**Building a Simple Statistical Arbitrage Strategy**

**Objective**

This assignment involves building a simple statistical arbitrage strategy that applies a short-term price reversion signal to China CSI500 stocks. The test consists of three parts. First, you will download the data and do some basic analysis. Second, you will design and backtest a basic trading signal. Third, you will apply a series of refinements to the strategy. This case study aims to test both your investment intuition and programming ability. As a result, you will be assessed on the quality of your code. We are looking for clean, concise, non-repetitive, efficient, and scalable code. We value the ability to iterate quickly through research. Honor code: Feel free to use online resources. However, please submit your own work and do not consult other people.

Your submission includes your python code (in \*.py or \*.ipynb with informative comments) and the table in Section 4 of this document.

**1. Data Acquisition**

Download the csv file and read it into python. This is a table where each row represents each trading day, and each column represents one stock in China CSI500 index. Each cell represents a stock’s close price on a particular day. Please ignore the stock split and dividend for this analysis.

Warm up questions:

Q1. Note that this table contains missing values due to up/down limit and some stocks are not tradable in the early days. Let us define the missing rate for each stock as

missing rate for each stock = (the number of missing values of this stock/the total number of trading days).

Can you plot the histogram of the missing rates of all stocks?

Q2. Let us define the missing rate for each trading day as,

the missing rate for each trading day = the number of missing values on this trading day/500.

Can you plot the missing rate for each trading day as a function of trading day?

**2. Basic Strategy**

We will use the data from the previous section to build a simple investment strategy.

**2.1. Determine Strategy Weights**

Now that we have stock prices, let us build our first trading signal. The intuition behind the strategy is that stocks that experience a rapid decline (rise) in price tend to be oversold (overbought). In other words, the strategy bets on the price reversion by buying losers and selling winners.

For the actual functional form, we will bet in proportion to the log difference between the 22-day and 5 day moving average prices. More precisely, for each stock,

Signal of the i-th stock on j-th trading day = log(22-day moving average price of stock i ) – log(5-day moving average price of stock i )

On a daily basis, individual stock prices tend to be dominated by moves in the broader market. This reduces the effective breadth of the strategy and causes correlation to the market.

We want to create a relative-value model that buys the stocks that have fallen the most and sells the stocks that have risen the most, but is neutral to the market (i.e., always has zero net exposure). We can do this by cross-sectionally Z scoring the signal. For each date, we subtract the average signal and then divide by the standard deviation of the signals. We also multiply by 0.02 for scaling purposes. More precisely, for each date,

Normalized signal for i-th stock on j-th day = 0.02 \* (signal for i-th stock on j-th day – average signal across all stocks on j-th day)/standard deviation of signal across all stocks on j-th day.

**2.2. Backtest Strategy**

Next, let us backtest the strategy. Using the normalized signal as the weights, calculate the daily precent returns to the basic strategy. Next, assuming we start with $1 in 2020/01/04, calculate the cumulative returns to the strategy with daily compounding. To properly assess returns for strategies employing leverage or shorting, we need to calculate returns in excess of cash. Feel free to assume a 5% constant cash rate. Calculate the following summary statistics for 2010/01/04 to 2020/12/31: annualized return, annualized standard deviation, Sharpe Ratio, and max drawdown. You can put these summary statistics in the first row of the first of the two tables in Section 4.

**2.3. Add Transaction Costs**

One unrealistic assumption in our backtest is the absence of transaction costs. To make our simulation more realistic, let us assume we pay 5 bps of the value of the stock per round trip (i.e., 2.5 bps to buy and 2.5 bps to sell). Calculate the summary statistics of the strategy again, this time net of transaction costs. You can put your results into the first row of the second of the two tables in Section 4. Please do not overwrite the code used for the prior simulation (this applies moving forward too).

**3. Strategy Refinements**

We will now attempt to refine the basic strategy.

**3.1. Risk-Weight Portfolio**

The basic strategy puts equal dollar amounts in each stock for a given normalized signal. However, stocks can have wildly varying volatilities (e.g., a defensive utility v.s. highly-leveraged bank). Let us adjust the weights by dividing by volatility. We will use the rolling 6-month volatility of daily returns. Once we have adjusted by volatility, we will multiply by 0.5 to re-scale the portfolio leverage to a similar level to that of the basic strategy.

Adjusted normalized signal of stock i on day j = 0.5 \* normalized signal for stock i and day j / volatility of stock i.

Again, calculate the summary statistics, both gross and net of the transaction costs, and put them in the second rows of the two tables.

**3.2. Neutralize by Volatility Bucket**

At this point, our strategy should be relatively insulated to the market. However, it may still develop biases to other macroeconomic forces. For example, in 2008, if financial stocks all fell more than the market, the strategy could load up dangerously on this non-diversified risk. One standard approach in statistical arbitrage is to neutralize by industrial sector (e.g., staples, financials…). However, we do not have sector mappings at our fingertips. Instead, let us neutralize by volatility buckets. This will help neutralize sectors and macro risks by proxy, as well as explicitly the “betting against beta” (BAB) quant risk factor. We will create 5 buckets of approximately equal number of stocks each, sorted on volatility. For example, bucket 1 will consist of the lowest-volatility stocks, while bucket 5 will contain the highest-volatility stocks. We can use volatility calculated over the past 10-years to define our buckets (this is a little bit of information leak, but that’s ok). Once we have our 5 buckets defined, we will recalculate the normalized signal on a bucket-by-bucket basis. In other words, we will apply the same cross-sectional Z-score technique in Section 2.1, but rather than compare each stock to the rest of the stocks, each stock is compared to the rest of the stocks in its volatility bucket. You can apply the same volatility adjustment and 0.5 scaling as in Section 3.1. For example,

Adjusted neutralized signal for stock i day j= 0.5 \* neutralized signal for stock i day j / volatility of stock i.

Neutralized signal for stock i day j = 0.02 \* (signal for stock i day j – average signal within bucket which stock i belongs to)/standard deviation of signals within the bucket which stock i belongs to.

Calculate the summary statistics (both gross and net) and fill in the tables.

**3.3. Dynamic Volatility-Neutralization (Optional)**

The previous part unrealistically assumed we knew the volatilities from 2010-2020 in all periods prior. Let us redo Section 3.2, but this time we will re-define our volatility buckets every day using the rolling 6-month volatility. Calculate the summary statistics (both gross and net) and fill in the tables.

**4. Summary Statistics**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No Transaction Costs | Return | Volatility | Sharpe Ratio | Max Drawdown |
| Section 2.2 | 0.39730098033379413 | 0.43802886592824686 | 0.8681048380749093 | 0.5600436015543598 |
| Section 3.1 | 0.36569265501135884 | 0.4700005432213975 | 0.7948296214251303 | 0.6591412981702156 |
| Section 3.2 | 0.499879543475241 | 0.43720928113398616 | 1.0327544233827877 | 0.5531278250425873 |
| Section 3.3 |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Transaction Costs | Return | Volatility | Sharpe Ratio | Max Drawdown |
| Section 2.2 | 0.24567124498459414 | 0.43808027414780876 | 0.6056760340338915 | 0.6762881857188652 |
| Section 3.1 | 0.27454454781134263 | 0.47006760756747007 | 0.6477752200257605 | 0.661296710987559 |
| Section 3.2 | 0.39205976754015337 | 0.4372697072507938 | 0.8619100947694662 | 0.5771104163481029 |
| Section 3.3 |  |  |  |  |