SMU
SINGAPORE MANAGEMENT UNIVERSITY

FNCE6004 – Advanced Portfolio Management

Question2 Sector Rotation Strategy

G1-Group 2

Group Members

Yan Shuhang	01401490
Yu Yedi	01388394
Cao Siqi	01396406
Lin Huixin	01384289
Huang Shan	01402026

1. Strategy

In this paper we investigate whether it is possible to earn excess return in ETF market using our sector rotation strategy. In general, our sector rotation strategy is to buy the ETFs based on the information of the last year. And we have designed four methods to perform our sector rotation strategy:

Method 1: Every year, buying ETFs with the **highest holding period return** last year.

Method 2: Every year, buying ETFs with the **lowest holding period return** last year.

Method 3: Every year, buying ETFs with the **highest trading volume** last year.

Method 4: Every year, buying ETFs with the **lowest trading volume** last year.

Later, we will use SPY and other sector ETFs data from 2009 to 2020 to test our rotation strategy.

2. Data

From etf.com, we draft a list of sector ETFs under 3 conditions:

Conditions	**Requirements**
Geography	U.S.
Asset Class	Equity, specified sector
Asset under management	Larger than 3 billion dollars

There are 27 ETFs satisfying the conditions. Then, we collect the 11-year data (2009-2019) of these ETFs from WRDS, and remove the ETFs that omit data. As a result, we get 23 ETFs, which information are displayed in **Appendix 1.**

At the beginning, our test will use only ten ETFs, so we choose 10 ETFs with the highest AUM, showing below:

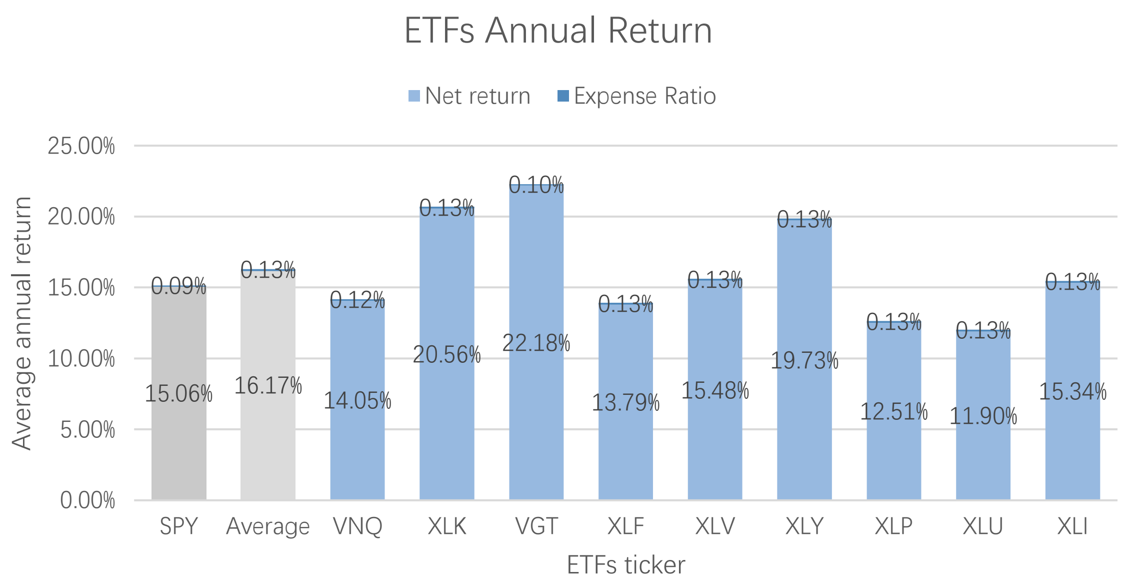


Figure 1： ETFs Annual Return

It can be noticed that the selected ETFs' net average annual returns (16.17%) are close to SPY's (15.06%), indicating our selection is rational. The relevant table can be found in **Appendix 2**

In addition, to evaluate the portfolio, we also collect U.S. 30-day T-bill's HPR from 2010 to 2019 as risk-free rate, displaying **in Appendix 3.**

3. Implementation

For the empirical tests, we code by python, in which the most important functions are **rotation** and **measure.**

def rotation (data, numetf, num, ascend, fee):….

The rotation () function can implement our sector rotation strategy, with 4 input variables: **data** can be the ETFs' returns or volume determining the information based to choose ETFs; **numetf** means the number of ETF under watch; **num** means number of ETF bought equally every year; **ascend** determinants choosing max or min of past; **fee** means whether considering the ETF's expense. And the function returns to a list of our portfolios and its returns annually.

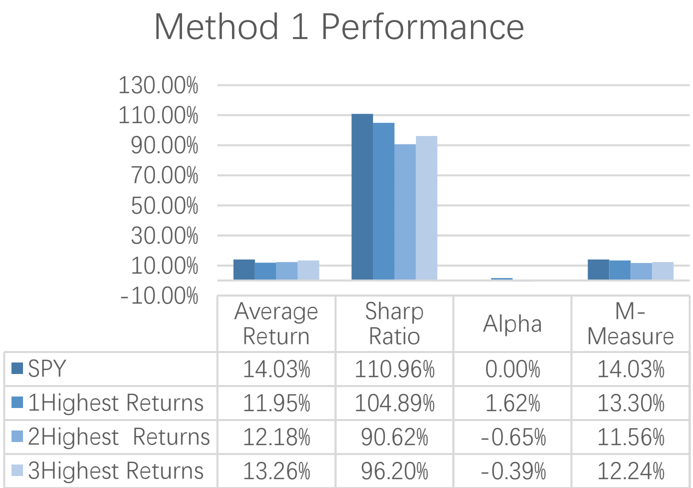
def measure (returns, RFR, BM):….

The measure () function can evaluate the performance of our portfolios, with 3 inputs: **returns** means the portfolios' return list; **RFR** is the risk-free rate (30-day T-bill annual returns); **BM** is the benchmark returns (SPY's returns) And the function returns to 4 indicators: Average Return, Sharpe Ratio, Alpha, M-Square

The complete codes can be found in **Appendix 4** or download from **GitHub:** https://github.com/MikeYan8080/Sector-Rotation-Strategy.

4. Results and Analysis

Method 1: Every year, buying 1/2/3 ETFs equally with the **highest holding period return** in the last year.



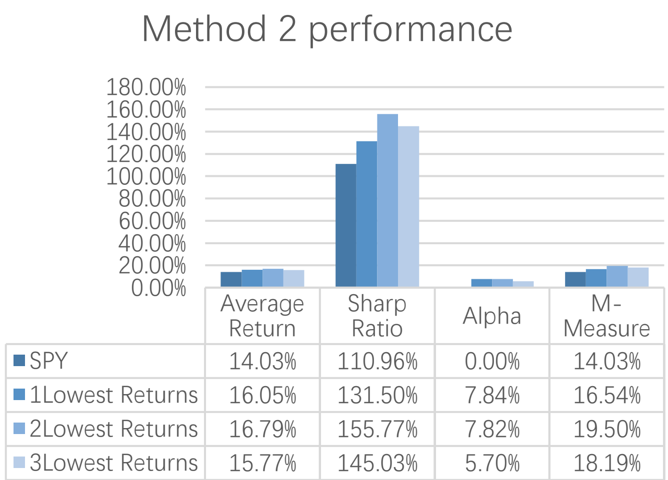
	Average Return	Sharp Ratio	Alpha	M-Measure
SPY	14.03%	110.96%	0.00%	14.03%
1Highest Returns	11.95%	104.89%	1.62%	13.30%
2Highest Returns	12.18%	90.62%	-0.65%	11.56%
3Highest Returns	13.26%	96.20%	-0.39%	12.24%

Figure 2: Method 1 Performance

It can be noticed that only in the Alpha measure, one of our portfolios (buying 1 highest) can beat the SPY. Purchasing 1 ETFs does better than the other two, but its performance is worse than the SPY's in other 3 measures.

Totally, the results indicate that the methods 1 is unlikely to earn an excess return.

Method 2: Every year, buying 1/2/3 ETFs equally with the **lowest holding period return** in the last year.



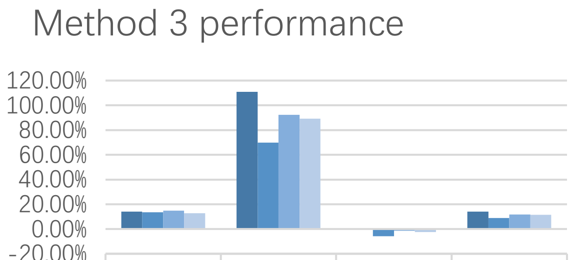
	Average Return	Sharp Ratio	Alpha	M-Measure
SPY	14.03%	110.96%	0.00%	14.03%
1Lowest Returns	16.05%	131.50%	7.84%	16.54%
2Lowest Returns	16.79%	155.77%	7.82%	19.50%
3Lowest Returns	15.77%	145.03%	5.70%	18.19%

Figure 3: Method 2 Performance

The results for Method 1 show that there are some negative alphas, which leads us to believe that a contrarian strategy might be more profitable, so we conduct the Method 2.

The results are amazing. All portfolios beat the markets in 4 measures. It is worth mentioning that the Alphas of portfolios are all over 5%. And among them, the Buying 2 lowest ETFs works highest in all version.

Method 3: Every year, buying 1/2/3 ETFs equally with the **highest trading volume** in the last year.



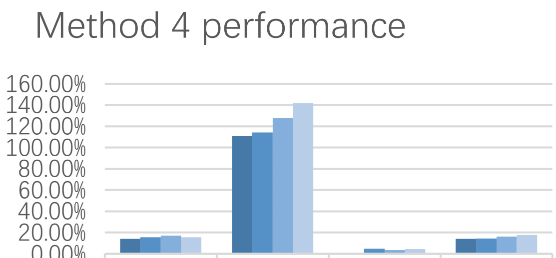
	Average Return	Sharp Ratio	Alpha	M-Measure
SPY	14.03%	110.96%	0.00%	14.03%
1Highest Volume	13.43%	69.78%	-5.74%	9.02%
2Highest Volume	14.85%	92.31%	-1.52%	11.76%
3Highest Volume	12.89%	89.20%	-2.39%	11.38%

Figure 4: Method 3 Performance

Obviously, the methods 3 has a very poor performance.

Among 3 portfolios, Buying 2 Highest Volume performs highest. Even though Buying 2 Highest Volume's average return surpass the SPY's, it takes much more risk so that its Sharp Ration, Alpha and M-Measures look very poor, comparing to the market.

Method 4: Every year, buying 1/2/3 ETFs equally with the **lowest trading volume** in the last year.



	Average Return	Sharp Ratio	Alpha	M-Measure
SPY	14.03%	110.96%	0.00%	14.03%
1Lowest Volume	15.72%	114.10%	4.69%	14.42%
2Lowest Volume	16.97%	127.82%	3.54%	16.09%
3Lowest Volume	15.47%	141.96%	4.40%	17.81%

Figure 5: Method 4 Performance

Similarly, if we conduct a contrarian strategy of Method 3, the results will be totally different.

Just like the Method2, the Method 4 does good job with all portfolio defeating the market. Among them, we think the Buying 3 lowest volume is the best one with highest Sharp Ratio and M-square.

In conclusion, our rotation strategy can lead to an extra return in some methods (Buying ETFs with lowest return/volume last year), while in other methods, the strategy doesn't work well.

5. Sensitivity Analysis

In the former tests, the methods 2 and methods 4 significantly outperform SPY in all measures. It seems that it can prove the rotation strategy really work. But we still doubt it, because it is possible that the extra return is not due to the rotation strategy, but to the initial ETF selection, as we only use 10 optional ETFs in tests. Furthermore, to better justify the rotation strategy, we do a sensitivity analysis on methods 2 (Buying 2 lowest return) and methods 4(Buying 3 lowest

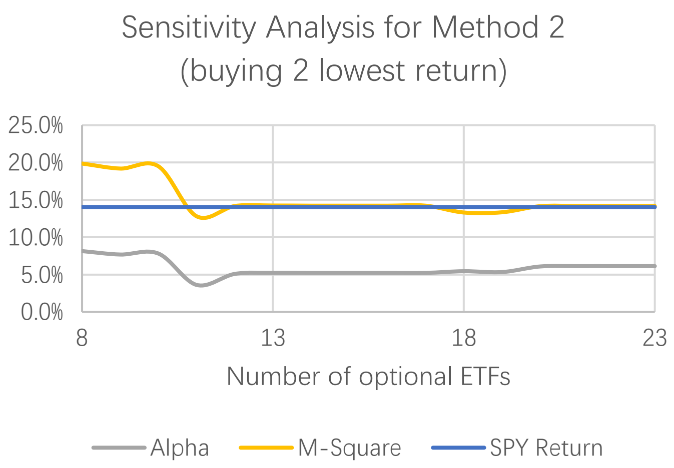


Figure 6: Sensitivity Analysis for Method 2

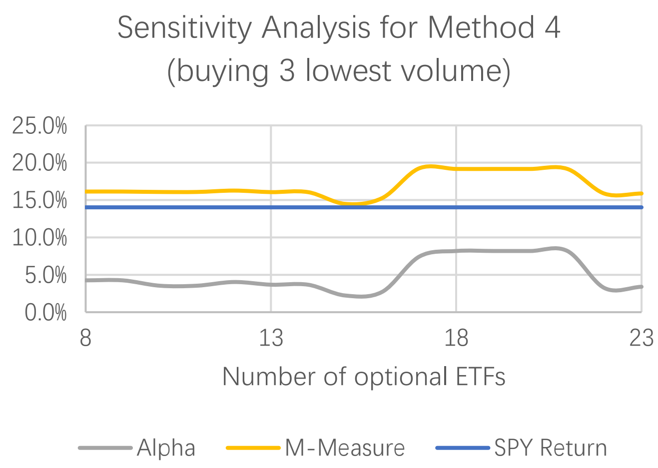


Figure 7: Sensitivity Analysis for Method 4

From the picture above, we can notice that both of Method 2 and Method 4 have positive Alpha whatever the number of optional ETFs. And Method 4 always has a higher M-square than SPY's return. As for Method 2, in some conditions, it cannot beat the market in M-square measure. Therefore, we can safely say that Method 4 can outperform the market (SPY).

6. Recommendation and Conclusion

For a US investor with a risk aversion of 6, we can calculate his utility if he chooses our best rotation strategy (Method4 Buying 3 Lowest volume), worst rotation strategy (Method1 Buying 2 Highest return) or choose the SPY ETF:

	SPY	Best Rotation Portfolio	Worst Rotation Portfolio
Utility	9.43	12.05	7.19

Based on our investigation, this investor can benefit more from sector rotation if he uses the best method (Method4 Buying 3 Lowest volume), whereas the worst rotation method (Method1 Buying 2 Highest return) cannot benefit him more than SPY. So, we recommend the investor should switch to invest fully in our best rotation strategy.

In conclusion, rotation strategy does work in some situations (Buying ETFs with lowest return/volume last year), while in other situations (Buying ETFs with highest return/volume last year), the strategy doesn't.

We think there are two reasons why lowest return/volume ETFs can earn more in next year:

1. It follows the business cycle. When the returns or volume touch the bottom, it will bounce back.

2. The bad performances of ETFs urge their managers to take more actions to make up the lost.

Appendix

1. ETF initially selected

Ticker	Name	Segment	Issuer	Expense Ratio	AUM
SPY	SPDR S&P 500 ETF Trust	- Large Cap	State Street Global Advisors	0.09%	312.13
VNQ	Vanguard Real Estate ETF	Real Estate	Vanguard	0.12%	38.23
XLK	Technology Select Sector SPDR Fund	Technology	State Street Global Advisors	0.13%	28.2
VGT	Vanguard Information Technology ETF	Technology	Vanguard	0.10%	28.09
XLF	Financial Select Sector SPDR Fund	Financials	State Street Global Advisors	0.13%	24.3
XLV	Health Care Select Sector SPDR Fund	Health Care	State Street Global Advisors	0.13%	20.18
XLY	Consumer Discretionary Select Sector Fund	Consumer Cyclicals	State Street Global Advisors	0.13%	15.53
XLP	Consumer Staples Select Sector SPDR Fund	Consumer Non-cyclicals	State Street Global Advisors	0.13%	13.48
XLU	Utilities Select Sector SPDR Fund	Utilities	State Street Global Advisors	0.13%	11.79
XLI	Industrial Select Sector SPDR Fund	Industrials	State Street Global Advisors	0.13%	11.59
VHT	Vanguard Health Care ETF	Health Care	Vanguard	0.10%	10.14
XLE	Energy Select Sector SPDR Fund	Energy	State Street Global Advisors	0.13%	9.18
FDN	First Trust Dow Jones Internet Index Fund	Internet	First Trust	0.52%	8.4
VFH	Vanguard Financials ETF	Financials	Vanguard	0.10%	7.99
IBB	iShares NASDAQ Biotechnology ETF	Biotech	Blackrock	0.47%	7.37
ITA	iShares U.S. Aerospace & Defense ETF	Aerospace & Defense	Blackrock	0.42%	5.65
IYW	iShares U.S. Technology ETF	Technology	Blackrock	0.42%	5.19
IHI	iShares U.S. Medical Devices ETF	Health Care Equipment	Blackrock	0.43%	5.01
VPU	Vanguard Utilities ETF	Utilities	Vanguard	0.10%	4.74
IYR	iShares U.S. Real Estate ETF	Real Estate	Blackrock	0.42%	4.41
XBI	SPDR S&P Biotech ETF	Biotech	State Street Global Advisors	0.35%	4.28
XLB	Materials Select Sector SPDR Fund	Basic Materials	State Street Global Advisors	0.13%	4.18
VIS	Vanguard Industrials ETF	Industrials	Vanguard	0.10%	3.6
QTEC	First Trust NAS-100 TechSector Index Fund	Technology	First Trust	0.57%	3.05

Figure 8: ETF selected

2. ETFs' returns

Figure 9: ETFs' Returns

Ticker	Annual Return	Expense Ratio	Net return
SPY	15.15%	0.09%	15.06%
Average	16.30%	0.13%	16.17%
VNQ	14.17%	0.12%	14.05%
XLK	20.69%	0.13%	20.56%
VGT	22.28%	0.10%	22.18%
XLF	13.92%	0.13%	13.79%
XLV	15.61%	0.13%	15.48%
XLY	19.86%	0.13%	19.73%
XLP	12.64%	0.13%	12.51%
XLU	12.03%	0.13%	11.90%
XLI	15.47%	0.13%	15.34%

3. 30-day T-bill annual returns

Figure 10: 30-day T-bill Annual Returns

Calendar Date	30 Day Bill Returns
20091231	0.097%
20101231	0.122%
20111230	0.043%
20121231	0.057%
20131231	0.028%
20141231	0.016%
20151231	0.009%
20161230	0.189%
20171229	0.791%
20181231	1.707%
20191231	2.147%

4. Program (python)

```
import pandas as pd
import numpy as np
from scipy import stats
from pyecharts.charts import Bar, Page
from pyecharts import options as opts

# import the initial ETF return data
data0 = pd.read_excel(r'C:\Users\严书航\Desktop\PM Project\ETF returns volume.xlsx')

# convert the date to Year & Month
```

```
data0.rename(columns={'Names Date': 'YearMonth'}, inplace=True)
data0.YearMonth = data0.YearMonth.map(lambda x: x//100)

# transpose the data to (time. etf)table show the monthly return
data1 = data0.pivot(index='YearMonth', columns='Ticker Symbol', values='Returns')

# drop the ETF does not have historical data from Jan 2009
data2 = data1.dropna(axis=1, how='any')

# Calculate the yearly return for each ETF
data3 = data2.reset_index()
data3.YearMonth = data3.YearMonth.map(lambda x: x//100)
data3 = data3.groupby('YearMonth').apply(lambda x: np.prod(x+1)-1)
data3.drop(columns='YearMonth', inplace=True)

# Output data to excel
# with pd.ExcelWriter(r'C:\Users\严书航\Desktop\PM Project\ETFs_Return.xlsx') as writer:
#     data2.to_excel(writer, sheet_name='Monthly Return')
#     data3.to_excel(writer, sheet_name='Yearly Return')

# Change the order of the data3
etfdata = pd.read_excel(r'C:\Users\严书航\Desktop\PM Project\PM ETF Chosen.xlsx',
sheet_name='ETF selected')
etfdata.set_index(etfdata.Ticker, inplace=True)
data3 = data3[list(etfdata.Ticker)]


# transpose the data3 and drop the benchmark('SPY')
data4 = data3.drop(columns='SPY', inplace=False)
data4 = data4.T

# transpose the data to (time. etf)table show the monthly Volume
data11 = data0.pivot(index='YearMonth', columns='Ticker Symbol', values='Volume')

# drop the ETF does not have historical data from Jan 2009
data22 = data11.dropna(axis=1, how='any')

# Calculate the yearly Volume for each ETF
data33 = data22.reset_index()
data33.YearMonth = data33.YearMonth.map(lambda x: x//100)
data33 = data33.groupby('YearMonth').apply(lambda x: np.sum(x))
data33.drop(columns='YearMonth', inplace=True)
data33 = data33[list(etfdata.Ticker)]

# Output data to excel
# with pd.ExcelWriter(r'C:\Users\严书航\Desktop\PM Project\ETFs_Return.xlsx') as writer:
#     data22.to_excel(writer, sheet_name='Monthly Volume')
#     data33.to_excel(writer, sheet_name='Yearly Volume' )

# transpose the data3 and drop the benchmark('SPY')
data44 = data33.drop(columns='SPY', inplace=False)
data44 = data44.T
```

```
# sector rotation strestege, numETF means number of ETF under watch, num means number of ETF bought yearly
# ascend means the strestege choosing max past or min past, Fee means considerring the ETF's expense
def rotation(data, numetf=10, num=2, ascend=False, fee=False):
    data5 = data.drop(columns=2019, inplace=False)
    data5 = data5[0: numetf]
    etf_choose = []
    for i in data5:
        data6 = data5[i].sort_values(ascending=ascend)
        for x in range(num):
            nyr = data4.loc[data6.index[x], i+1]
            if fee:
                nyr -= etfdata.loc[data6.index[x], 'Expense Ratio']
            item1 = [i+1, data6.index[x], data6[x], nyr]

            etf_choose.append(item1)

    data7 = pd.DataFrame(etf_choose, columns=['Year', 'Ticker', 'Last-year Return', 'Return'])
    # print(data7)

    # with pd.ExcelWriter(r'C:\Users\严书航\Desktop\Portfolios.xlsx', mode='a') as writer:
    #       data7.to_excel(writer)
    return data8['Return']


# SPY's   Returns from 2010 to 2019
spyR = data3.drop(2009).SPY
# SPY's   Returns discount the expense from 2010 to 2019
spyRD = spyR - etfdata.loc['SPY', 'Expense Ratio']


# risk free rates come form 30 day T-bill form 2010 to 2019
risk_free_rate = [0.001216, 0.000428, 0.000565, 0.000277, 0.000164, 0.000092, 0.001886, 0.007914, 0.017066, 0.02147]


# Function to calculate : Average Return, Sharp Ratio, Alpha, M-Measure
def measure(returns, RFR=risk_free_rate, BM=spyR):
    AR = np.mean(returns)
    SR = np.mean(returns-RFR)/np.std(returns-RFR, ddof=1)
    Alpha = stats.linregress(x=(BM-risk_free_rate), y=(returns-risk_free_rate)).intercept
    MS = SR*np.std(BM-risk_free_rate, ddof=1) + np.mean(risk_free_rate)
    return [AR, SR, Alpha, MS]


# Function to show the results comparing to Bench Mark
def result(data, ascends, names, fees=True):
    table = pd.DataFrame(measure(spyR),
                         index=['Average Return', 'Sharp Ratio', 'Alpha', 'M-Measure'],
                         columns=['SPY-Bench Mark'])
    for x in range(3):
        name = str(x+1) + names
        table[name] = measure(rotation(data, num=x+1, ascend=ascends, fee=fees))
    return table
```

```
# rotation based on return, picked the last year best 1,2,3
a = result(data4, False, 'Highest Returns')
# rotation based on return, picked the last year worst 1,2,3
b = result(data4, True, 'Lowest Returns')
# rotation based on volume, picked the last year best 1,2,3
c = result(data44, False, 'Highest Volume')
# rotation based on volume, picked the last year best 1,2,3
d = result(data44, True, 'Lowest Volume')


page = Page(layout=Page.SimplePageLayout, page_title='Projection about Portfolio strategy')
for i, x in enumerate([a, b, c, d]):
    bar = Bar(init_opts=opts.InitOpts(width='600px', height='350px'))
    bar.add_xaxis(list(x.index))
    # x.applymap(lambda t: format(t, '.2%'))
    for j in x:
        bar.add_yaxis(j, ['{:.3f}'.format(t) for t in list(x[j])])
    bar.set_global_opts(title_opts=opts.TitleOpts(title="results", pos_top='top'),
                        legend_opts=opts.LegendOpts(pos_top='10%'))
    bar.set_series_opts(label_opts=opts.LabelOpts(is_show=False))
    page.add(bar)
    # with pd.ExcelWriter(r'C:\Users\严书航\Desktop\Results.xlsx', mode='a') as writer:
    #     x.to_excel(writer, sheet_name=str(i))

page.render(r'C:\Users\严书航\Desktop\Results.html')


# sensitivity analysis
table2 = pd.DataFrame(measure(spyR),
                      index=['Average Return', 'Sharp Ratio', 'Alpha', 'M-Measure'],
                      columns=['SPY-Bench Mark'])

table3 = pd.DataFrame(measure(spyR),
                      index=['Average Return', 'Sharp Ratio', 'Alpha', 'M-Measure'],
                      columns=['SPY-Bench Mark'])

for n in range(8, 24):
    m = str(n)
    table2[m] = measure(rotation(data4, numetf=n, num=2, ascend=True, fee=True))

for n in range(8, 24):
    m = str(n)
    table3[m] = measure(rotation(data44, numetf=n, num=3, ascend=True, fee=True))

# with pd.ExcelWriter(r'C:\Users\严书航\Desktop\Results.xlsx', mode='a') as writer:
#     table2.to_excel(writer, sheet_name='method 2')
#     table3.to_excel(writer, sheet_name='method 4')

# calculate utility
return_method4 = rotation(data44, numetf=10, num=3, ascend=True, fee=True)
```

```
def utility(returns, Ra=6):
    u = np.mean(returns)*100 - 0.005*Ra*np.var(returns, ddof=1)*10000

    return u


u_spy = utility(spyRD)
u_m4 = utility(return_method4)

print(u_m4)
print(u_spy)
print(utility(rotation(data4, numetf=10, num=2, ascend=False, fee=True)))
```