**True Beacon Assignment Thought Process**

Firstly, I began to thoroughly read the problem statement at hand. I had prior knowledge of pairs trading but volatility pairs trading was something slightly new to me.

So, initially I went through a few articles on volatility pairs trading to get a better grip over the concept.

Next, I paid attention to the list of deliverables and began making a plan on how I would proceed with the assignment problem.

Downloaded the data file and straight away began to code. Extracted the data in the file to a data frame. Conducted initial EDA exercises to understand data better.

Around 1500-2000 null values were present, and they decided to use the forward fill method to take care of them. There weren’t long patches of missing data hence “ffill” was a simple and efficient way in my opinion.

Also realized that data was present for post-market hours as well as weekends with the values obviously not changing. Hence decided to drop them and use on market hour data to build my strategy.

Converted the date time index into separate columns to help handle data better and make the coding part a bit easier and intuitive.

Initially I began to code the zscore based systems and all subsequent methods I tried in single code cells before compiling them efficiently in one function.

For zscore computation I used a rolling window of about 376 datapoints(minutes) which was about 1 day of trading data and hurst window was over a period of 2 hours (120 points). The spread formula was already provided.

**Thought behind the strategy:**

1. If the zscore goes above(below) the entry(-entry) threshold take a short(long) position in the spread.
2. Else if the abs[zscore] goes below the exit threshold, square off the position.
3. The entry and exit were kept as 1.0 and 0.25 respectively, a few other combinations were tried but these parameters weren’t specifically fine-tuned.
4. No assumptions were made as to how the spread was traded. Hence the PnL and Drawdown metrics were calculated in terms of investing in one unit spread at a time. This was done since only a comparative study was being conducted between strategy methods to find the best and most robust one.

All strategies and their variations tried out along with their results are provided in the excel file attached with the project.

For each strategy a sharpe ratio, total PnL, max drawdown (in rupees per unit) and number of trades was calculated.

Further 2 metrics were created to gauge robustness as well as fair comparison of the strategies.

1. Efficiency Score=(PnL\*Sharpe)/Number of Trades
2. Quality Index= (PnL) / (Number of Trades\*Max Drawdown) \* 100

These metrics worked as a substitute to involving transaction costs associated with the strategy. The metrics gave a better picture of the profit and sharpe generated as opposed to the number of trades taken.

I started my experimentation with a simple Z-score based trading system (Strategy 1). It was straightforward — I calculated the spread, used a 1-day rolling mean and standard deviation to compute z-scores, and generated trades when the spread diverged. It gave decent results (Sharpe: 0.84, PnL: 236.47 per unit spread).

Markets aren’t static — relationships are constantly changing so, I implemented a Kalman Filter to estimate the spread dynamically. Unlike a fixed rolling window, the Kalman approach updates in real time, learning from the latest data. This helped me avoid stale signals and improved my entry/exit timing. It made my model adaptive, which is crucial in volatile environments.

While working with volatility data, I noticed that not every period is mean reverting. Some are trending, and trading during those times leads to losses. So, I added a Hurst Exponent Filter —for regime detection. If H was less than 0.5 (indicating mean reversion), I allowed trades; otherwise, I stayed out. This was done to cut down bad trades.

Out of all the strategies I tested, Strategy 9 (Kalman Filter + Hurst Filter, 1-day lookback, entry Z=1, exit Z=0.25) would be the best for me

* **Sharpe Ratio:** 1.28 — much better than the baseline. (around 52% increase)
* **PnL Final:** ₹349.51 — significantly higher returns.
* **Drawdown:** 0.9 — good risk control.
* **Efficiency score:** 10.5%—around 23.5% increase
* **Trade Quality Index:** 8.88 — solid trade efficiency.

It gave me the best balance between returns, risk control, and consistency.

Some ideas that could be tried are:

1. Volatility targeting: Adjusting your position size based on the volatility of the asset, so that your risk remains constant. Ex: Reducing exposure in volatile markets.
2. Trading differently across market hours: Usually higher volatilities are observed during the initial and final hours of the day. So, the entry and exit signals could be tweaked accordingly to take this into account.
3. Kelly criterion: While I have not made any assumptions on how the spread is traded, once options data is available, capital per trade can be adjusted based on win-loss probabilities to maximise profits.