Implementation of the Equity Index Rebalance Strategy

Alternative Investment

Zhong Zeng

May 6th, 2017

1. Introduction

In this project, I implemented the strategy of equity index rebalance. I used Russell 2000 as my index. The first part is introduction. In the second part, I will describe the data source, universe, and how I cleaned the data. In the third part, I will introduce stock indices, particularly Russell 2000, and why I chose Russell 2000. In the fourth part, I will explain my strategy and how I implement this strategy. In the fifth part, I will evaluate this strategy, including its return, volatility, Sharp ratio, and risk matrices. In the sixth part, I will discuss the caveats of my implementation and future extensions of this strategy. In the seventh part, I will make trade recommendation based on my strategy implementation and evaluation. I will also explore different timing and portfolio construction methods in the seventh part. In the appendix, I have attached the Python code that was used to clean the data, run the strategy, calculate evaluation criteria, and generate the graphs.

2. Data

2.1 Data Source

I collected data from Compustat. I accessed the Compustat database through Wharton Research Database Services (WRDS). The specific data source is called Compustat Monthly Updates - Security Daily. The unique ID is CUSIP[1].

I selected all common and ordinary stocks the headquarters of which are in the United States of America, by setting two conditions of TPCI = 0[2] and ISO Country Code – Headquarters = USA[3].

[1] CUSIP stands for Committee on Uniform Securities Identification Procedures. A CUSIP number identifies most financial instruments, including: stocks of all registered U.S. and Canadian companies, commercial paper, and U.S. government and municipal bonds.

By specifying the condition of TPCI = 0, I filtered out other securities, including Preferred, preference, Warrant or right, Convertible bond, Unit - an issue that is comprised of a combination of common shares and warrants, Municipal dollar or taxable bond, Corporate or government bond, Mutual or investment trust fund, and Certificate.

2.2 Stock Universe
The stock universe is the entire common and ordinary stocks whose headquarters in the United States of America. There are 7,531 stocks observed on May 2, 2016. The total number of stocks may vary across time.

2.3 Time Period
The time period is from Jan 1, 2006 to Dec 31, 2016. The time frequency is daily.

2.4 Data Cleaning
I didn't filter out or drop any data in the stage of collecting data. I chose to do it this way for two reasons. First, the data from WRDS is relatively clean. Second, I chose to filter abnormal data and outliers in the later stage instead. When calculating daily prices and returns of my portfolio, I filtered out prices deviating too much from historical level and droped all daily returns that are either larger than 0.5 or smaller than -0.5. There are hundreds of stocks in my portfolio and no rebalancing of my portfolio, so it is reasonable to believe the value of the portfolio couldn't change for more than 50% in a day.

2.5 Interpolation
I used forward interpolation for missing data.

3. Stock Indices

3.1 Stock Indices
A stock index is a measurement of the value of a section of the stock market. It is computed from the prices of selected stocks. The stocks are selected into an index based on certain criteria. However, the composition of an index (either the selected stocks or their weights) changes over time.

3.2 Russell 2000
I choose Russell 2000 as my index for the index rebalancing strategy.

---

[2] This item contains the code that identifies the type of security an issue is. 0 means meaning common and ordinary stocks.

[3] Variable Name: loc. ISO Country Code – Headquarters. This item contains the code that identifies the country where the company headquarters is located. The country codes are established by the International Standards Organization (ISO).

The index is maintained by FTSE Russell. The Russell 2000 Index is a small-cap stock market index of the bottom 2,000 stocks in the Russell 3000 Index. The Russell 2000 Index consists of stocks whose market capitalizations (market cap) are ranked from 1000 to 3000 among all stocks whose company are incorporated in the United States.

The Russell 2000 is reconstituted every year. The ranks of the market caps are calculated at the end of May every year. However, the actual reconstitution is schedule at the end June every year.

I choose Russell 2000 for the following two reasons. First, the Russell 2000 is by far the most common benchmark for mutual funds that identify themselves as "small-cap", while the S&P 500 index is mostly used for large capitalization stocks. Second, the Russell 2000 and also other Russell indices have relatively easy, clear, and transparent selection criteria and calculation methods. They are basically chosen according to their market cap, which equals the stock price times the number of outstanding shares.

4 The Index Rebalance Strategy

4.1 Reconstitution Effects

Empirical results and academic studies have shown the following phenomenon for Russell 2000. The prices of the stocks that are to be added into Russell 2000 tend to go up. On the country, the prices of the stocks that are to be deleted into Russell 2000 tend to go down.

4.2 Economic Intuitions

There are two economic intuitions of this phenomenon. On the one hand, index membership itself has value because it is associated with permanent changes in liquidity, information flows, or both. Because inclusion in an index is usually associated with a permanent increase in trading volume and liquidity, inclusion lowers future trading costs and, therefore, permanently increases intrinsic value. On the other hand, reconstitution effects could be explainable in terms of temporary price pressure[4].

4.3 Entering and Existing Signal

As I have discussed, the prices of the stocks that are to be added into Russell 2000 tend to go up. On the country, the prices of the stocks that are to be deleted into Russell 2000 tend to go down. The calculation of market cap and the determination of components of new Russell 2000 are at the end of May. The actual rebalance of Russell

[4] Madhavan, A. (2003). The Russell Reconstitution Effect. *Financial Analysts Journal*, 59(4), 51-64. Retrieved from http://www.jstor.org/stable/4480496

2000 is at the end of June. There is a window between the end of May and the end of June. Following is the timeline of Russell 2000.

Date	Events
End of May	Russell is selected based on the close price on that day.
Beginning of June	The preliminary list of index additions and deletions is released.
Last Friday of June	Reconstituted index becomes effective.
Beginning of July	The final membership list becomes available.

Chart 4.3.1 Russell 2000 Timeline

I will enter before this window. In the refinement part, I will try different timings, including entering before and during this window. The entering signal is whether the stocks are to be added to or deleted from Russell 2000 or remain unchanged. Before the index rebalance, I long the stocks that are to be added into an index, and short the stocks that are to be deleted from an index.

I make my portfolio equally weighted for simplicity. In the refinement part, I will discuss different weighting methods.

Then, I will hold the portfolio for a certain period and then liquidate all of my positions. I will try different exiting dates in the refinement part.

4.3 Timing and Date Convention

There is very large timing risk in this strategy. If I enter into a position too earlier before the end of May, I may not predict the final components of Russell 2000. Nevertheless, I will not make much money or even loss money if I enter into a position too late after the end of May. It's because other investor would have acted before us and driven up or down the stocks' prices. In this implementation, I choose to enter into the position on May 1st, hold the portfolio until July 10th, and liquidate all my positions on July 10th. If May 1st is not a trading day, I move to the following calendar day until I reach a trading day. If July 10th is not a trading day, I move backward to the previous calendar day until I reach a trading day. As mentioned above, I will try different timings in the refinement part.

5. Evaluation

5.1 Return, Volatility, and Sharp Ratio

The return and Sharp ratio of this strategy are not good. The possible reason is that this index rebalance strategy has been well studied and explored. Thus, the alpha is not significant. Following is the evaluation criteria of my strategy.

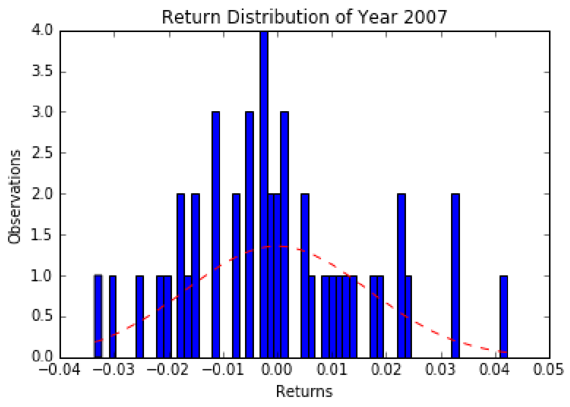
Time Period	Average Return	Standard Deviation	Sharp Ratio	5% Value at Risk	10% Value at Risk	5% Conditional Value at Risk	10% Conditional Value at Risk
2006 - 2016	-1.38934	0.788262	-1.7879	-0.08016	-0.05604	-0.13998	-0.10274

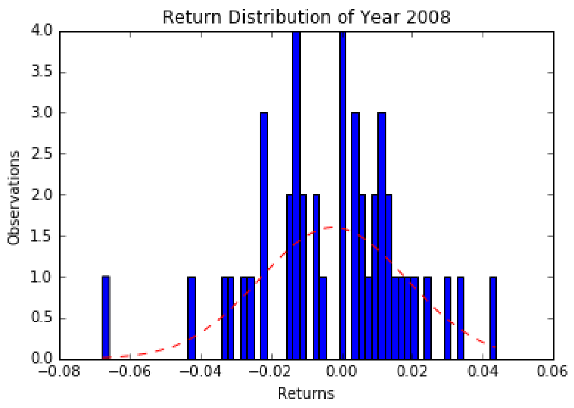
Chart 5.1 Evaluation of the Strategy

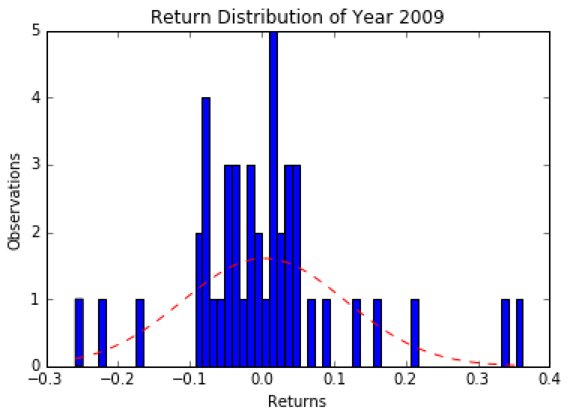
5.2 Distribution of Percentage Returns

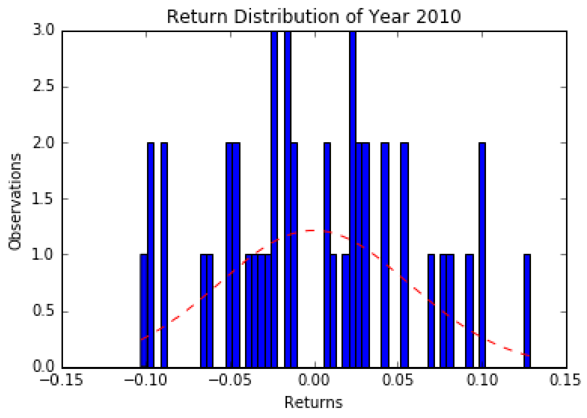
I have fitted normal distribution to the daily profit and loss. Following are the graphs of distribution of daily profit and loss. The histograms are the real distribution of the daily percentage returns. The dash lines are the fitted normal distributions.

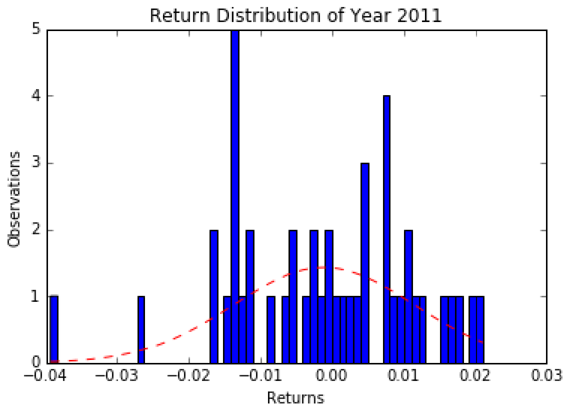
From the graphs, I can see that the normal distribution doesn't fit the return of this strategy very well. The real data has fatter tail. Investors should realize that the data violates the normal distribution assumption.

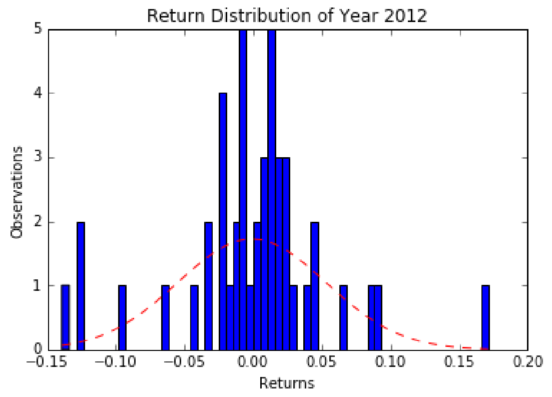


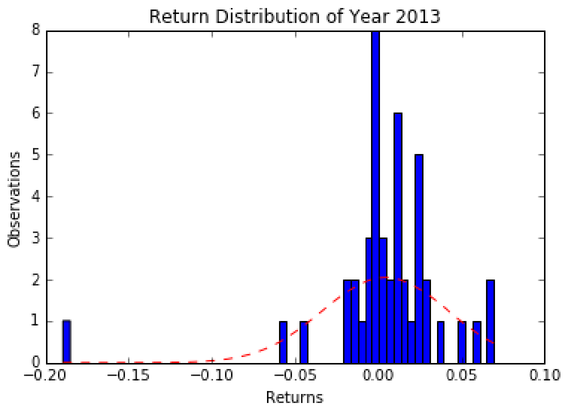


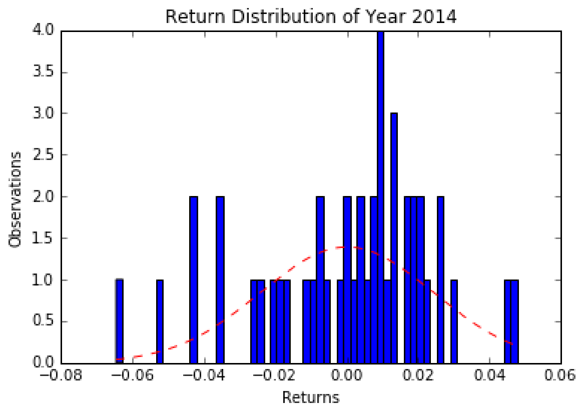

Return Distribution of Year 2009
Observations
Returns

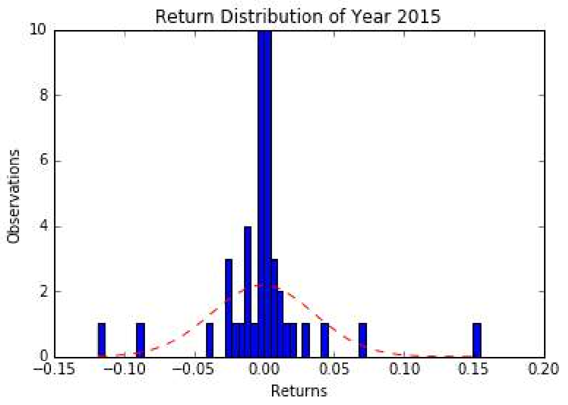

Return Distribution of Year 2010
Observations
Returns

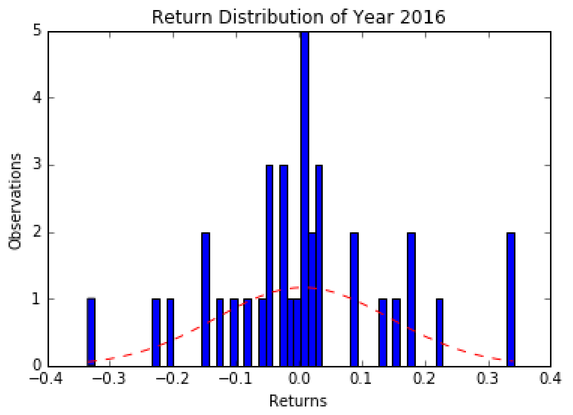

Return Distribution of Year 2011
Observations
Returns

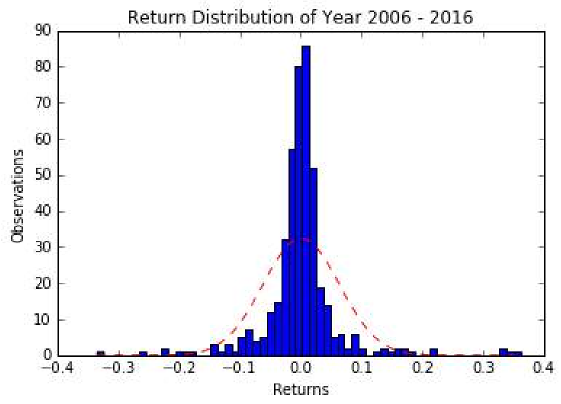

Return Distribution of Year 2012
Observations
Returns


Return Distribution of Year 2013
Observations
Returns


Return Distribution of Year 2014
Observations
Returns

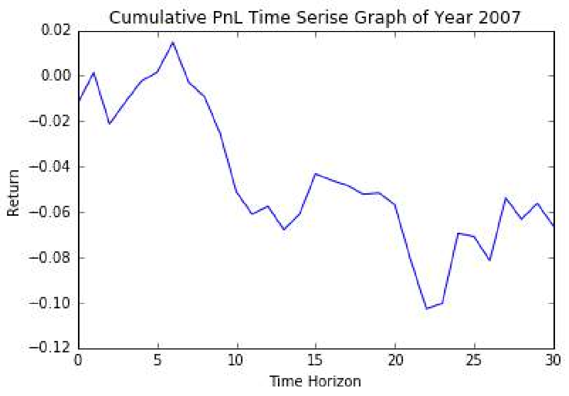

Return Distribution of Year 2015
Observations
Returns

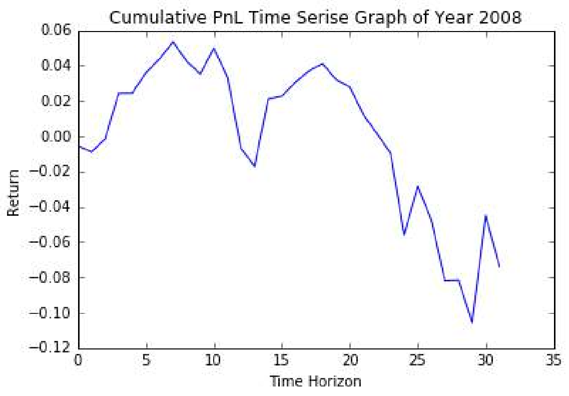

Return Distribution of Year 2016
Observations
Returns

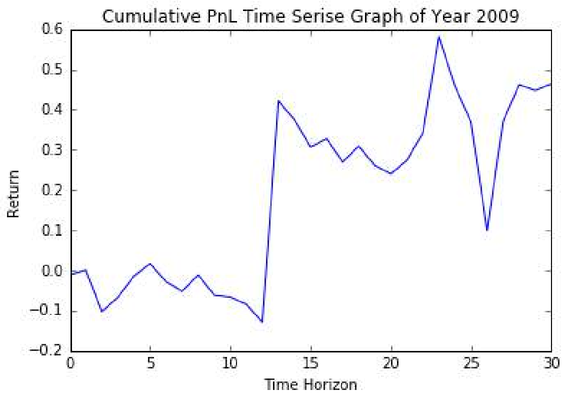


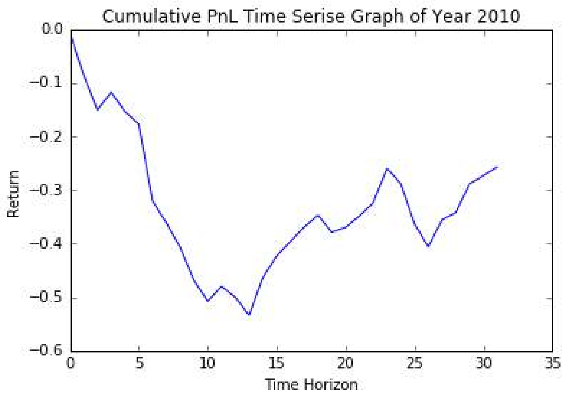
5.3 Time Series of Cumulative Simple Profit and Loss

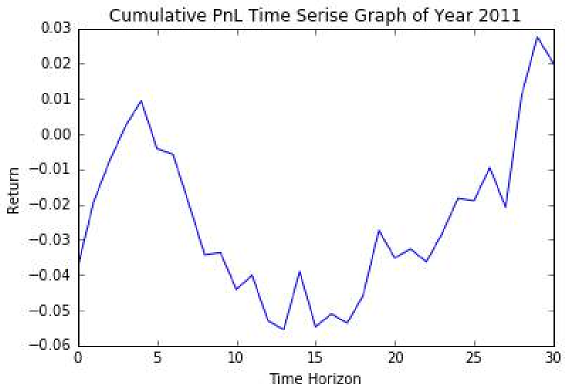
Follow are the graphs of Time Series of Cumulative Simple Profit and Loss. The time series of cumulative simple profit and loss vary year from year. Sometimes, it doesn't even look good on the graph. I recommend not to implement this strategy without serious consideration.

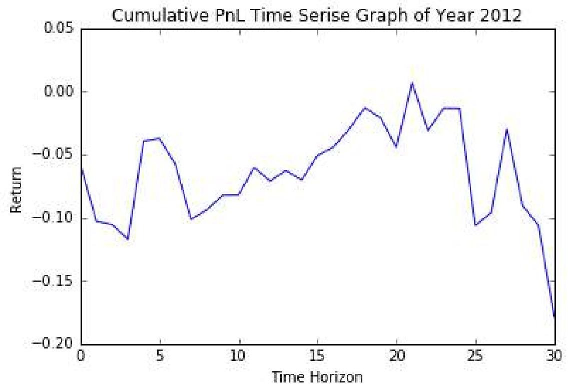


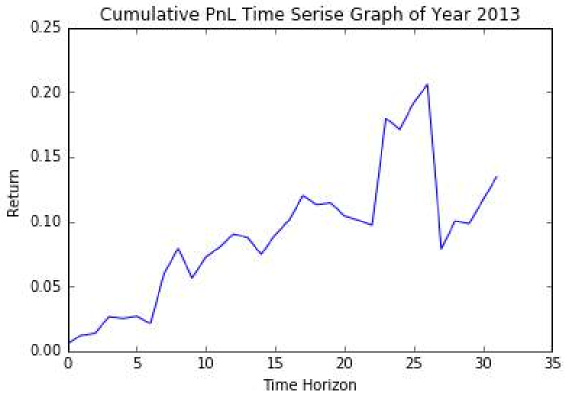


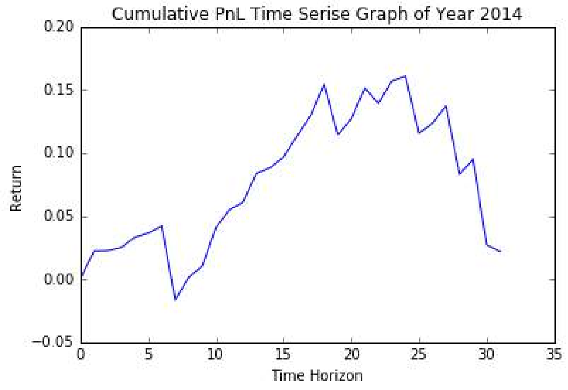


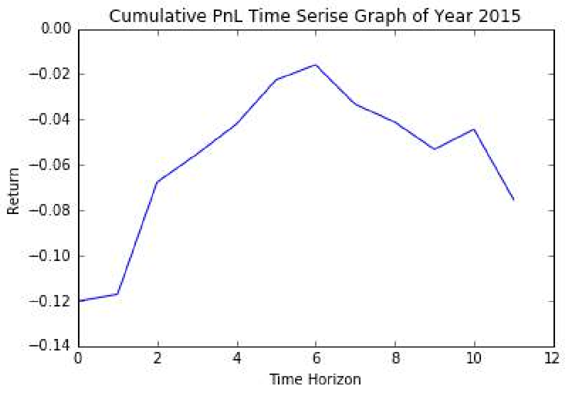


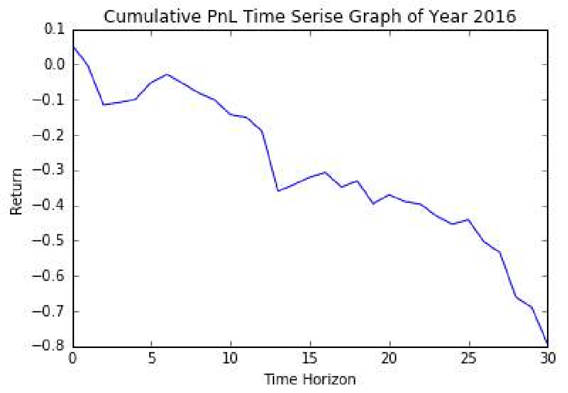

Cumulative PnL Time Serise Graph of Year 2011
Return
Time Horizon

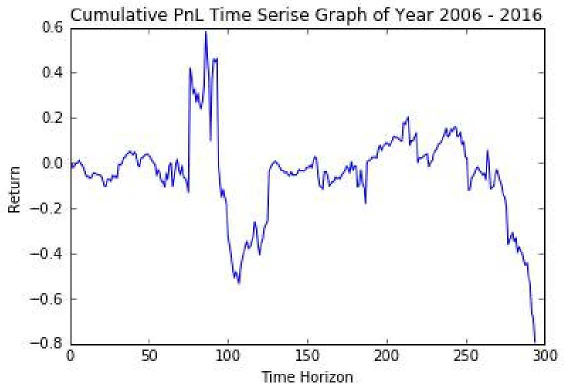

Cumulative PnL Time Serise Graph of Year 2012
Return
Time Horizon


Cumulative PnL Time Serise Graph of Year 2013
Return
Time Horizon


Cumulative PnL Time Serise Graph of Year 2014
Return
Time Horizon


Cumulative PnL Time Serise Graph of Year 2015
Return
Time Horizon


Cumulative PnL Time Serise Graph of Year 2016
Return
Time Horizon


Cumulative PnL Time Serise Graph of Year 2006 - 2016
Return
Time Horizon

5.4 Value at Risk (VaR) and Conditional Valuation at Risk (CVaR)

The 5% VaR is -0.26902. The 5% CVaR is -0.33947. The 10% VaR is -0.23154. The 10% CVaR is -0.29318. The distance between 5% VaR and 5% CVaR are larger than that between 10% VaR and 10% CVaR.

As I can also see from the graph of the Distribution of Percentage Return, the normal distribution doesn't fit the data well. In addition, the data is not stationary. The performance varies year from year. Therefore, I should be cautious when I use VaR and CVaR to assess the risk.

5.5 Short Selling Constrains

In the implementation, I assume that I can short any amount of any stocks without any cost. This may not be true in the real world.

5.6 Crowdedness

This strategy has been highly researched. Many investment managers use this strategy. It is very crowded. This doesn't lead us to expect alphas from this strategy.

6. Possible Future Extension

6.1 Data

In this implementation, I calculate the components of Russell 2000 myself. Further research could use the actual Russell 2000 components.

6.2 Transaction Cost

There are transaction costs associated with this strategy, including bid-ask spread and commissions. Future research could take transaction cost into account.

6.3 Other Indices

Russell 2000 is a widely traded index for small cap stocks. As I can see from my implementation and evaluation, the performance of the index rebalance strategy using Russell 2000 is not good. Further research could also be done with respect to other indices in the other markets.

7. Trade Recommendation

7.1 Timing

I have tried the following combinations of entering date and exiting date.

Entering Date	Existing Date
May 1st	July 10th

May 1st	Aug 30th
May 20th	July 10th
May 20th	Aug 30th
May 31st	July 10th
May 31st	July 20th

Chart 6.1.1 Combinations of Different Timings

None of these combinations yield a decent return or Sharp ratio. The alpha of this strategy has been worn out. I encourage investors to try different indices, instead of trying different dates on Russell 2000.

7.2 Portfolio Construction

I have used equally weighted portfolio in the previous implementation, where I long or short the dollar value of each stock.

I now explore the approach of constructing the portfolio by market capitalization weighting. In this approach, I make the weights of the stocks in my portfolio proportional to the weights of the stocks in Russell 2000.

The reason why market cap weighted is better than equally weighted is as following. The larger the portion of a stock in Russell 2000, the more this stock is to be traded, and the more attention this stock receives from investors. Thus, this stock's price will increase more than other stocks which have less weight in Russell 2000. Vice versa, the stocks to be deleted from Russell 2000 works in the opposite way. The larger the portion of a stock in previous Russell 2000, the more this stock's price is going to decrease.

In order to make the weights of the stocks in my portfolio proportional to the weights of the stocks in Russell 2000, I do the following derivation.

In Russell 2000, I have the following relationship among stock A, stock B, and etc.

$$\begin{aligned} Weight_A : Weight_B : \dots &= Mkt\ Cap_A :\ Mkt\ Cap_B : \dots \\ &= (Price_A \times \#outstding_A) : (Price_B \times outstding_B) : \dots \end{aligned}$$

where $Mkt\ Cap$ is market capitalization, $\#\ outstding$ is the number of outstanding shares.

Also, I have the following relationship by definition.

$$Weight_A : Weight_B : \dots = (Price_A : Price_B : \dots) \times (\#shares : \#shares_B : \dots)$$

Combining these two equations, I can reach the following conclusion.

$$\#shares_A : \#shares_B : \dots = \#outstding_A :\ \#outstding_B : \dots$$

where *#shares* is the number of shares in Russell 2000 or my portfolio.

Thus, I make the number of shares of stocks in my portfolio proportional to their number of outstanding shares.

I now show the result from the approach of constructing the portfolio by market capitalization weighting. The entering date is May 20th, while the existing date is July 10th. Following is the graph of the distribution of interday return.

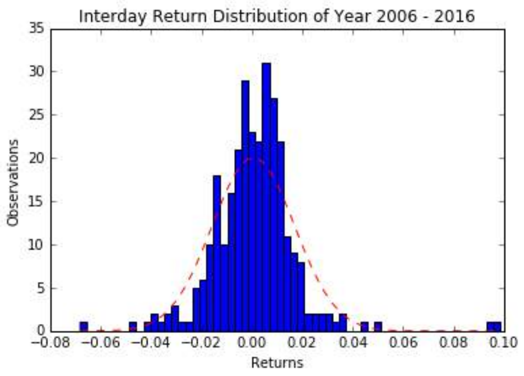


Chart 7.1 Distribution of Interday Returns

Following is the graph of the time series of cumulative simple profit and loss.

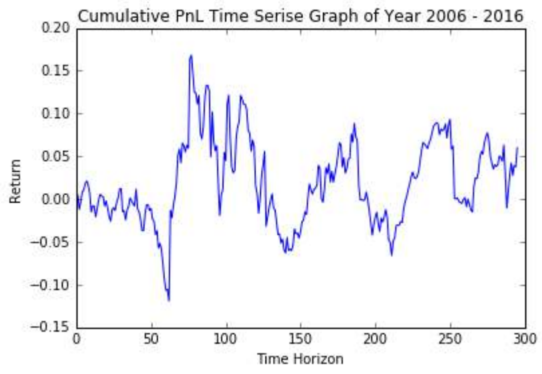


Chart 7.2 Time Series of Cumulative Simple Profit and Loss

Following is a chart of evaluation performance of market cap weighted portfolio of index rebalance strategy.

Time Period	Average Return	Standard Deviation	Sharp Ratio	5% Value at Risk	10% Value at Risk	5% Conditional Value at Risk	10% Conditional Value at Risk
2007	0.043547	0.136953	0.171938	-0.01459	-0.01067	-0.01933	-0.01708

2008	-0.92263	0.181037	-5.20683	-0.02053	-0.01824	-0.02185	-0.02058
2009	0.413834	0.431684	0.912319	-0.02863	-0.01914	-0.05362	-0.03663
2010	0.814174	0.433301	1.832846	-0.03877	-0.0292	-0.04484	-0.03872
2011	-0.1511	0.195582	-0.8748	-0.0225	-0.01527	-0.03045	-0.02552
2012	0.469775	0.220678	2.03815	-0.02243	-0.01702	-0.02925	-0.02627
2013	-0.18598	0.177362	-1.16137	-0.01619	-0.01516	-0.02181	-0.01874
2014	0.52005	0.111459	4.486396	-0.0132	-0.00538	-0.01567	-0.0122
2015	-0.01805	0.10086	-0.37728	-0.0084	-0.00646	-0.01073	-0.00861
2016	0.442081	0.233503	1.807605	-0.02502	-0.01511	-0.0352	-0.02942
Overall	0.152632	0.258604	0.512877	-0.02245	-0.0162	-0.03498	-0.027

Chart 7.3 Evaluation of the Strategy

The chart shows more satisfying results than the equal weighted portfolio. Nevertheless, there are negative returns observed in some years. And the Sharp ratio is relative low, which is around 0.51. I still don't recommend this strategy.

7.3 Conclusion

I recommend investors to search for alpha in new indices other than Russell 2000, since my backtesting suggests negative returns.

Appendix

```
# -*- coding: utf-8 -*-
"""
Created on Fri Apr 21 11:19:28 2017

@author: Zhong Zeng
"""

import pandas as pd
import numpy as np
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt

def aDate(sdata, yr2, mth2, dt2):
    # return the dataframe of stocks at a specified date
    # Input variable: data, year, month, date
    # Type: dataframe, int, int, int
    date2 = yr2*1e4+mth*1e2+dt2
    #yr2 = int( date2 / 1e4)
    #dt2 = int( date2 % 1e2)
    #mth2 = int (( date2 % 1e4) /1e2)
    while True:
        date2 = yr2*1e4+mth*1e2+dt2
        sd2=sdata[sdata['datadate']==date2]
        if(sd2.empty==False):
            break
        if(dt2<31):
            dt2=dt2+1 # move to the following calendar day
        else:
            dt2=1
            if(mth2<12):
                mth2=mth2+1 # move to the following calendar month
            else:
                mth2=1
                yr2=yr2+1 # move to the following calendar year
    return [int(date2), sd2]
```

```
def formPort( sdata, adds, buy, weighting):
    # Select certain stocks to formulate a equally weighted portfolio
    # Input variable: data, list of stocks to be added/deleted, 1/-1 for buy/sell
    # Type: dataframe, dataframe, 1/-1
    if (weighting=='equal'):
        w='prccd'
    elif(weighting=='mktcp'):
        w='cshoc'

    sdata_add=sdata[bd-1<sdata['datadate']][sdata['datadate']<ed][sdata['cusip'].isin(adds['cusip'])]
    sdata_add.interpolate(method='linear', limit_direction='forward') # Interpolate Data
    # form a dataframe of holding # of shares
    sdata_add_h = sdata_add[sdata_add['datadate']==bd][[w,'cusip']]
    sdata_add_h = sdata_add_h.rename(columns = {w:'h'})

    if (weighting=='equal'):
        sdata_add_h['h'] = buy * 1e4 / sdata_add_h['h']
    elif(weighting=='mktcp'):
        sdata_add_h['h'] = sdata_add_h['h'] # / 1e4

    # the holding number of shares; join doesn't work
    sdata_add = pd.merge(sdata_add, sdata_add_h, on='cusip', how='outer')
    # holiding values
    sdata_add['prt'] = sdata_add['prccd']*sdata_add['h']
    return sdata_add

def eva(rt_s):
    # return evaluate matrics
    # Input variable: list of return
    # Type: list of float
    riskfree=0.02
    # Return, Standard Deviation, and Sharp Ratio
    rts=np.mean(rt_s)*250 # annulized return of the strategy
    stds=np.std(rt_s)*250**0.5 # annulized standard deviation of the strategy
    spr=(rts-riskfree)/stds

    # Value at Risk, and CVaR
    var5=np.percentile(rt_s, 5)
    var10=np.percentile(rt_s, 10)
    cvar5=np.mean([num for num in rt_s if num<var5])
    cvar10=np.mean([num for num in rt_s if num<var10])
    return [rts, stds, spr, var5, var10, cvar5, cvar10]

def plotReturn(rt_s, yr2, bins, cum):
    # Return Distribution
    # Input variable: list of return, year, #bins, is cumulative or not
    # Type: list of float, int, int, str
    plt.xlabel('Returns')
    plt.ylabel('Observations')
    plt.title(cum+' Return Distribution of Year '+str(yr2))
    pnln, pnlbins, pnlpatches = plt.hist(rt_s, bins)
    # add a 'best fit' normal distributed line
    y = mlab.normpdf( pnlbins, np.mean(rt_s), np.std(rt_s))*(pnlbins[1]-pnlbins[0])*len(rt_s)
    l = plt.plot(pnlbins, y, 'r--', linewidth=1)
    plt.show()

    # PnL Time Series
    plt.xlabel( 'Time Horizon')
    plt.ylabel( 'Return')
    plt.title( cum+' PnL Time Serise Graph of Year '+str(yr2))
    plt.plot( rt_s) # [str(int(dti%1e4)) for dti in dt_s],
    # plt.plot([dti%1e4-630 for dti in dt_s], rt_s) # 0 on x-axis is the day of actual rebalance
    plt.show()

#"""
# Set path of the csv file
path = 'F:\Michael\Fordham_University\Alternative_Investment\Idx_Rb\Data\\'
sfile = 'data3.csv' # stock data from WRDS

# Col Names: gvkey iid  datadate    cusip   cshoc   prccd   conml   loc
# Used Col: datedate(int) cusip(int) cshoc(float64) prccd(float64)
sdata = pd.read_csv( path + sfile, usecols=['datadate', 'cusip', 'cshoc', 'prccd'])
print('csv data file read' + '\n')

# Add year
```

```python
sdata['year']=sdata['datadate']/1e4
sdata['year']=sdata['year'].astype(int)
print('year added' + '\n')

# Add and Calculate Market Capitalization (float64)
sdata['mktcap']=sdata['cshoc']*sdata['prccd']
print('Mkt Cap calculated' + '\n')

# Add Rank (float64) by Market Cap ('mktcap') group by 'date'
sdata['rnk']=sdata.groupby('datadate')['mktcap'].rank(ascending=False)
print('Rank added' + '\n')

#-------------Done w/ Reading Files and Data Input-------------------#
#"""
#------------------------------Input Variables-----------------------#
yrb=2006 # the beginning year
yre=2016 # the ending year
mth = 5 # month of entering
dt = 20 # date of entering
mth_r = 5 # month of reconsititution calculation
dt_r = 31 # date of reconsititution calculation
rnk_lb = 1000 # lower bond of the rank
rnk_ub = 3000 # upper bond of the rank
# bd = 80 # entering/begining date
mth_e = 7 # exiting month
dt_e = 10 # exiting date
print('Input variables added' + '\n')

#-------------Initialize Lists of Evaluation Matrix-----------------#
ls_to=[] # list of turn over rate
ls_add_rt=[] # list of added stock returns
ls_del_rt=[] # list of deleteded stock returns
ls_add_n=[] # list of number of added stocks
ls_del_n=[] # list of number of deleted stocks

ls_prt_rt=[] # list of lists of portfolio's returns yearly
rt_s_all=[] # list of portfolio's returns in total
rt_c_all=[] # list of portfolio's cumulative returns in total
```

```python
Avg_Return=[] # list of portfolio's returns
Std_Dev=[] # list of portfolio's volatility(Standard Deviation)
Sharp_Ratio=[] # list of portfolio's Sharp ratio
VaR_5prct=[] # list of portfolio's 5% VaR
VaR_10prct=[] # list of portfolio's 10% VaR
CVaR_5prct=[] # list of portfolio's 5% CVaR
CVaR_10prct=[] # list of portfolio's 10% CVaR

#----------------Implement and Evaluate Strategy-------------#
for yri in range(yrb, yre): # yrb+2
    # Rank stocks and choose them by market cap
    yr1 = yri
    mth1 = mth_r
    dt1 = dt_r
    [rd, sd1] = aDate(sdata, yr1, mth1, dt1) # rd is the eintering date of last year

    yr2 = yri+1
    mth2 = mth
    dt2 = dt
    [bd, sd2] = aDate(sdata, yr2, mth2, dt2) # bd is the begining/entering date

    # Find the stocks to add or delete
    sd1rsl = sd1[sd1['rnk']<rnk_ub][sd1['rnk']>rnk_lb] # range of stocks in year1
    sd2rsl = sd2[sd2['rnk']<rnk_ub][sd2['rnk']>rnk_lb] # range of stocks in year2
    adds = sd2rsl[ ~sd2rsl['cusip'].isin(sd1rsl['cusip'])]
    dels = sd1rsl[ ~sd1rsl['cusip'].isin(sd2rsl['cusip'])]

    ls_add_n.append(adds.shape[0])
    ls_del_n.append(dels.shape[0])

    # Calculate entering and existing date
    # date2 is the existing date
    ed = int(yr2*1e4 + mth_e*1e2 + dt_e)

    # Form the portfolio
    sdata_add = formPort( sdata, adds, 1, 'mktcp')
    # sdata_add = formPort( sdata, adds, 1, 'equal')
```

```
sdata_del = formPort( sdata, dels, -1, 'mktcp')
# sdata_del = formPort( sdata, dels, 1, 'equal')
sdata_prt=sdata_add.append( sdata_del)
# Calculate daily price of the portfolio
p_d=sdata_prt.groupby('datadate')['prt'].sum() #.to_frame() #.tolist()

# Calculate daily return of the portfolio
#Failed Attempt
#p_d['datadate']=p_d.index
#p_d['dt_rnk']=p_d['datadate'].rank(ascending=True)
#p_d['dt_rnk-1']=p_d['dt_rnk']-1
#p_d.join( p_d, on=['dt_rnk', 'dt_rnk'], how='inner')
rt_s=[] # list of daily return
rt_c=[]  # list of cumulative return
dt_s=[] # list of daily date

for prti in range(1, len(p_d)):
    rti=p_d.iloc[prti]/p_d.iloc[prti-1]-1 # Cumulative simple return
    rti_c=p_d.iloc[prti]/p_d.iloc[0]-1 # intraday simple return
    if(rti<0.5 and -0.5<rti and rti_c<1 and rti_c>-1):
        rt_s.append(rti) # list of daily return during investing period
        rt_c.append(rti_c)  # list of cumulative return
        dt_s.append(p_d.index[prti]) # list of daily return during investing period

# Store daily return of the portfolio
ls_prt_rt.append((list, rt_s))
rt_s_all.extend(rt_s) # append works in a wrong way
rt_c_all.extend(rt_c)

#----------------Evaluate the Strategy-------------#
[rts, stds, spr, var5, var10, cvar5, cvar10] = eva(rt_s)
Avg_Return.append(rts) # list of portfolio's returns
Std_Dev.append(stds) # list of portfolio's volatility(Standard Deviation)
Sharp_Ratio.append(spr) # list of portfolio's Sharp ratio
VaR_5prct.append(var5) # list of portfolio's 5% VaR
VaR_10prct.append(var10) # list of portfolio's 10% VaR
CVaR_5prct.append(cvar5) # list of portfolio's 5% CVaR
CVaR_10prct.append(cvar10) # list of portfolio's 10% CVaR
```

```
    #-Plot the return distribution and pnl time-series
    plotReturn(rt_s, yr2, 60, 'Interday')
    plotReturn(rt_c, yr2, 60, 'Cumulative')

# evaluate the strategy overall
[rts, stds, spr, var5, var10, cvar5, cvar10] = eva(rt_s_all)
Avg_Return.append(rts) # list of portfolio's returns
Std_Dev.append(stds) # list of portfolio's volatility(Standard Deviation)
Sharp_Ratio.append(spr) # list of portfolio's Sharp ratio
VaR_5prct.append(var5) # list of portfolio's 5% VaR
VaR_10prct.append(var10) # list of portfolio's 10% VaR
CVaR_5prct.append(cvar5) # list of portfolio's 5% CVaR
CVaR_10prct.append(cvar10) # list of portfolio's 10% CVaR
plotReturn(rt_s_all, str(yrb)+' - '+str(yre), 60, 'Interday') # plot the return distribution and pnl time-
series
plotReturn(rt_c_all, str(yrb)+' - '+str(yre), 60, 'Cumulative') # plot the return distribution and pnl
time-series

ls_yr=[str(i) for i in range(yrb+1, yre+1)]
ls_yr.append('Overall')
eva_mtx=pd.DataFrame.from_items([ \
    ('index',ls_yr),('Avg_Return',Avg_Return),
    ('Std_Dev',Std_Dev),('Sharp_Ratio',Sharp_Ratio),
    ('VaR 5%',VaR_5prct),('VaR 10%',VaR_10prct),
    ('CVaR 5%',CVaR_5prct),('CVaR 10%',CVaR_10prct)])
```