To-do list

- Process the data
- Write the backtesting framework and run a simple strategy
- Output the PnL Curve and performance
    - Annual Return
    - Annual Vol
    - Max DD
    - Sharpe ratio
    - Sortino ratio

Process the data

Use double-index function "unstack" in pandas to get two tables: last price & volume.

Dealing with NA value.

Strategy Backtesting

Due to the limited time, I choose a simple momentum strategy:

Denote μ as the expected return, σ as the volatility and A is the risk aversion, we use

$$\mathrm{Mom} = \mu - A\sigma$$

as the momentum signal, choose the top $n = 10$ stocks and invest them with equal weights. A fixed lookback window length and a rebalance periodb are predetermined.

There are many kinds of trading cost: comission, bid-ask spread, tax, market impact, and intra-day volatility... But I assume a market with no trading cost just for simplicity, and a possible improvement is to add a fixed trading cost in proportion to the turnover.

Output and Analysis

Parameters

Initial budget = 10e6 dollars

Lookback window length = 120 trading days

Risk aversion = 1

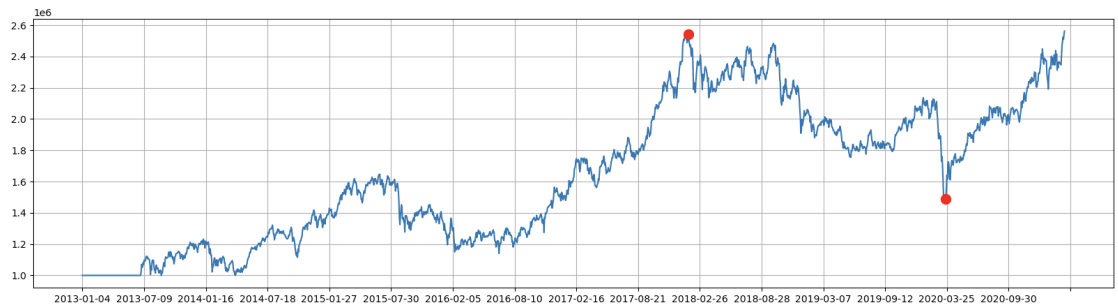
Rebalance every 20 trading days

Performance

```
Annualized Return:  11.832964562928803 %
Annualized Volatility:  23.480414915911798 %
Downside Deviation:  1.194872015854983 %
Max Drawdown(in percentage):  43.70012148554696 %
Max Drawdown(in dollars):  1055904.9004372242
Sharpe Ratio:  0.503950403146839
Sortino Ratio:  9.90312301728968

Process finished with exit code 0
```

PnL Curve



Analysis

The strategy seems to work well in comparsion with risk-free assets in general, especially in the aspect of downside risk. However, if the data set is from US market, the performance is just ordinary, which did not out perform Dow Jones Index significantly.

Consider that I assume there is no transaction cost and stocks have perfect divisibility, the practical result would be worst than this.

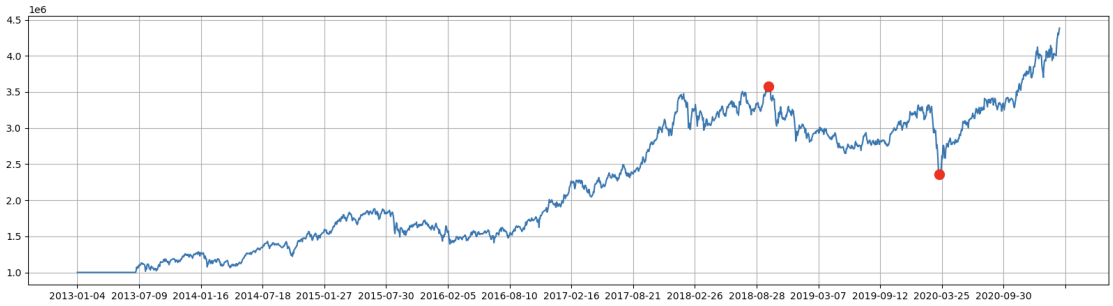
As for the potential improvement, I could try to remove these two assumptions "no transaction cost" and "perfect divisibility" firstly. If the time is enough, I'd like to make use of the "volume" data. And I believe if there are more availble data (like OHLC, intra-day, financial report data..), I could write a better strategy.

Improvement

1. Corrected return calculation

I used log-return to calculated stock return in current trading day, but it was wrong since I used $s_i = (1 + r_i)s_{i-1}$ to calculate the return generated holding stocks later.

Now I corrected it and get the right PnL Curve under fraction-free market:



With the performance table:

```
Annualized Return:  18.586152836986415 %
Annualized Volatility:  23.479296980172894 %
Downside Deviation:  1.1987555237969192 %
Max Drawdown(in percentage):  28.417806024970726 %
Max Drawdown(in dollars):  1222704.4773271917
Sharpe Ratio:  0.7915975019474178
Sortino Ratio:  15.504539889932627
```

And it works much better now.

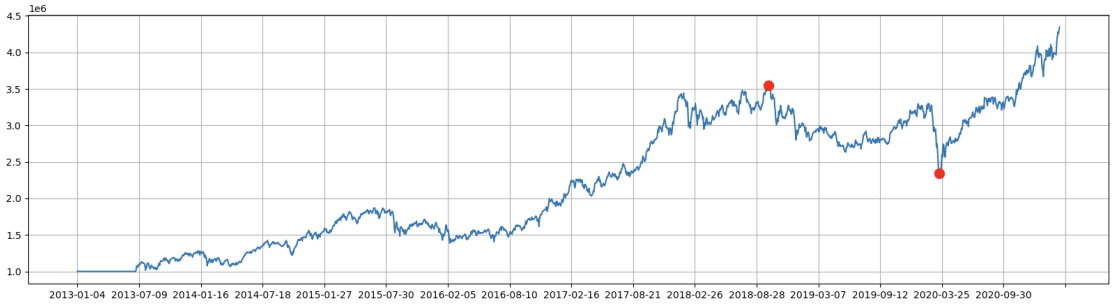
2. Corrected the Downside Risk

I did not use the annualized measurement yesterday, and now is corrected as well.

3. Only Integer Number of Stock is Availble

Yesterday I assumed that stocks are perfectly divisible, but I left an Dataframe "cash" for improvement. Now the trading process is backtested based on the real number of shares.

We can see that there is only little influence of this assumption (annual return from 18.586% to 18.478%). But I think I am supposed to spare no effort to simulate the real trading process.

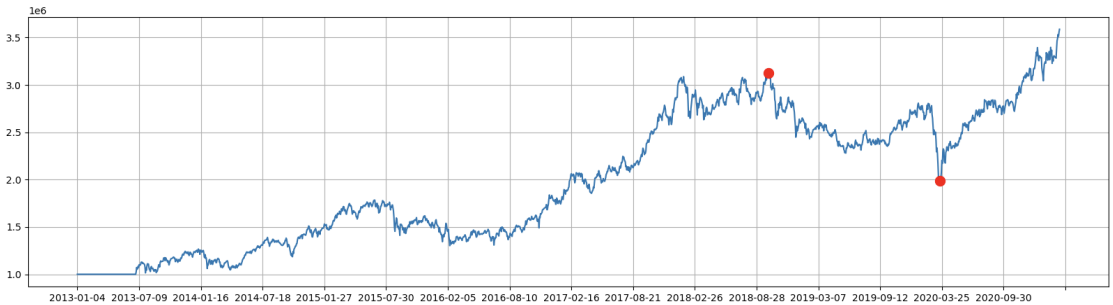



```
Annualized Return:  18.478251002372833 %
Annualized Volatility:  23.289095184245426 %
Downside Deviation:  17.731056714223822 %
Max Drawdown(in percentage):  28.561049364203484 %
Max Drawdown(in dollars):  1207565.3300000038
Sharpe Ratio:  0.7934293220147502
Sortino Ratio:  1.0421404262696659
```

4. Trading Cost is Included

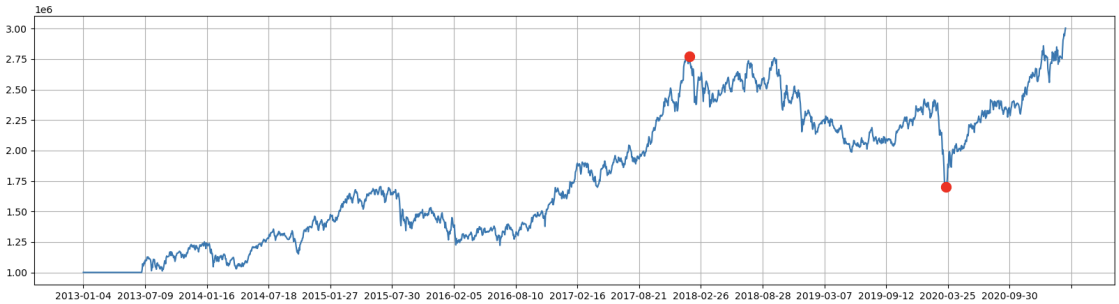
As I stated yesterday, trading cost is an vital part of the market, and now there is a constant rate of trading cost (double-sided) in the backtest framework:

Trading cost = 0.1%



```
Annualized Return:  16.06496458975861 %
Annualized Volatility:  23.273856497645557 %
Downside Deviation:  17.81073876956396 %
Max Drawdown(in percentage):  34.59092440019058 %
Max Drawdown(in dollars):  1136627.2513247966
Sharpe Ratio:  0.6902579549454463
Sortino Ratio:  0.9019819333497479
```

Trading cost = 0.2%




```
Annualized Return:  13.831552491388898 %
Annualized Volatility:  23.28459051163415 %
Downside Deviation:  17.782924992031866 %
Max Drawdown(in percentage):  41.16676597959653 %
Max Drawdown(in dollars):  1072701.2109674027
Sharpe Ratio:  0.5940217194061438
Sortino Ratio:  0.7777996306899181
```

5. Conflict between n Stocks and the Number of Availble Stocks

I used $n = 10$ in my test case, so there is no such a conflict. But for a larger n, it is possible that the length (denoted by M) of all availble stocks. Now I use $\min(n, M)$ for every loop to backtest the portfolio.

6. Deficiency of the Framework

Although I fixed some bugs and made improvement during the last 20 hours, there are still some deficiencies.

The one that I most concerned is that I use the last price of the day before rebalance to calculate trading cost, which should be the open price of the rebalance day. But the open data is not availble for this test.

And another is I did not make full use of all data, the volume data is never used in my strategy.