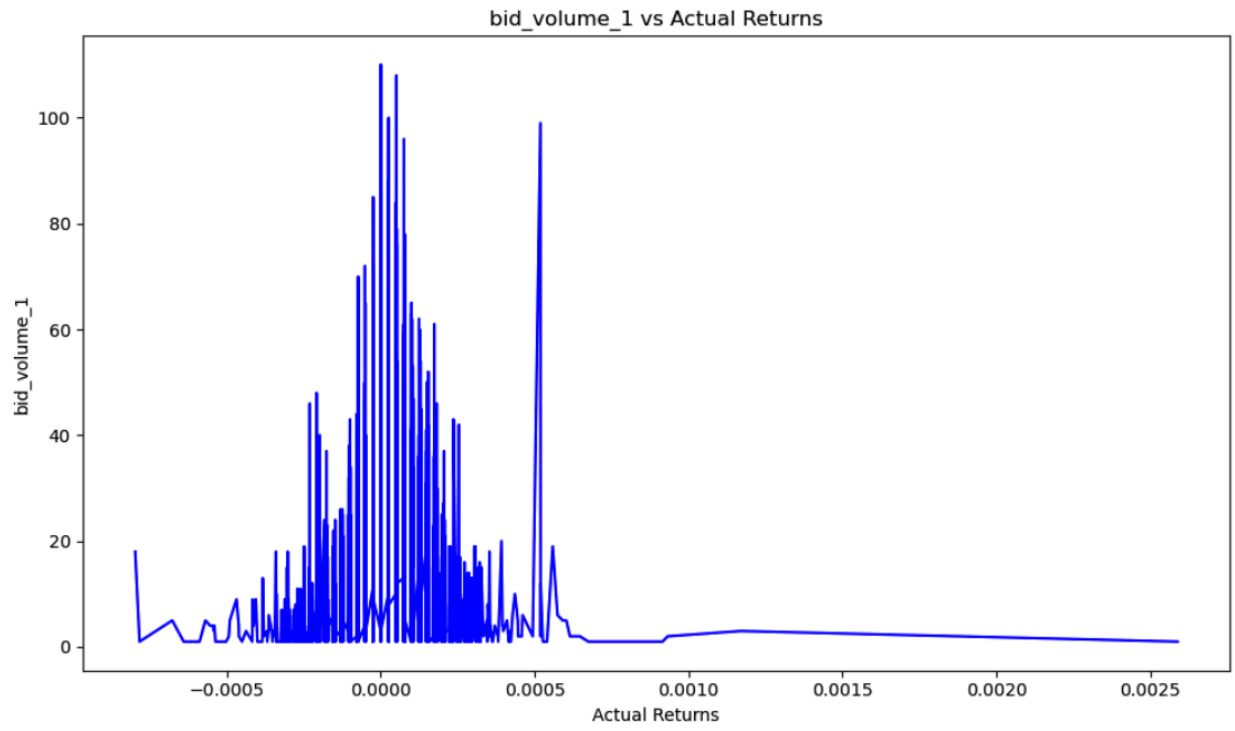

bid_price_1 vs Actual Returns
bid_price_1
Actual Returns
2040
2020
2000
1980
1960
1940
1920
1900
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

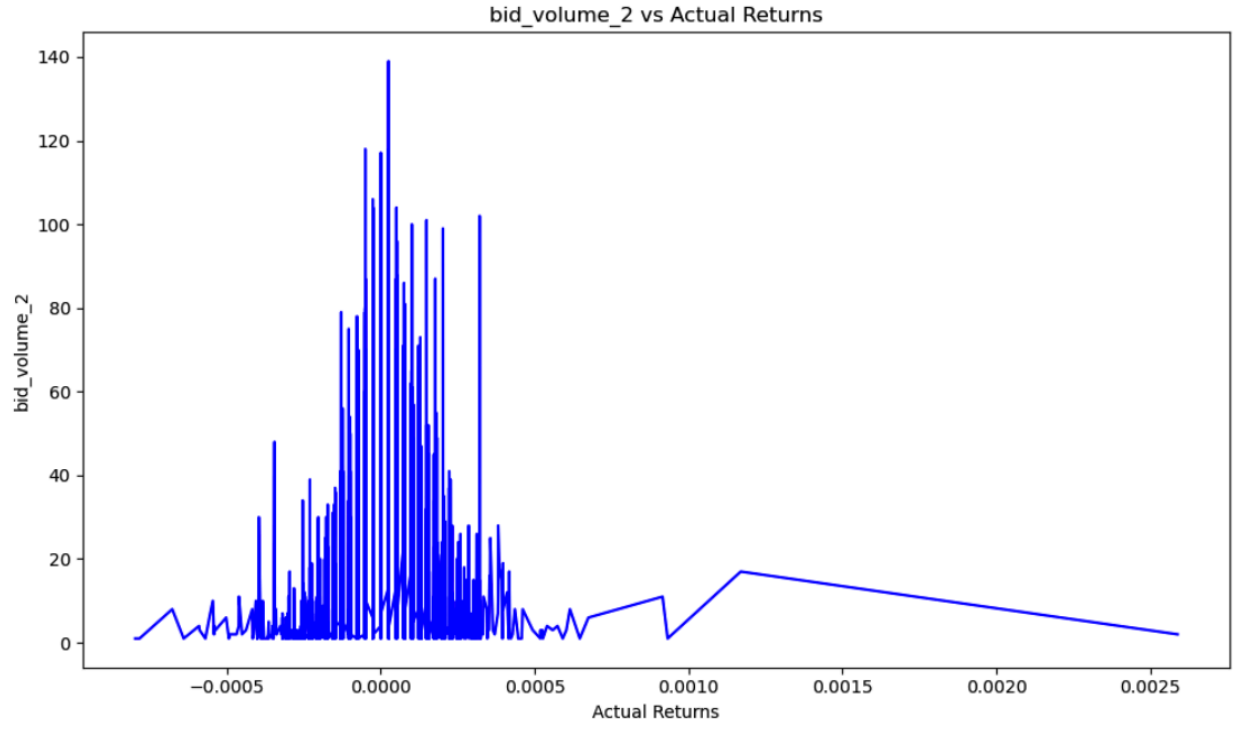

bid_price_2 vs Actual Returns
bid_price_2
Actual Returns
2040
2020
2000
1980
1960
1940
1920
1900
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

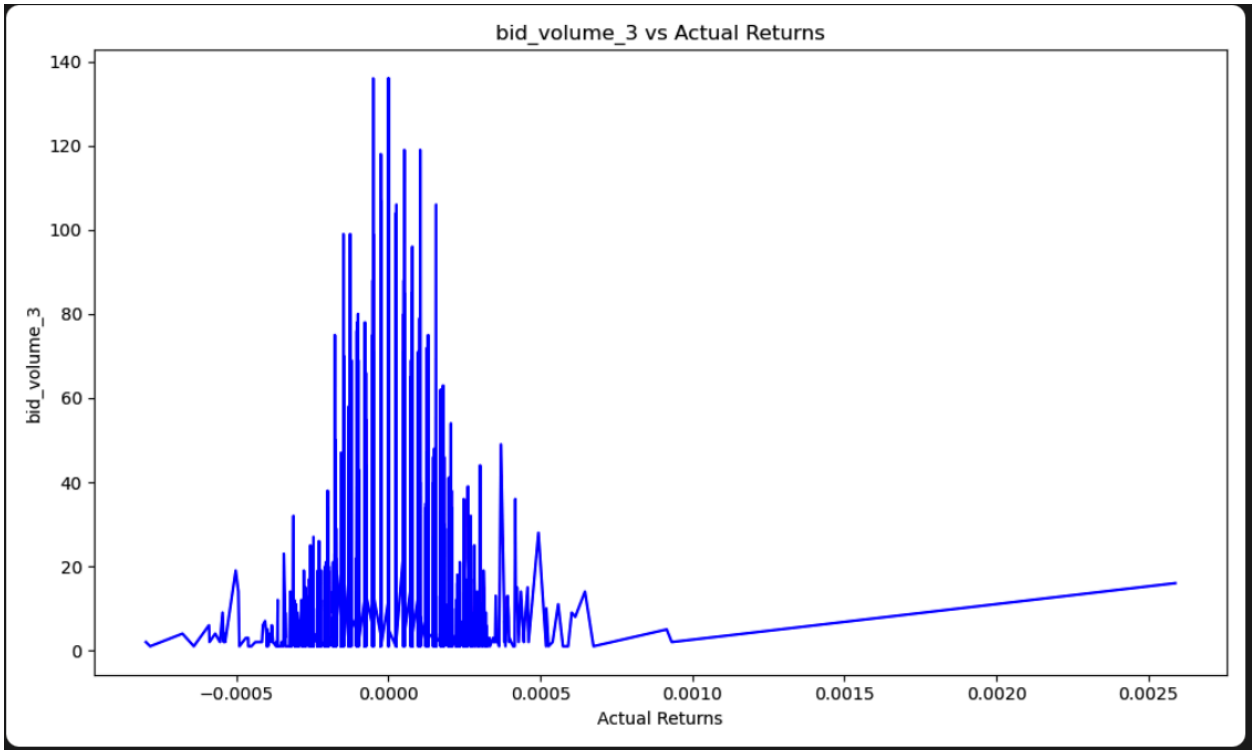

bid_price_3 vs Actual Returns
bid_price_3
Actual Returns
2040
2020
2000
1980
1960
1940
1920
1900
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

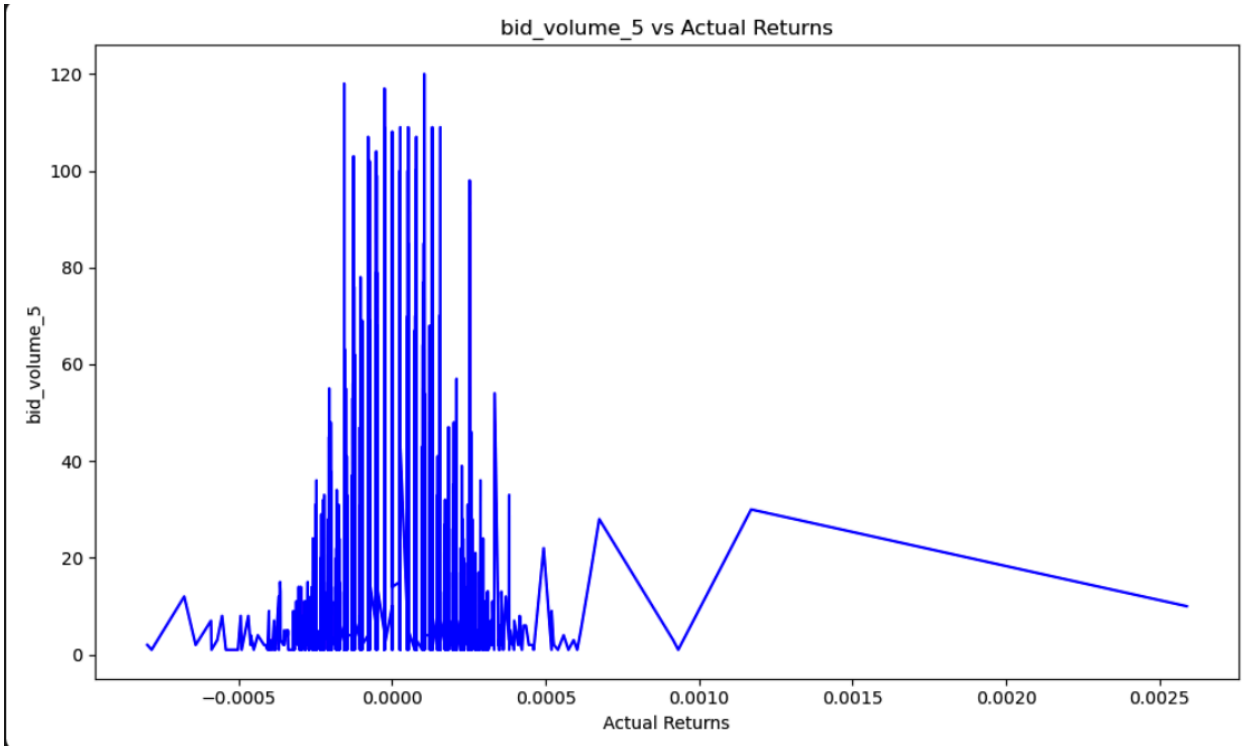

bid_price_4 vs Actual Returns
bid_price_4
Actual Returns
2040
2020
2000
1980
1960
1940
1920
1900
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

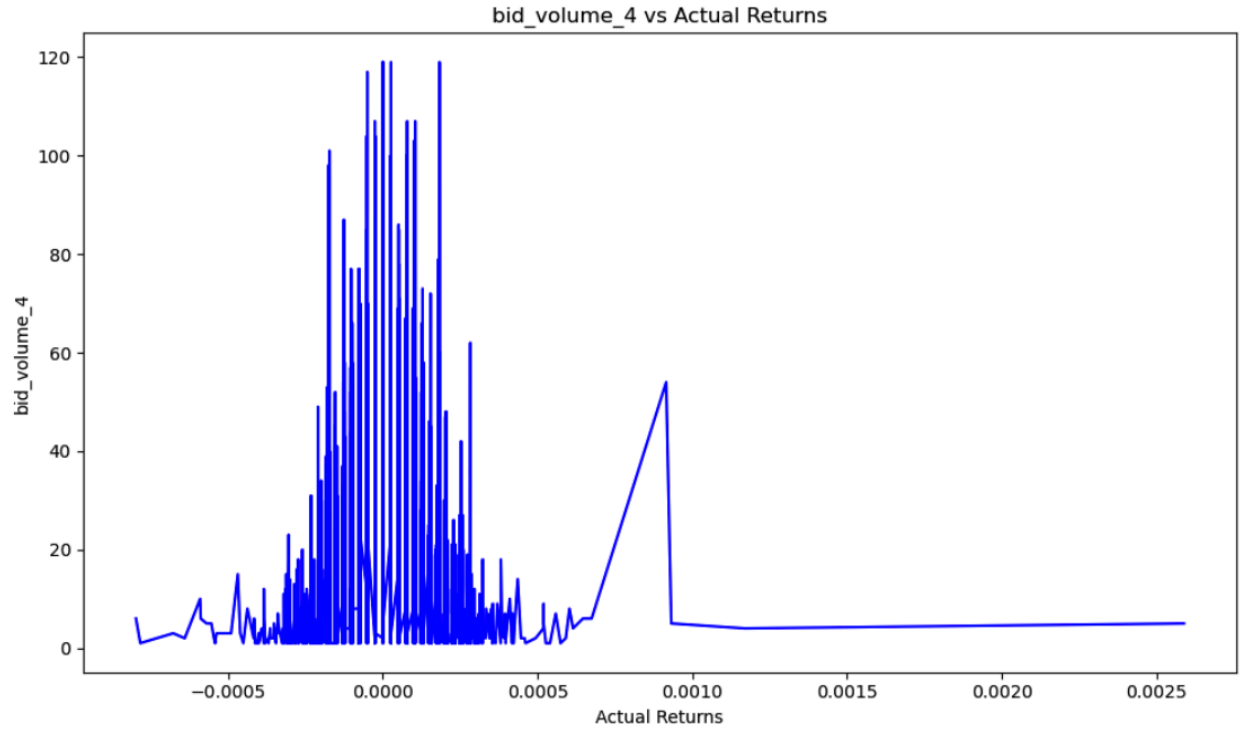

bid_price_5 vs Actual Returns
bid_price_5
Actual Returns
2040
2020
2000
1980
1960
1940
1920
1900
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

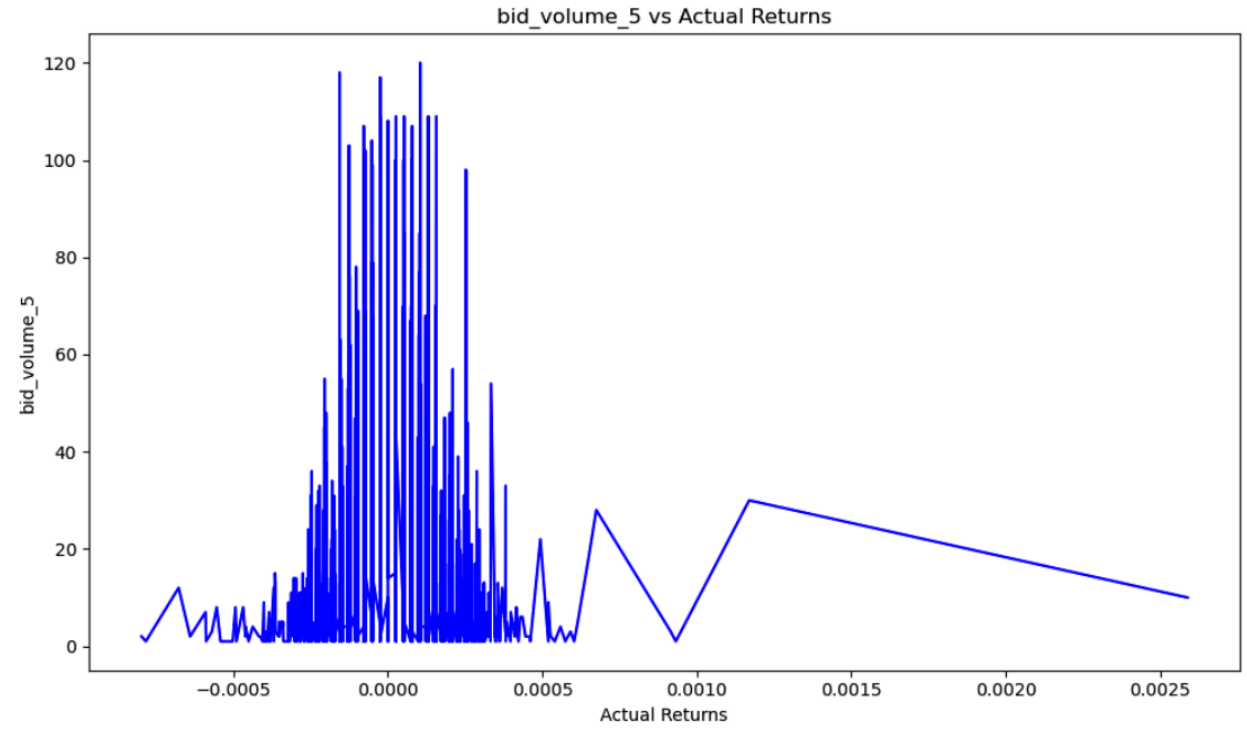

bid_volume_1 vs Actual Returns
bid_volume_1
Actual Returns
0
20
40
60
80
100
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

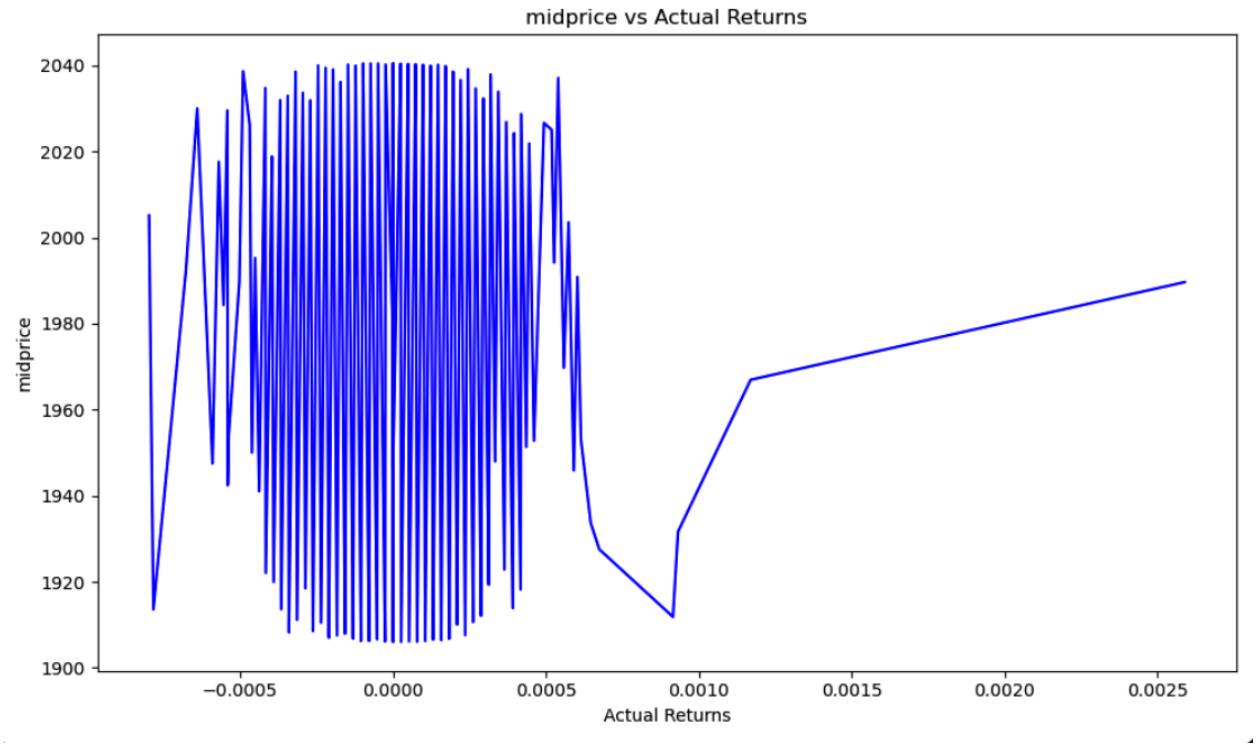

bid_volume_2 vs Actual Returns
bid_volume_2
Actual Returns
0
20
40
60
80
100
120
140
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

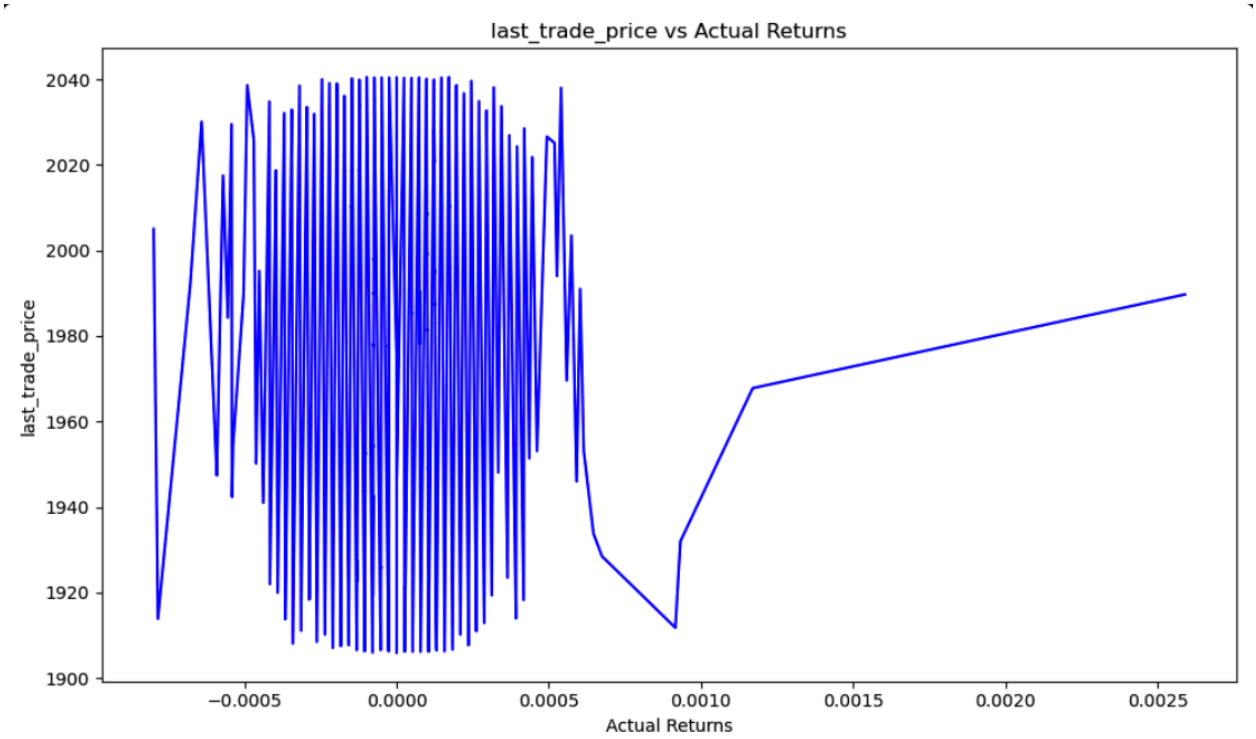

bid_volume_3 vs Actual Returns
bid_volume_3
Actual Returns
140
120
100
80
60
40
20
0
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

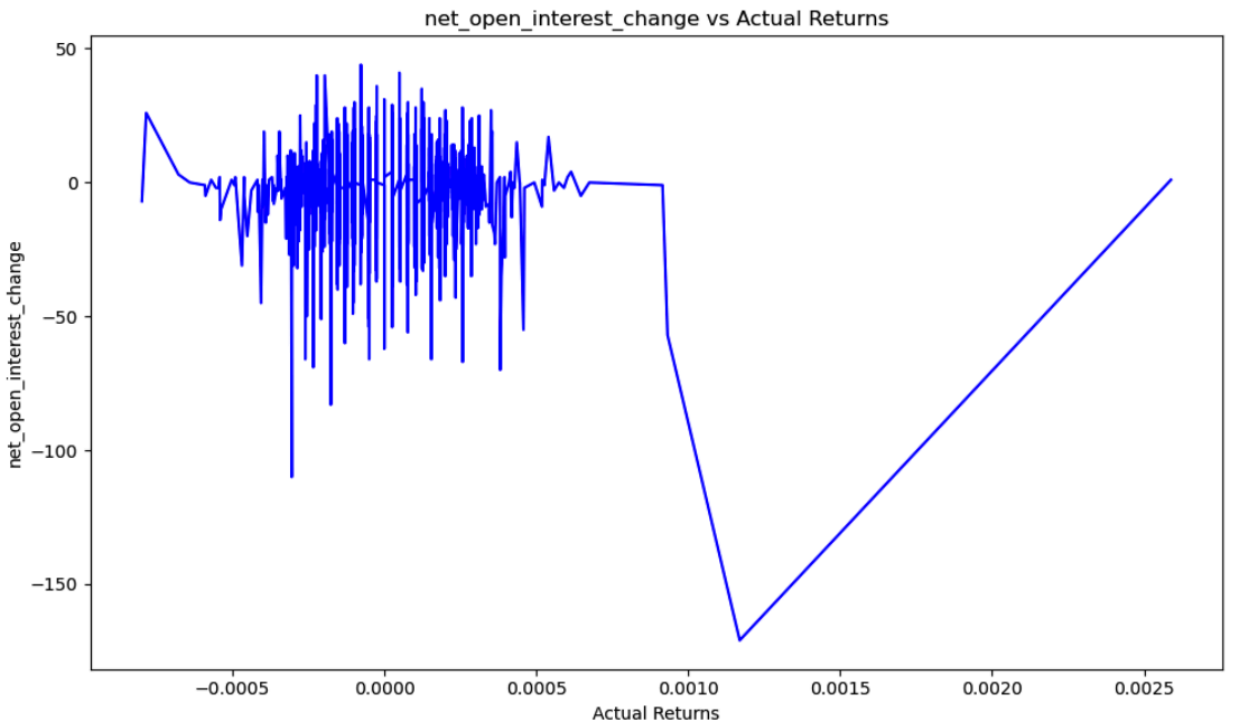

bid_volume_5 vs Actual Returns
bid_volume_5
Actual Returns
120
100
80
60
40
20
0
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

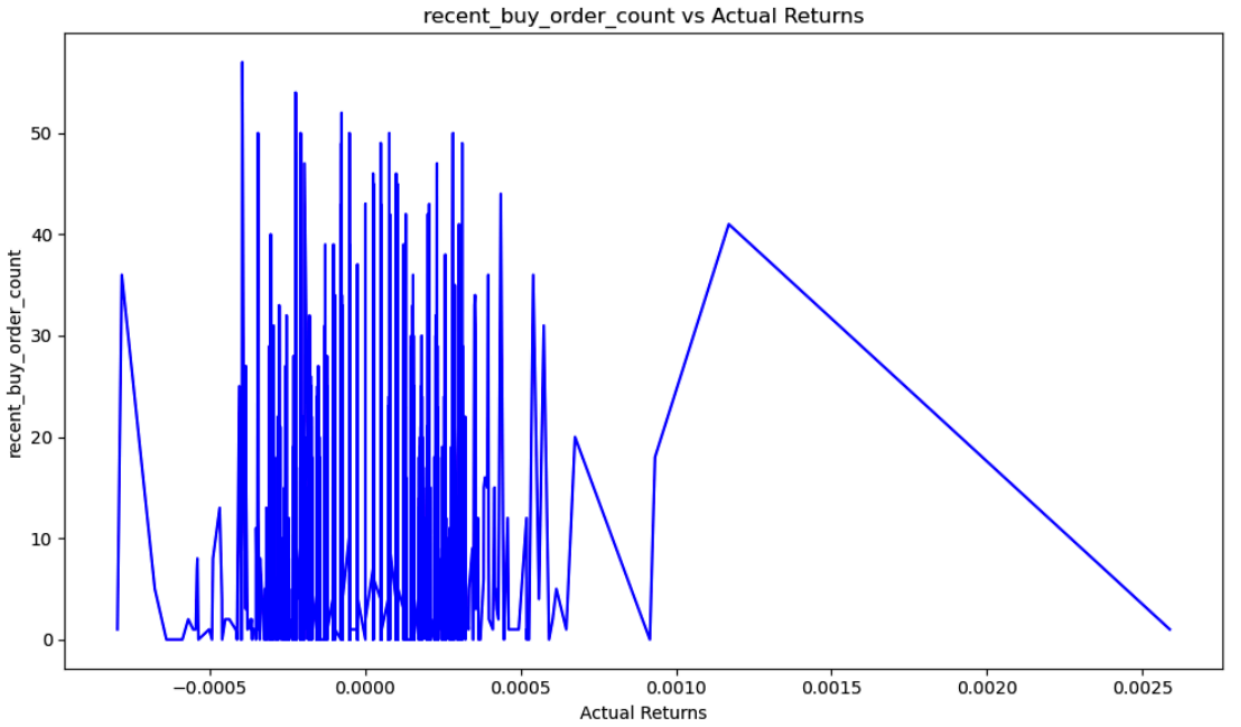

bid_volume_4 vs Actual Returns
bid_volume_4
Actual Returns
120
100
80
60
40
20
0
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

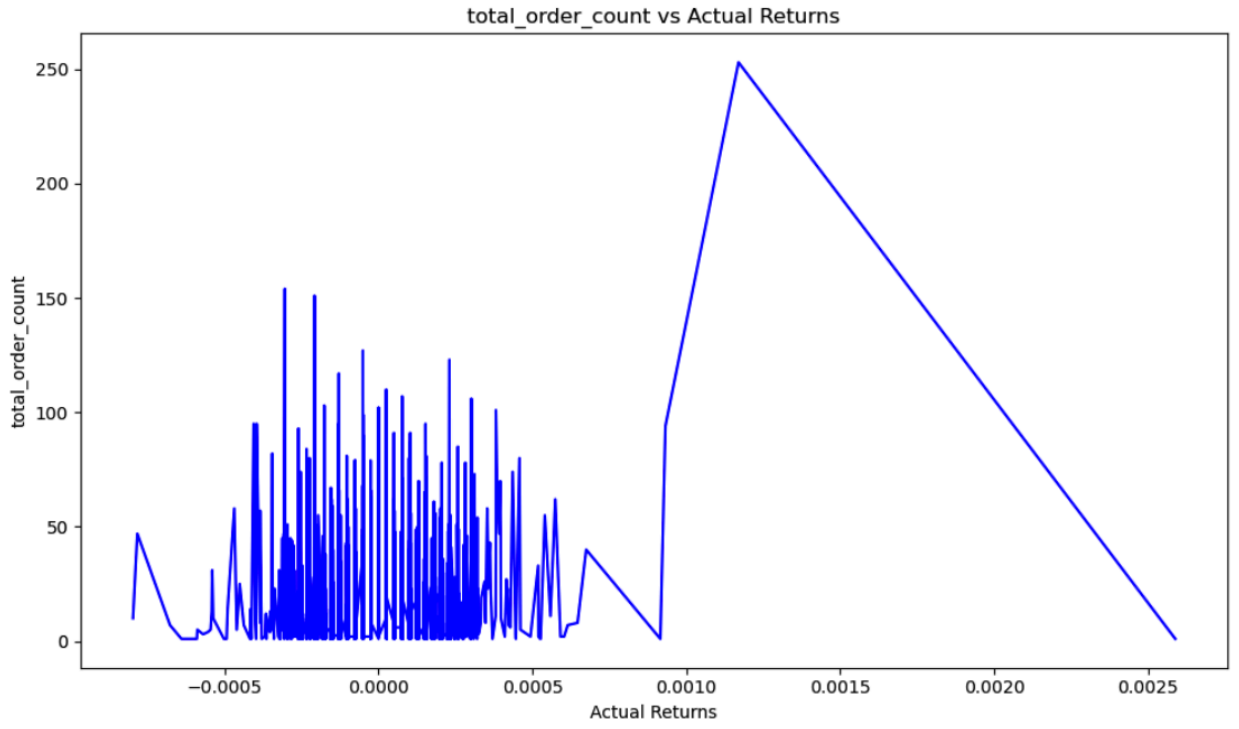

bid_volume_5 vs Actual Returns
bid_volume_5
Actual Returns
120
100
80
60
40
20
0
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025

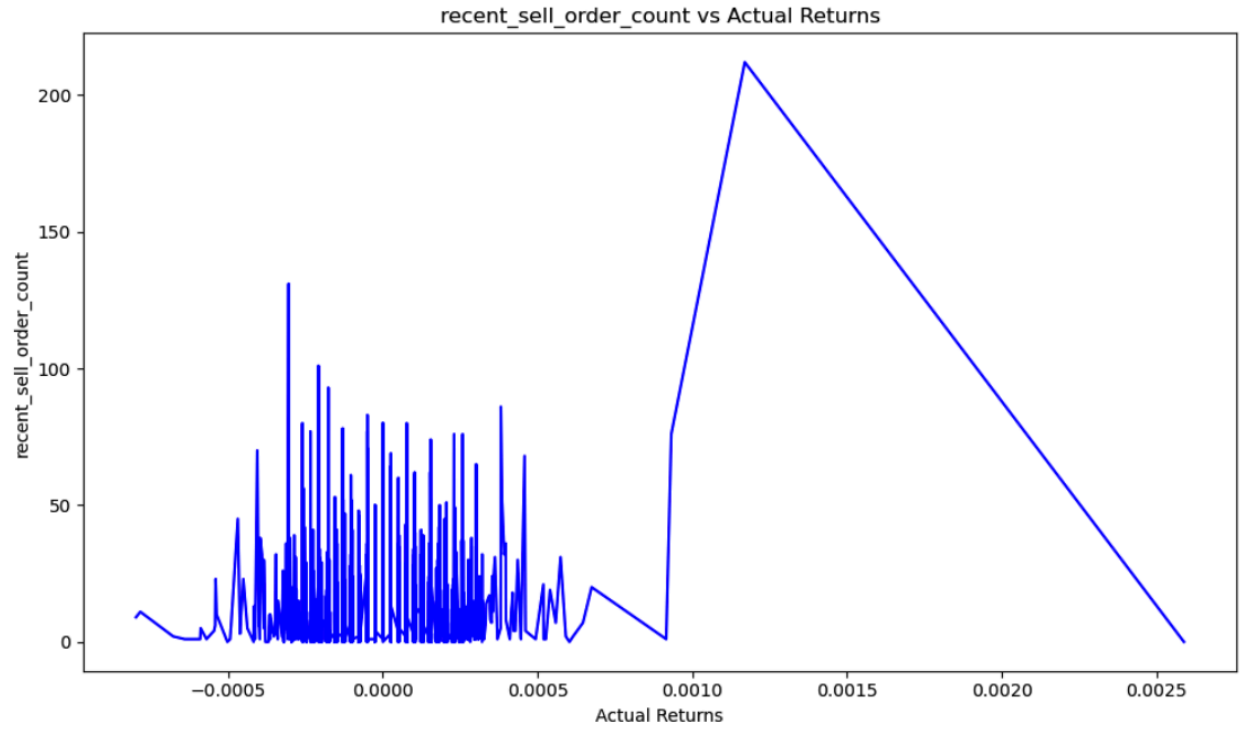

midprice vs Actual Returns
midprice
Actual Returns
2040
2020
2000
1980
1960
1940
1920
1900
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025


last_trade_price vs Actual Returns
last_trade_price
Actual Returns
2040
2020
2000
1980
1960
1940
1920
1900
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025


net_open_interest_change vs Actual Returns
net_open_interest_change
Actual Returns
50
0
−50
−100
−150
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025


recent_buy_order_count vs Actual Returns
recent_buy_order_count
Actual Returns
50
40
30
20
10
0
−0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025





Then proceeded about with the general linear model, trying to incorporate as many features as possible in the model, and then went about finding functions that resemble these plots, and how to incorporate them.

Some general observations included that,

- Working with standard deviations than the deviations or the values themselves proved to be greatly helpful.
- Lasso or Ridge Regression did not give impressive results.
- To predict the models, we need to ensure that the outliers or spikes are not considered in the whole analysis.
- Bid Ask Spread Momentum has an extremely high weightage in the solution.

Features:

- **Price-related**: midprice, last trade price, bid and ask prices
- **Volume-related**: recent buy/sell order counts, bid/ask volumes
- **Liquidity-related**: net open interest change, total order count

Alpha Features: Designed to capture price momentum, liquidity, market activity, and volume imbalances.
Target: `actual_returns`, which you're predicting using the combination of original features and alphas.

The code essentially uses a combination of alphas, polynomial regression and machine learning with financial techniques to obtain desired results.

The StandardScaler feature moves the mean to zero while scaling to unit variance.

The make_pipeline uses all of the aforementioned methods to create a pipeline. Log Transformation is applied to handle distributions that are skewed.

Alphas are used to capture the trends in the train_data and make judgements about market dynamics.

Here's what each transformation achieved:

- **`midprice_squared`**: Takes the log of the midprice to compress large values and manage skewness. This helps stabilize variance and makes the data more Gaussian-like.
- **`last_trade_price`**, **`recent_buy_order_count`**, **`recent_sell_order_count`**, **`net_open_interest_change`**, **`total_order_count`**: These features undergo a similar log transformation to handle skewness and scale values appropriately, particularly for features that could have large outliers or non-negative distributions.

Certain features like bid and ask prices (across multiple levels) and volumes as Gaussian-distributed. These features are standardized using `StandardScaler`. Standardizing them ensures that they have a mean of 0 and a standard deviation of 1.

After creating these alphas, trained a **polynomial regression model** to capture both linear and non-linear relationships between the features and the target variable (future returns). Used a degree-2 polynomial transformation, meaning the model could learn interaction terms and squared effects of features, improving its ability to fit more complex patterns in the data.

Here's the breakdown of the alphas you've created:

- **`alpha_1`: Bid-Ask Price Ratio (Liquidity Indicator)**
  Formula: `rank(bid_price_1 / (ask_price_1 + 1e-9))`
  This alpha measures the ratio between the best bid and ask prices, providing a measure of liquidity. A higher bid compared to ask signals a tight spread, indicating a more liquid market.
- **`alpha_2`: Momentum-Style Alpha Based on Price and Volume**
  Formula: `rank(bid_price_1 * bid_volume_1 - ask_price_1 * ask_volume_1)`
  This alpha captures a momentum indicator by computing a volume-weighted price difference between bids and asks. Higher bid price and volume compared to ask price and volume may signal upward momentum, and vice versa.
- **`alpha_3`: Mean Reversion Based on Trade Prices and Volumes**
  Formula: `rank(last_trade_price - midprice_squared) * rank(recent_buy_order_count - recent_sell_order_count)`
  This alpha combines mean reversion logic (difference between the last trade price and midprice) with a volume imbalance between buy and sell orders. If buy order counts exceed sell orders, this alpha can signal potential price corrections (mean reversion).
- **`alpha_4`: Volume Imbalance (Liquidity Flow)**
  Formula: `rank(recent_buy_order_count - recent_sell_order_count)`
  The difference between recent buy and sell orders. A higher value indicates that buy orders dominate, which could predict upward price pressure.
- **`alpha_5`: Open Interest Change Relative to Total Order Count**
  Formula: `rank(net_open_interest_change / (total_order_count + 1e-9))`
  This alpha measures how open interest (which reflects active participation in the market) changes in proportion to the total order count. A large change in open interest relative to total orders can signal strong market interest, which may influence future price movements.
- **`alpha_6`: Market Activity Indicator Based on Bid-Ask Volume**
  Formula: `rank([bid_volume_1, ask_volume_1].sum(axis=1))`
  This alpha sums the first-level bid and ask volumes, reflecting overall market activity.

High activity on both sides of the book (bids and asks) can signal increased liquidity or volatility, making it an important feature.