**Project Description**

Before getting started with the project description, I am sorry for not using MATLAB to finish this project. Here are the causes/reasons:

1. My MATLAB subscription has expired, and I had trouble on installing the latest version.
2. I have recently used more Python and are more familiar with the functions within it. Therefore, using Python does help me to complete a better project.

I apologize for any inconvenience this may induce.

**File Description**

All of the data is in the **Functions** document while all of the Python Code is in **PythonCode.**

Problem\_1.py

Problem\_2.py

Trading\_algo.py

all\_Result\_300\_69409\_371409.csv

all\_Result\_300\_69409\_371409.csv is the summary of 300 back-testing results with different parameters on trading 69409 and 371409. It is a simulated on Last Price and where we can trade fraction of equity.

**Project Description**

Problem 1:

Problem 1 is more straightforward. The code itself contains comment in explaining the meaning of that sections of code.

Problem 2:

**Project Overview**

Since this project is about pairs trading, the most common way to pairs trading is to find highly correlated and cointegrated pairs of stocks to trade.

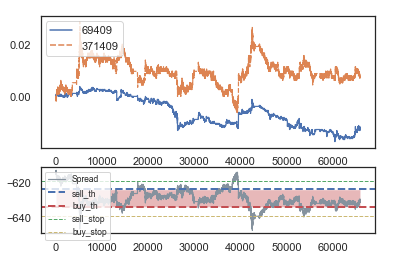


Figure 1:Return Plot and Ratio of Price plot

Above is an example. The first one is the return plot of two securities. We can see the return movement of two futures are very related. As one goes up, the other one follow.

The second plot is the ratio of the price of securities. We can see this ratio is stationary. Therefore, we can use the mean-reverting property to trade. One way to achieve this purpose is Linear Regression. We can regress the price/return of two securities. If the estimated next tick value is above/ below actual value, we can make a trade on it.

Suppose Y denotes for security one, X denotes for security two. Thus, we have:

If the estimated next Y value is above our estimate, we believe Y is overpriced. We can short Y and long X with beta amount. We want to long X in this case is because while Y is over-priced, Y is over-priced relative to X. Thus, we can gain both side of trade by doing long-short in statistical arbitrage.

The summary of trading algorithm development process I used is

1. **Finding highly correlated and stationary pairs (by using cointegration test)**
2. **Trading Using Linear Regression between two underlying (Long one security and short beta amount of the other)**

Note: Beta is the Linear Regression coefficient

(You can find details of the procedure from the code commend.)

**Data Preprocessing**

Here are two assumptions that I have used:

1. There are relatively small changes in price in each second.
2. If price tick is not given, I assume there is not price change. Therefore, I will use the last tick price.
3. The price before the selection time (before 9:15 or 13:30) is continuous to our selection time.

I made the first two assumptions because from the descriptions, we should have two ticks in each second. However, there are times that we do not have two ticks and even worse we have only one tick in couple minutes. Since we are doing pairs trading, we need to align data across different contracts. The way I did is to align data to second data (we have one tick data every second). If there is no tick in couple minutes, I assume there is no price update.

I made the third assumptions because there are cases where we do not have tick data in 9:15. The tick data shows up couple minutes later. It does not make sense to fill data from period afterward. Thus, I decide to take previous data from the last tick before 9:15. Similar logic applies to the afternoon session (1:30 p.m.).

**Pairs Selection**

1. Correlation Test
2. Cointegration Test
3. Liquidity Test (From LastVol)

After data-preprocessing, we have aligned data for 38 contracts.

We can easily compute correlation matrix between them in Python using Pearson distance. Pearson distance is helpful in finding linear correlation. (Note: in the future, we can also apply Spearman distance to find correlation which focus on monotonic relationship. However, since we will use linear regression in trading, we will adopt Pearson distance)

After computing their correlation, we can rearrange them and find the top 5% correlation pairs because computing the P-value in cointegration test takes relative long time, we will simply filter out a lot by choosing top correlation pairs only). Since high correlation does not guarantee cointegration, we will need to do the cointegration for them. **To avoid using whole sample of data which may lead to overfit, we will only use the first day of data in correlation and cointegration test.**

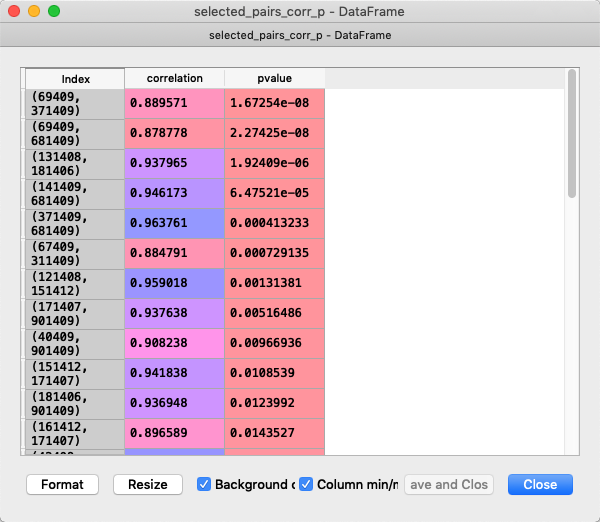


Figure 2: correlation and p-value in cointegration test for each pair

Then, we will compute the LastVol of each contract to examine if they are liquid enough to conduct pairs trading. We will first compute the 70% percentile total Vol of all contracts that are actively traded and get the top 30 most liquid contract from our contract list (most of them are very liquid). **We found that the top five pairs are fairly liquid and pass our test.**

**Back-testing**

In applying Linear Regression in pairs trading, there are lots of case where we need to trade a fraction of contract in order to perfectly hedge. One partial solution to this is to **Pick Y to be the underlying with larger notional amount**. In this way, during regression, our beta will be a value greater than 1 in most of the case, which is easily to round up to integer value without losing precision.

However, this still does not completely solve the issue with trading fraction of contracts. Thus, I will present two cases, one is where we can ideally trade (very possibly a fraction) amount of contract. I will also present the ‘Leverage Approach’, where I trade ten times of the original trade amount and then do rounding. This is realistic because in actual trading, we cannot trade fraction of contracts. We will see the results are not far from each other.

Moreover, there are two price data set I will use in the simulation: 1. Price from Last. 2. Price from Mid (Average of Bid and Ask) which will consider the effect of market friction

Therefore, there are four combinations:

1. Last Price Data with trading fraction amount of contract
2. Last Price Data with trading integer amount of contract
3. Mid-Price Data with trading fraction amount of contract
4. Mid-Price Data with trading integer amount of contract

In back-testing, I have **four** parameters: open, close, stop, lookback.

Open refers to when will we open our positions.

Close refers to when will we close our positions.

Stop refers to when will we stop losing and close our positions.

Lookback refers to we will regress on how many tick data before the current tick.

(Open, Close, Stop constant refers to how many deviation away from the mean spread)

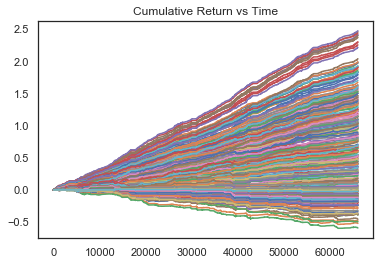


Figure 3: Cumulative Plot of various parameter sets on (69409,371409)

This is the return plot by back-testing 300 sets of parameters on Last Price data set on contract (69409, 371409). The back-testing does not consider transaction fee and it traded continuously from 5/19 to 5/23. You can find the parameter set and their return from all\_Result\_300.csv.

**The result is very good with the best result we get is 247% return in 5 days.**

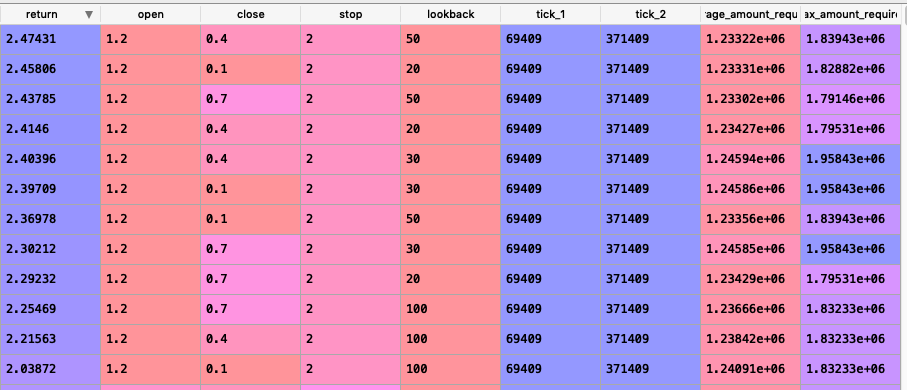


Figure 4: Parameter Set with final return

From the back-testing, we found the following results:

1. lower lookback period leads to higher return.
2. Open position close to 1.2 lead to return.
3. higher stop leads to higher return.

**(Note: the close parameter means we will close our position in open-close and same for stop. For example, if open=1.2, close=0.4, stop=2, we will open at 1.2 std, stop loss at 2std and close at 0.4std)**

From the lesson I got, I did more test on other pairs but with less parameter sets.

Below is the back-testing result on (69409, 681409), (131408, 181406) and (141409,681409):



Figure 5: Cumulative Return on (69409, 681409)



Figure 6: Cumulative Return on (141409, 681409)



Figure 7: Cumulative Return on (131408, 181406)

For the pair (131408, 181406), we also trivially tried longer lookback period, but no great results shown. This is probably because the underlying is not very volatile, and few trade opportunities are available. It is also very likely that our back-testing does not get the optimal parameter set due to only small number of parameters test.

Even though we had found three pairs of contracts showing great return, however, using the last price is not accurate since most of the time we need to consider the friction in the market. We adopt the below parameter set to explore the performance in different data sets:

parameter set (lookback, open, close, stop) = **(50, 1.2, 0.8, 3.2)**

We would like to examine if using Mid price and integer amount of trade.

|  |  |  |
| --- | --- | --- |
|  | Last Price | Mid Price |
| Trade fraction Amount | 364.4% | 24.14% |
| Trade integer Amount | 4M Yuan (~207%) | 270300 Yuan (~11%) |

Since in the trading integer amount, we will compute their actual profit rather than return. The return is computed by dividing the return by the average amount required.

The last two columns are average and maximum amount required to trade, this is computed by multiplying the notional of contract by our average/maximum amount trade each time. From is the result We can see the value is not huge, most hedge fund or professional individual investors can do the trade too.

Thus, we can see the huge impact from market friction in high-frequency trading. There are way more impacts from the price we get (Market Friction) comparing to trading integer amount of contract.

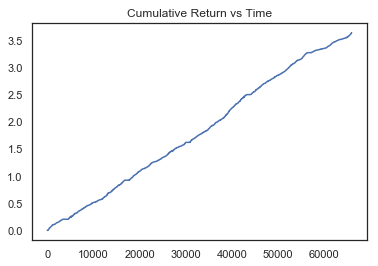


Figure 8: Last Price, fraction contract trade

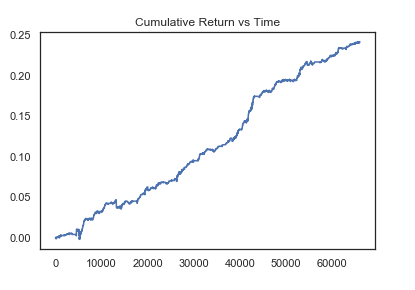


Figure 9 Mid Price, fraction contract trade

**Summary and Future Work**

Below are three pairs and their results:

|  |  |  |  |
| --- | --- | --- | --- |
| TICKS | (69409, 681409) | (141409,681409) | (69409,371409): |
| Cumulative Return(Last Price, Fractional amount) | 501.69% | 205.67% | 364.4% |
| Cumulative Return(Mid Price, Integer amount) | 9.6% | 18.79% | 11% |
| Open | 1.2 | 1.2 | 1.2 |
| Close | 0.4 | 0.5 | 0.8 |
| Stop | 3.2 | 2 | 3.2 |
| Lookback | 20 | 20 | 50 |

The cumulative return using Mid price and integer amount are more similar to actual trading. They all achieve positive results.

Before getting into actual trading, we should back-test more parameter set for each pair individually. Even though we can see cointegration between pairs, they are stationary within some time interval. From the previous work, we saw that the pair (131408, 181406) does not achieve great result by using the parameter having good results on (69409,371409). More back-testing are needed in the that case. Moreover, we also need to consider the issue of transaction cost. In the next step, we could conduct more back-testing on the parameter sets specific on using Mid price. Also, we could also add transaction cost in the back-trading.