A Research on Investment Strategy of Enhanced Indexing base on the Greater Bay Area Index

Summary

Enhanced indexing is one of the strategies in quantitative investment which popular nowadays. As one of four bay area around the world, Guangdong-Hong Kong-Macao Greater Bay Area coverage huge amount of talent people and funds. So, the financial market of Greater Bay Area is considered as one of the most potential market by investors. Our model picks the greatest ten stocks as the investment object next week base on the historical rate of return. In Question 1, the investment amount of each stock is limited to 10% of total amount, it means that we just need to choose ten stocks and equally distribute the investment amount from the total. In Question 2, the investment amounts are changeable, so we give each stocks a score by their performance in last 7 days, and divide the total amounts in portion compute by the scores. In Question 3, we use the same method as Q2 but extend the time to 30 days.

In Question 4, we used the same method as Q3 but the buy time and sell time are decided by using theory of Optimal Stopping. Our research are based on the given Greater Bay Index, by testing different stock choice strategy, presenting a portfolio absolutely better then the Greater Bay Index. Simultaneously, reduce the computing performance demand by optimize our program.

Keywords: The Greater Bay Area, the theory of Optimal Stopping, Linear Regression, Weighted operation.

I Problem Restatement

1.1 the Background of Problems

Among the common methods of investment and financial management nowadays, Stocks are worthy of the highest risk one. The main reason of that is the stock price are not only influenced by the conditions of company's operating, supply and demand, interest rate and reserve of banks, psychology of investor, but more also the price of gold, foreign exchange, bond or commodity, and mainly the political economy. However, building and development of stock market makes an important impact in facilitating capital flow, improving the efficiency of using of funds, promoting enterprise and market system reform. Meanwhile, the uncertainty of its income cause by its characteristics of high risk and 'high benefit' let more and more people who want to seeking thrill enter the stock market.

As the winner of 1981 Nobel Prize in Economics, James Tobin said, *don't put all your eggs in one basket*. Diversify investment can reduce the risk. So, we should diversify investment rather than put all our principle in one stock to lower the risk of investment and at the same time, exceed the rate or return of stock index as much as possible. Therefore, Enhanced Indexing strategy comes into being.

Enhanced Indexing strategy adopts quantitative enhancement models, pursue an investment which has a return level above the target index, and strive to effective risk control, lower the transaction cost and optimize the portfolio. It will not fully copy the constituent stocks in tracking index, instead, increasing weights of optimistic stocks, remove or decreasing weights of pessimistic stocks. By keeping on monitoring the model of transaction cost, minimize transaction cost as possible. In summary, it takes excess income and meanwhile controls active risk.

1.2 Problem Restatement

A securities company constructs the Greater Bay Area Index by picking 30 gold stocks in the Greater Bay Area. The Enhanced Indexing strategy the company adopted currently is, picking up the 10 strongest stocks in last week in 30 constituent stocks as investment target of next week. Swap at every Monday, calculate the benefits by the closing price that day. Investment amounts of every stocks are fixed as 10% of principal, handling fee is 2.5 ‰o.

1) Calculate the yield curve of this investment strategy in 2011-2020, evaluate its investment effectiveness referring to the Greater Bay Index.
2) If the time of swap is constant, the investment amounts of single stock are variable. Find out the optimal yield curve of this investment strategy.
3) If the time of swap is constant, the investment amounts of single stock are fixed as 10% of principal. Design your own strategy of stocks picking, construct a model base on the market trade data.
4) If the time of swap and the investment amounts of single stock are variable. Design your own strategy of stocks picking, construct a model base on the market trade data.

   Requirement:

Complete the position in the first three trading days, position not less than 50%, each stock cannot exceed 10% of total assets. Model scoring factor reference: both care of profit and risk, the model is reasonable, applicable, have a certain innovation.

II Problem Analysis

According the trade data of 30 stocks between 2011 to 2020, we'll analysis and process the data provided, construct math model of our investment strategy.

2.1 Analysis of Question 1

First, we pick the 'strong' stock by considering the average value of rate of return of trading day last week.

Assume the actual purchase amount that deduction fee as , introduce the closing price of days and the opening price of that day , the number of shares of that day is , the rate of return of that day is , then substituting the data of trading days in each week into the formula above and averaging to get the average rate of return of the stock and record it as .

The stock buying method: Because every purchase requires a handling fee of 2.5‱, assume the principal as , therefore the actual purchase amount of the first time is , the average distribution to each stock is . So, the sum of the income and principal of the stock purchased for the first time is . The sum of the first income and principal is used as the second principal. At the same time, it should be noted that since the position is exchanged every Monday, the closing price on Monday will be used to consider the income, once the closing price is announced, the stock market will suspend trading until the opening of the next day, so repurchase of stocks will have to wait until the opening of the next day. So, the purchase price is also the opening price of the next day. Record the profit plus the principal after the closing of the week n-1 as , so, the profit plus the principal of the week n is . (is the closing price of one stock at a Monday of week n+1, is the opening price of one stock at a Tuesday of week n)

Q1 requires to evaluate its investment effectiveness referring to the Greater Bay Index. So, we regarding the Greater Bay Area Index as a stock, its opening price on Tuesday of the specified week is , and the closing price on Monday of the next week is , and its return rate is , the rate of return of the 10 selected stocks is . If the rate of return of our investment strategy in a certain week is higher than the rate corresponding to the Greater Bay Area Index, it means that our rate of return of this week is higher than the market average. It shows that our investment strategy has a well performance this week. Calculate all the yield data from 2011 to 2020, and put the yield curve of ten stocks and the yield curve corresponding to the Greater Bay Area index on the same plane. If the yield curve of ten stocks is always or in a larger period of time above the yield curve corresponding to the Greater Bay Area Index, which indicates that our investment strategy has achieved better investment results , the model has certain practicability.

Program optimization

More volatility means more risk, so with equal returns or roughly equal returns, we tend to prefer to buy stocks with flatter stock prices. Therefore, we introduce the factor of standard deviation of stock price. Let the trading volume of a day in a week be $T!$ The trading volume is $t!$, then the trading volume for a week is 【】 , the trading volume is 【】 , and the average trading price of this stock for that day is $T!$ / $t!$, and note that the average stock price for a week is 【】 , then the stock's The standard deviation is 【】 .

We use a fractional system to select the strongest ten stocks, i.e., we allocate an appropriate weight in front of each stock's return and standard deviation to make it the basis for our judgment of dominant stocks. However, due to the large order of magnitude difference between the two, a direct and arbitrary allocation of the weights will inevitably lead to large errors. Therefore, we choose to adopt a randomly selected one-week sample as the reference method to set the weights.

The method is as follows: first select a sample of 30 stocks in the second week, solve and rank the top ten stocks by their stock codes 1,2,3,...,10. The top ten stocks are selected and their stock symbols are 1,2,3,...,10. The returns of these ten stocks in the first week are $i\$$, $i/$,$i0$..., $i\$\%$ and the variance is $\sigma\$$, $\sigma\$$, Let the weights of the return amount and standard deviation be $\omega\$$,$\omega/$, respectively. Then it should satisfy

Binding conditions

Arbitrarily select the coordinate points in the closed area, the horizontal coordinate indicates $\omega\$$ and the vertical coordinate indicates $\omega/$. The formula used to determine the strong stocks at this time is $\omega\$i! + \omega/\sigma!$, and substitute the data of the first week to find the score that The top ten stock codes are obtained, and then the top ten stock codes are obtained by finding the top 30 stock returns in the second week, and then the top ten stock codes are compared with the stock codes obtained from the first week's data analysis, and the number of duplicate codes is recorded, thus the comparison is calculated continuously until the weekly data from 2011 to 2020 are compared, and the total number of duplicate codes is recorded. Then, we select other coordinates in the feasible domain and perform the comparison again. Finally, the most suitable coordinate point with the highest number of duplicate codes was found.

However, all nine lines pass through the origin, so they cannot form a closed region. At the same time, our team did not come up with other more reasonable lines that would form a closed region with the nine lines already drawn. So in the end, our group's strategy for selecting strong stocks (scoring method) is still based on the average of last week's trading day returns.

2.2 Analysis of Question 2

The investment amount of a single stock can be varied flexibly, then the investment amount of each stock can be determined based on the scores of the ten best stocks. Let the score of a stock be

s!, then the investment weight of that stock is 【】 therefore the investment amount of a stock is 【】 (m is the principal amount invested for a given week). The optimal yield curve for this investment strategy is 【】 (o is the opening price on Tuesday of a given week and c is the closing price on Monday of the following week). However, there may be cases where the score is negative. In case of a negative score, 99% of the principal is invested in the stock with a positive score and 1% of the principal is invested in the stock with a negative score. (For values with negative scores, after taking the absolute value, the values are sorted from highest to lowest. e.g.: If the scores are: 3, 1, -1, -2, -5, then they should be transformed to: 3, 1; 5, 2, 1, with 99% of the total investment amount for the first few stocks and 1% of the total investment amount for the last few stocks)

2.3 Analysis of Question 3

Since the timing of the transfer remains the same, and the number of stocks invested and the percentage of each stock invested remain unchanged from question 1, the question does not indicate that the ten strongest stocks of the previous week should be selected. Therefore, the change in our investment strategy in this question compared to question 1 is to change the reference from last week's data to the previous 30 days' data (or all previous data if less than 30 days) to select the strongest ten stocks. After analyzing a sample of several stocks' movements, we found that in most cases, the movement of a stock after a certain point in time is most influenced by the previous 30 days or so. A too-long time span results in some factors, which have less impact on the current trend, being taken into account equally. And similarly, a too-short time span results in some stock price fluctuations caused by consecutive time periods not being fully taken into account. By choosing 30 days as the maximum time horizon to be considered, the accuracy of the forecast can be effectively improved.

2.4 Analysis of Question 4

Referring to the previous 30 days (or all previous data if less than 30 days), select the ten strongest stocks using the way in question one and invest in the ten stocks. The score of each stock is calculated in the way of question two, and the amount of investment is determined by the percentage of the score, and the time of purchase is referred to the 37% theory. The actual opening time of a business day is 4 hours, and the total opening time of the first three days is 12 hours, so the time point corresponding to the 37% theory is around 10:20 a.m. on Tuesday. Therefore, the lowest price of a stock before 10:20 on Tuesday is used as the reference price, and once the price of the stock is lower than the reference price after that, it will be bought quickly. If the score of the stock is higher than 10%, the stock will be purchased at 10% of the total assets, or if there are unpurchased shares until the last minute, directly at the price at which it will last close. If the position is still less than 50%, the position should be exactly equal to 50% after the purchase is completed, and the weighting of the purchased stocks should be consistent with the score weighting. The total market opening time for a week is 20 hours, so the time point corresponding to the 37% theory is Tuesday at 2:40 p.m. Therefore, after completing the position, the reference price at this time is the lowest price as of Tuesday at 2:40 p.m. If a price lower than the reference price appears again afterwards and the position is not full, buy according to the ratio, and if there is still a position left at the end, buy the stock with the highest score directly.

III. Explanation of symbols

SYMBOL	MEANING
	the principal amount of the stock acquired
	the opening price of a stock
	the closing price of a stock
i	the daily return of a specified stock
	the average of the specified stock's returns
	the trading volume of a specified stock on a given day
	trading volume of the specified stock on a given day
	average price of the specified stock for a week
	the standard deviation of the stock price of a given stock
	weight
	amount of stock purchased after fees
	the score of a stock

IV. Data Pre-processing

For Appendix I with 30 stock quotes data provided by the topic, we use pandas to read and perform preliminary processing. First, the data are grouped by stock code and indexed by time. Then, the row labels are resampled by week. Since the data given in the topic includes multiple stocks of SZSE and SSE, and the effective trading days of each stock are different, the data are rebuilt using a uniform time series, i.e., the actual trading day series of the two stock exchanges from 2011 to 2020 for indexing the data. The missing data are replaced by the mean of the known data before and after. From the investment strategy, it is known that the closing price on Monday and the opening price on Tuesday of each trading week are needed to determine whether a single week in the resampled series is a complete trading week and to extract data for the dates that meet the requirements.

The Appendix II, which contains Greater Bay Area Index ticker data, provided in the topic is considered as a portfolio in the same format as the model results. An index reconstruction operation similar to the one above is performed on it to ensure that the data are all in the same dimension in the same format.

After considering the computing power of the equipment, the time cost and the need of the model, we decided not to use the minute quotes data provided in the title, as we believe that this data set has limited effect on the overall model.

V. Development and Solution to the Model

5.1 Answer to Question 1

5.1.1 Development of Model 1

Let the actual amount of the stock purchased after deducting the commission be , the principal be , the closing price on a given day be and the opening price on that day be .

Number of shares of this stock purchased on a given day:

Return for that stock on a given day is:

The average rate of return for a given stock is:

The actual amount of shares purchased for a stock is:

The average amount divided to each stock is:

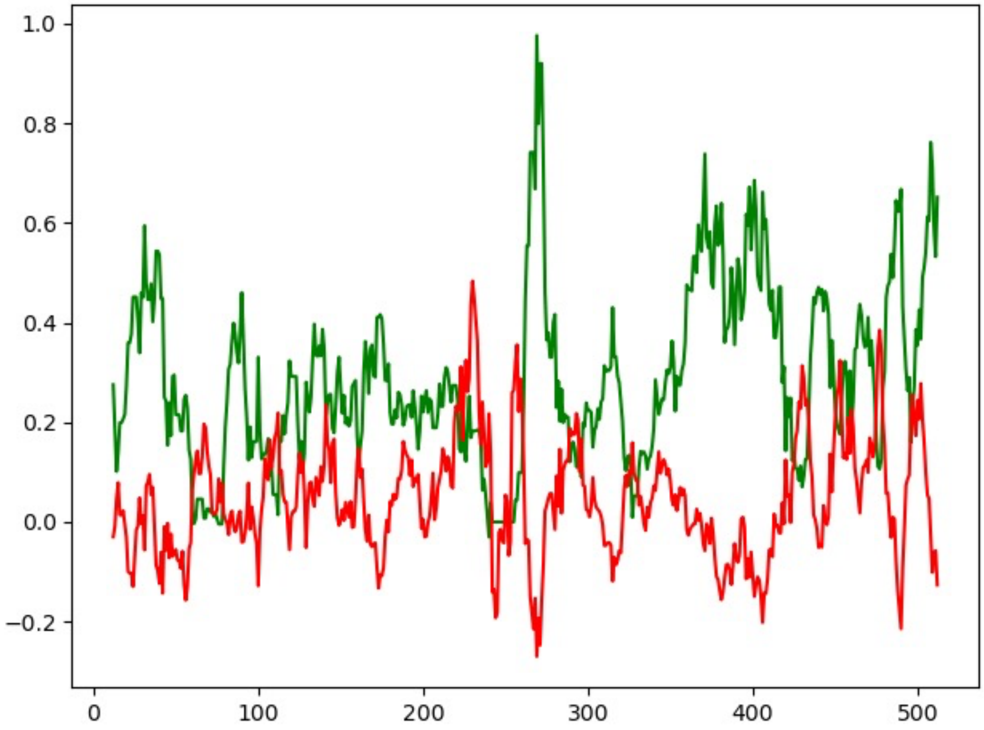
The sum of the proceeds and the principal amount generated by the purchased stock is:

The rate of return on a stock is:

The rate of return on the ten stocks is:

5.1.2 Solution of Model 1

Based on the data given in the question, substitute the above model and use python to calculate and process it, the following is obtained:



Graphical analysis: The green line is the yield curve of the investment portfolio generated by the model, the red line is the yield curve based on the Greater Bay Area Index, and the x-axis of the graph is a time series with the week in which January 3, 2011 is located as the 0 point, on a weekly basis, to the week in which October 30, 2020 is located.

It is clear from the graph that the model is generally able to outperform the reference index.

Moreover, the model is able to obtain higher returns in times of financial crisis and poor macroeconomic conditions such as trade frictions between the US and China.

5.2 Answer to Question 2

5.2.1 Development of Model 2

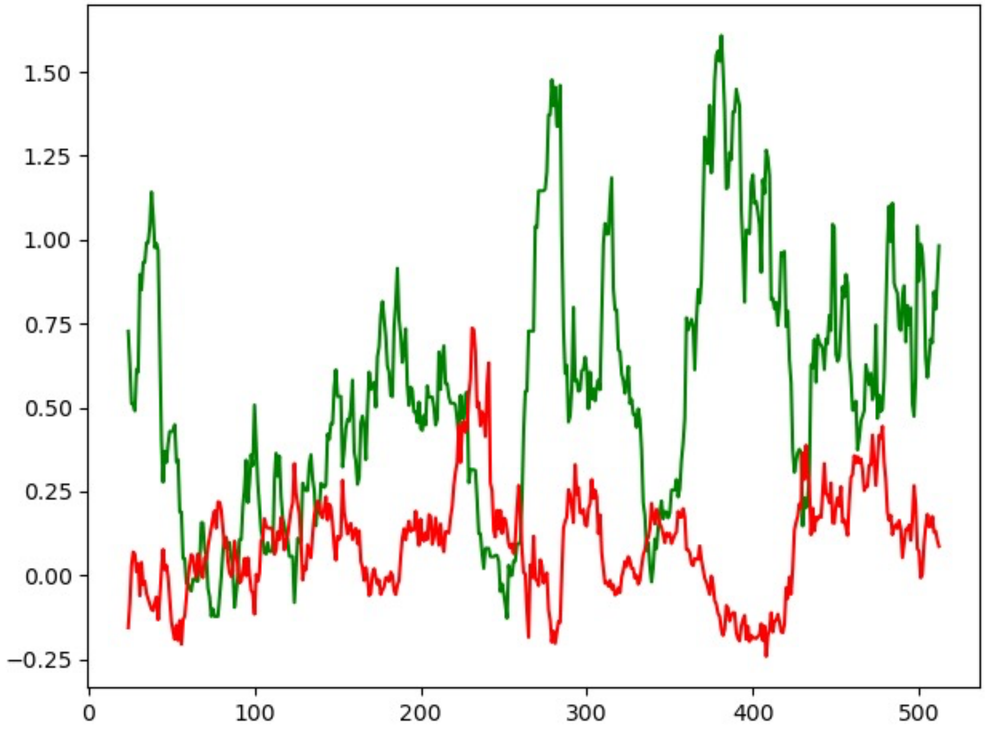
Assume that the score of a stock is:

And the amount invested in a particular stock is:

The optimal rate of return of this investment strategy is:

5.2.2 Solution of Model 2

Based on the data given in the question, substitute the above model and use python to calculate and process it, the following is obtained:



Graphical analysis: The green line is the yield curve of the investment portfolio generated by the model, the red line is the yield curve based on the Greater Bay Area Index, and the x-axis of the graph is a time series with the week in which January 3, 2011 is located as the 0 point, on a weekly basis, to the week in which October 30, 2020 is located.

The investment strategy remains highly effective during periods of stock market instability, such as the financial crisis and the U.S.-China trade friction. The investment return is significantly higher than that of Model 1.

5.3 Solution of Question 3

5.3.1 Development of Model 3

Set a total score of a stock is equal to

The amount invested in the stock is:

The optimal rate of return of this investment strategy is:

VI. Model evaluation and promotion

Advantages of the model:

1. rich parameter consideration and high fault tolerance.
2. The model generates portfolios that generally outperform the reference index.

3. In several tests with historical data, the model was able to obtain higher returns in poor macroeconomic times.
4. The model requires less computational resources to perform its calculations

Disadvantages of the model:

1. The depth of data processing is not sufficient.
2. Standard deviation is not introduced, which makes the model riskier.

Extensions of the model:

The model is highly extensible, and more parameters can be considered on this basis in devices with higher computational power.

VII. Reference

[1] Sergio M. Focardi, Frank J. Fabozzi, The Mathematics of Financial Modeling and Investment Management, CRUP, ISBN 978-7-300-12544-2

Appendix:

Code:

1）

```
import pandas as pd
from tqdm import tqdm
import numpy as np
import matplotlib.pyplot as plt

HANDLING_FEE = 2.5e-4


stock = pd.read_csv('第一届湾区数模赛题及数据/A题/附录一：30支股票行情.csv', header=0,
index_col=1, parse_dates=['time'])
codes = stock['code'].unique()

stock_g = stock.groupby("code")

timeindex = list(stock_g.get_group(codes[0])['open'].resample('W').groups.keys())


gb_index = pd.read_csv('第一届湾区数模赛题及数据/A题/附录二：大湾区指数行情.csv',
header=0, index_col=1, parse_dates=['time'])

trend = []

for code in tqdm(codes):
    pers = []
    op = stock_g.get_group(code)['open'].resample('W')
    cl = stock_g.get_group(code)['close'].resample('W')
    for week in timeindex:
        try:
            bias = cl.get_group(week)/op.get_group(week) - 1
            pers.append(bias.mean())
        except KeyError:
            pers.append(pers[-1])
    trend.append(pers)

trend = pd.DataFrame(trend)
```

```
week_top10 = []
for i in tqdm(range(trend.shape[1])):
    week_top10.append(pd.DataFrame.sort_values(trend, i).index[:10])

principal = np.zeros(trend.shape[1])
principal[0] = 1

buy_p = stock[stock.index.weekday == 1].resample('W')['code', 'open']
sell_p = stock[stock.index.weekday == 0].resample('W')['code', 'close']

for n, week in tqdm(enumerate(week_top10[:-1])):
    try:
        bp = buy_p.get_group(timeindex[n])
        bp.index = bp['code']
        bp = bp['open']
        sp = sell_p.get_group(timeindex[n+1])
        sp.index = sp['code']
        sp = sp['close']
    except Exception:
        principal[n + 1] = principal[n]
    else:
        for sc in week:
            try:
                principal[n+1] += ((principal[n] * 0.1)/sp[codes[sc]]) * bp[codes[sc]]
            except Exception:
                principal[n+1] = principal[n]
                break

yield_curve = pd.Series(principal).pct_change(periods=12)


gb_principal = np.zeros(trend.shape[1])
gb_principal[0] = 1

buy_p = gb_index[gb_index.index.weekday == 1].resample('W')['open']
sell_p = gb_index[gb_index.index.weekday == 0].resample('W')['close']

for week in tqdm(range(trend.shape[1]-1)):
    try:
        gb_principal[week+1] = gb_principal[week]*sell_p.get_group(timeindex[week+1])[0] /
```

```
buy_p.get_group(timeindex[week])[0]
        except Exception:
                gb_principal[week + 1] = gb_principal[week]


gb_yield_curve = pd.Series(gb_principal).pct_change(periods=12)

plt.plot(range(513), yield_curve, color='green')
plt.plot(range(513), gb_yield_curve, color='red')
plt.show()
```

2）

```
import pandas as pd
from tqdm import tqdm
import numpy as np
import matplotlib.pyplot as plt

HANDLING_FEE = 2.5e-4


stock = pd.read_csv('第一届湾区数模赛题及数据/A 题/附录一 :30 支股票行情.csv', header=0,
index_col=1, parse_dates=['time'])
codes = stock['code'].unique()

stock_g = stock.groupby("code")

timeindex = list(stock_g.get_group(codes[0])['open'].resample('W').groups.keys())


gb_index = pd.read_csv('第一届湾区数模赛题及数据/A 题/附录二：大湾区指数行情.csv',
header=0, index_col=1, parse_dates=['time'])
trend = []


for code in tqdm(codes):
        pers = []
        op = stock_g.get_group(code)['open'].resample('W')
        cl = stock_g.get_group(code)['close'].resample('W')
        for week in timeindex:
```

```
        try:
            bias = cl.get_group(week)/op.get_group(week) - 1
            pers.append(bias.mean())
        except KeyError:
            pers.append(pers[-1])
    trend.append(pers)

trend = pd.DataFrame(trend)
trend.index = codes
investment_per = pd.DataFrame(np.zeros([codes.shape[0], trend.shape[1]]))
investment_per.index = codes.T
week_top10 = []

for i in tqdm(range(trend.shape[1])):
    t = pd.DataFrame.sort_values(trend, i)[i][:10]
    week_top10.append(t.index)
    p_tmp = t[t > 0]
    if p_tmp.shape[0]:
        pp = 1 - 0.01 * p_tmp.shape[0]
        for code in t.index:
            if trend.at[code, i] > 0:
                investment_per.at[code, i] = (p_tmp[code] / p_tmp.sum()) * pp
            else:
                investment_per.at[code, i] = 0.01
    else:
        n_tmp = (pd.Series.sort_values(t)).abs()
        n_tmp.index = n_tmp.index[::-1]
        for code in t.index:
            investment_per.at[code, i] = (n_tmp[code] / n_tmp.sum())
    investment_per[i].fillna(0)

principal = np.zeros(trend.shape[1])
principal[0] = 1

buy_p = stock[stock.index.weekday == 1].resample('W')['code', 'open']
sell_p = stock[stock.index.weekday == 0].resample('W')['code', 'close']

for n, week in tqdm(enumerate(week_top10[:-1])):
    try:
        bp = buy_p.get_group(timeindex[n])
        bp.index = bp['code']
        bp = bp['open']
        sp = sell_p.get_group(timeindex[n+1])
        sp.index = sp['code']
        sp = sp['close']
    except Exception:
        principal[n + 1] = principal[n]
    else:
        for sc in week:
            try:
                principal[n+1] += ((principal[n] * investment_per.at[sc, n])/sp[sc]) * bp[sc]
            except Exception:
                principal[n+1] = principal[n]
                break

yield_curve = pd.Series(principal).pct_change(periods=24)

gb_principal = np.zeros(trend.shape[1])
gb_principal[0] = 1

buy_p = gb_index[gb_index.index.weekday == 1].resample('W')['open']
sell_p = gb_index[gb_index.index.weekday == 0].resample('W')['close']

for week in tqdm(range(trend.shape[1]-1)):
    try:
        gb_principal[week+1] = gb_principal[week]*sell_p.get_group(timeindex[week+1])[0] /
buy_p.get_group(timeindex[week])[0]
    except Exception:
        gb_principal[week + 1] = gb_principal[week]


gb_yield_curve = pd.Series(gb_principal).pct_change(periods=24)

plt.plot(range(513), yield_curve, color='green')
plt.plot(range(513), gb_yield_curve, color='red')
plt.show()
```

3)

```
import pandas as pd
```

```
from tqdm import tqdm
import numpy as np
import matplotlib.pyplot as plt

HANDLING_FEE = 2.5e-4


stock = pd.read_csv('第一届湾区数模赛题及数据/A题/附录一：30支股票行情.csv', header=0,
index_col=1, parse_dates=['time'])
codes = stock['code'].unique()

stock_g = stock.groupby("code")


gb_index = pd.read_csv('第一届湾区数模赛题及数据/A题/附录二：大湾区指数行情.csv',
header=0, index_col=1, parse_dates=['time'])
trend = []

for code in tqdm(codes):
    pers = []
    bias = stock_g.get_group(code)['close']/stock_g.get_group(code)['open'] - 1
    trend.append(bias.pct_change(periods=30))


trend = pd.DataFrame(trend)
trend.index = codes

timeindex = list(stock_g.get_group(codes[0])['open'].resample('W').groups.keys())

bp = trend[trend.columns[trend.columns.weekday == 1]].replace([np.inf, -np.inf], np.nan).fillna(0)
sp = trend[trend.columns[trend.columns.weekday == 0]].replace([np.inf, -np.inf], np.nan).fillna(0)

investment_per = pd.DataFrame(np.zeros([codes.shape[0], trend.shape[1]]))
investment_per.index = codes.T
week_top10 = []

for i in tqdm(range(sp.shape[1])):
    try:
        t = pd.DataFrame.sort_values(sp, sp.columns[i])[sp.columns[i]][:10]
        week_top10.append(t.index)
        p_tmp = t[t > 0]
        if p_tmp.shape[0]:
            pp = 1 - 0.01 * p_tmp.shape[0]
            for code in t.index:
                if trend.at[code, i] > 0:
                    investment_per.at[code, i] = (p_tmp[code] / p_tmp.sum()) * pp
                else:
                    investment_per.at[code, i] = 0.01
        else:
            n_tmp = (pd.Series.sort_values(t)).abs()
            n_tmp.index = n_tmp.index[::-1]
            for code in t.index:
                investment_per.at[code, i] = (n_tmp[code] / n_tmp.sum())
        investment_per[i].fillna(0)
    except Exception:
        pass

principal = np.zeros(bp.shape[1])
principal[0] = 1

buy_p = (stock[stock.index.weekday == 1])[['code', 'open']].groupby('code')[['open']]
sell_p = (stock[stock.index.weekday == 0])[['code', 'close']].groupby('code')[['close']]

for n, week in tqdm(enumerate(week_top10[:-1])):
    for sc in week:
        try:
            sp = sell_p.get_group(sc).reset_index(drop=True)
            bp = buy_p.get_group(sc).reset_index(drop=True)
            principal[n+1] += ((principal[n] * 0.1)/sp.at[n, 'close']) * bp.at[n, 'open']     #
investment_per.at[sc, n]
        except Exception:
            principal[n+1] = principal[n]
            break

yield_curve = pd.Series(principal).fillna(0).pct_change(periods=24)

gb_principal = np.zeros(481)
gb_principal[0] = 1

buy_p = gb_index[gb_index.index.weekday == 1].resample('W')['open']
```

```
sell_p = gb_index[gb_index.index.weekday == 0].resample('W')['close']

for week in tqdm(range(480)):
        try:
                gb_principal[week+1] = gb_principal[week]*sell_p.get_group(timeindex[week+1])[0] /
buy_p.get_group(timeindex[week])[0]
        except Exception:
                gb_principal[week + 1] = gb_principal[week]

gb_yield_curve = pd.Series(gb_principal).fillna(0).pct_change(periods=24)
```

PLATFORM:

Microsoft® Word 2019 MSO (16.0.13231.20110) 32bit
Base on Windows 10 Home 17763.1339

PyCharm 2020.2.3 (Professional Edition)
Python 3.8.5 with Anaconda 1.7.2
Included pandas 1.1.3
    numpy 1.19.2
    matplotlib 3.3.2
    tqdm 4.50.2
Base on MacOS Catalina 10.15.7