The Alpha Strategy:

Sector ETF Rotation with Good Alpha

Subject : FNCE30010 Algorithmic Trading 2020

Submission : Project 2

Academics : Professor Peter Bossaerts & Dr. Nitin Yadav

Word Count : 2480

Name	**Student ID**	**Email**
Sidakpreet Singh Mann	921322	sidakpreetm@student.unimelb.edu.au
Pei Chi Hung	825751	phung1@student.unimelb.edu.au
Chan Jie Ho	961948	chanjieh@student.unimelb.edu.au

TABLE OF CONTENTS

1 Abstract

In the context of a Sector ETF Rotation strategy where we invest in highest performing sector ETFs based on historical returns and volatility, we investigate whether the augmentation of the momentum-based ‘timing the market’ approach with a strategic allocation factor can lead to the improvement of a base Sector ETF Rotation strategy implementation via the creation of a better-rounded portfolio. In particular, we opted to use the “Good” Alpha on a base strategy found on Quantopian, and backtested between 2003 and 2017, using the S&P 500 index as a benchmark. Our findings show that this strategy has the potential to improve the base strategy during bull markets such as 2003 to 2007 and seems to have no effect on the base momentum trading strategy in bear markets such as 2007 to 2009.

2 Introduction

The economy is made out of different sectors. Among some of the most common sectors are the Financial sector, Technology sector, Consumer Staples sector, and many others, all of which behave differently given the state of the economy – some sectors perform better while the economy is growing while others underperform in the same time period (Beers, 2020). Therein lies the rationale behind the Sector Rotation strategy where one rotates sectors depending on their outlook.

2.1 Sector ETF Rotation Strategy

The strategy in question is a momentum-based strategy that aims to time the market and rotate out of underperforming sectors while the stocks within the sector are still valuable before they drop. Also, purchasing stocks in other sectors at a discount before they increase in value. An example of this mindset could be seen when investors are anticipating an oncoming recession, they can rotate into Defensive sectors as they are less sensitive to the downturns of an economy such as Consumer Staples and Healthcare because these sectors encompass essential services.

2.2 Drawbacks

This strategy is not without its limitations (Barone 2019). One major issue can be the investors’ ability to correctly time their rotations with the market – a particularly difficult feat as the economy doesn’t follow a fixed cycle of equal recession and growth periods. This could result in suboptimal investment choices.

Another issue comes from transaction fees. By following the market cycle, there may be a frequent need to rebalance, thus compounding transaction costs, offsetting earned profits.

Further, (Barone 2019) shows that the majority of the profitability of momentum based strategies comes from shorting underperforming assets which has the potential to increase the portfolio's risk extensively.

2.3 Chosen Implementation

One such strategy that tries to tackle these issues is an implementation posted by Eric Bell on the Quantopian community in 2017 which took inspiration from The Lazy Trader (2015) and Cohn (2014). Bell's implementation has the following features:

1. Base universe comprising the 9 Sector ETFs from the US – sector ETFs are the set of stocks that are the closest representation to an actual sector in the economy and are therefore suitable for this strategy;
2. Keeps a long-only position – this avoids the additional risks that come with shorting stocks;
3. Rebalances his portfolio monthly – this helps reduce the number of transactions made, thus reducing the transaction costs.

Bell's rebalancing function involves choosing the best 3 ETFs using a ranking system that utilises a scoring formula that uses a combination of 70% of the 1-month mean daily return and 30% of the annualised volatility. This allows the strategy to take into account both the momentum and penalises stocks with high volatility as highlighted by The Lazy Trader (2016).

3 Improvement

While there were many limitations tackled by Bell's implementation, in our research, we learned that momentum strategies are best incorporated as part of a broader diversified portfolio (Carlson, 2017). Hence, we hypothesize that incorporating the Good Alpha as part of it's selection system would lead to the creation of better-rounded portfolios, which would result in a higher Sharpe ratio.

3.1 The "Good" Alpha

The Good Alpha, as per a research paper by Bossaerts and Yang (2015) and first implemented by Blume (1984); and Dybvig and Ross (1985), is a strategic allocation method that utilises the calculation of the alpha relative to one's current portfolio. The assumptions of the Good Alpha (as per our lectures) namely universe consisting of only ETFs, monthly rebalancing and rebalance weights of 5% are consistent with Bell's implementation (the base strategy that we improved). Furthermore, the alphas of ETFs are far more precise due to their lower volatility as typically ETFs are diversified relative to that sector.

Hence, we set out to regress returns of an ETF on our portfolio instead of regressing it on a benchmark index, in order to decide if it should be included in our portfolio. This is further explained in **Section 4.4**.

4 Implementation

Note: *This implementation was performed in Quantopian - a backtesting platform that has now phased out its community services so recreating these results using the same data is no longer possible. This algorithm can still be adapted to be used with zipline - the open source backend for the research section of the Quantopian backtesting website (works with Jupyter notebooks).*

Our efforts were focused more on the implementation of the Good Alpha strategy, and therefore we have focussed more on the methodology so the results can be easily recreated.

4.1 Challenges

The Quantopian platform provided a backend in which users could utilize large public datasets in order to backtest strategies. Within the IDE, users could specify the use of inbuilt "factors" which were functions that would perform predefined computations such as calculating the daily returns of each asset in the universe of chosen equities. Custom factors could also be defined in which the users would need to override the computation function ("compute") which would accept input vectors, and users could build their own output vector. This output vector would then be presented in the pipeline output as its own column, with each row representing an equity on a certain date.

However, the **limitation** of custom factors is that they are often only run on the data present within the datasets. From research, we found that it is quite difficult to adapt custom factors to build a custom factor that queries both the statistics of the security, as well as any data maintained outside the datasets. Further, due to the time-out limits imposed on the calculations, performing any computationally heavy task such as rolling regressions would often lead to time-outs. The platform would try to process 6 months of data at the same time leading to timeouts. After a short search, we found that most online backtesting platforms impose this limit and therefore accounting for it is quite important.

Due to the above reasons, the solution we came up with was to **not use factors**, and instead perform computations at the start of each trading day. This means we could work around the time-outs by spreading the computations across the backtest rather than computing it all simultaneously.

4.2 Universe of stocks

Setting the SPDR S&P 500 ETF (SPY) as our benchmark, we define our Universe with the SPDR Select Sector Funds below:

- Consumer Staples Select Sector SPDR ETF (XLP)
- Consumer Discretionary Select Sector SPDR ETF (XLY)
- Energy Select Sector SPDR ETF (XLE)
- Financials Select Sector SPDR ETF (XLF)
- Health Care Select Sector SPDR ETF (XLV)
- Industrials Select Sector SPDR ETF (XLI)
- Materials Select Sector SPDR ETF (XLB)
- Technology Select Sector SPDR ETF (XLK)
- Utilities Select Sector SPDR ETF (XLU)

Overall, the Sector ETFs are highly liquid and tend to be cost efficient due to the competition in this space. Moreover, as these ETFs are concentrated in firms with large caps, they offer stability which makes them appealing to investors who seek to implement the sector rotation strategy.

4.3 Initial Portfolio

The initial portfolio consisted of US$100K in cash. There was no initial allocation of stocks.

4.4 Quantopian Implementation

The code for this strategy can also be found in the appendix section 9.1.

1. At the start of each day, starting portfolio value is recorded within the context object. At the end of the same day, the ending value is recorded in order to calculate the daily return on our portfolio.

2. A rolling window of 30-day daily returns for our portfolio is maintained within the context object at all times. Since our portfolio is not an asset that exists within a database, this needs to be manually maintained. Each day, a new daily return float is added to the container, and the first daily return is removed. This forms our vector of portfolio returns on which we will be regressing assets.

3. We run the base ETF rotation strategy, and determine what the top 2 ETFs are which Bell's strategy specifies we should take a long position in.

4. A dictionary is maintained to record the "Good Alpha" for each asset. Next, we regress the entire universe of ETFs 30-day historical returns on our portfolio to determine the "Good Alpha" (based on the definition in our lecture notes). An alpha is only recorded if **it is significant at the 10% level.**

5. We loop over the recorded alphas. If an ETF has a **positive alpha**, we wish to set it's target allocation to 5% of our portfolio. If this ETF was also one of the top 2 ETFs chosen by the momentum strategy, then we increase this target allocation to 10%.

6. If an ETF has a **negative alpha**, and the base strategy picked to long this ETF, we choose not to buy it instead. If we have a long position in this ETF already, then we decrease the weight in it.

7. Based on the decisions described in points 5 and 6 we rebalance the portfolio at the start of each month. This order is placed using the inbuilt quantopian "order" and "order target percent" functions.

   This process is implemented across multiple backtest periods, and performance metrics are generated using the inbuilt zipline "tearsheets".

5 Results

We backtested both of the base sector rotation strategy and the combination of Good Alpha and momentum strategy with the starting point of US$100K in cash and S&P 500 (SPY) as the benchmark.

Apart from backtesting on the full period *from* 25 December 2003 *to* 22 March 2017, we divided this time range to two bull periods (*from* 2003 *to* 2007 and *from* 2009 *to* 2017) and a bear period (*from* 2007 *to* 2009) to investigate how both strategies perform in different market conditions.

We observed that the strategy creates a marginal improvement in the sharpe ratio and returns across the testing periods while resulting in a slight increase in volatility.

5.1 Full period from 2003 to 2017


Cumulative returns
Equity(8554 [SPY])
Backtest
Cumulative returns
3.50
3.00
2.50
2.00
1.50
1.00
2004
2006
2008
2010
2012
2014
2016


Cumulative returns
Equity(8554 [SPY])
Backtest
Cumulative returns
3.50
3.00
2.50
2.00
1.50
1.00
2004
2006
2008
2010
2012
2014
2016

Figure 1. Backtest comparing the Cumulative Returns of the benchmark (Grey line) and the strategy (Green line) – Top: Base rotation strategy; Bottom: Improvised strategy

	Annual Return	Cumulative Return	Volatility	Sharpe	Drawdown
Base	9.68%	239.27%	16.93%	2.5	-32.9%
Improvised	9.76%	242.35%	17.37%	2.57	-32.91%

Table 1. *Backtest performance on base and improvised strategies*

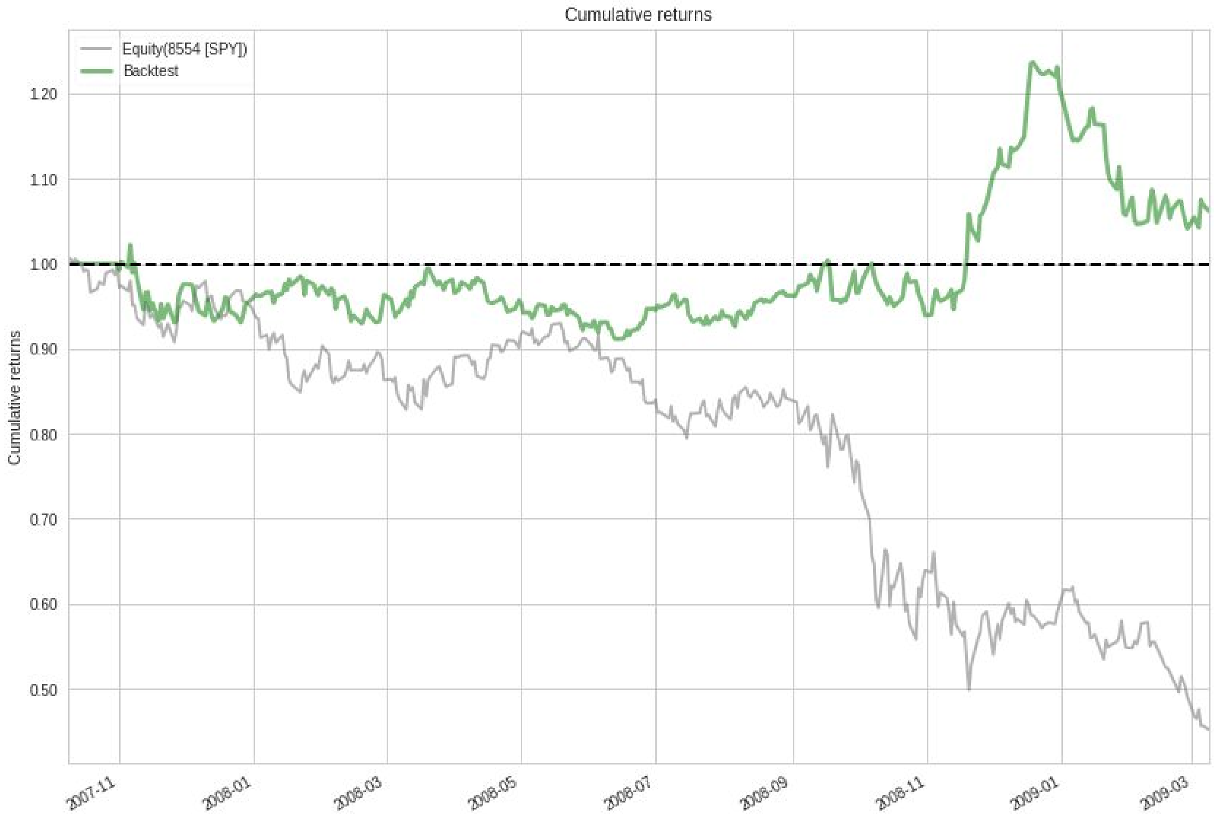
5.2 Bull period from 2003 to 2007

Figure 2. *Backtest comparing the Cumulative Returns of the benchmark (Grey line) and the strategy (Green line) – Top: Base rotation strategy; Bottom: Improvised strategy*

	Annual Return	**Cumulative Return**	**Volatility**	**Sharpe**	**Drawdown**
Base	15.46%	72.13%	15.49%	1.21	-13.52%
Improvised	15.95%	74.91%	15.78%	1.4	-13.524%

Table 2. *Backtest performance on base and improvised strategies*

5.3 Bear period from 2007 to 2009


Cumulative returns
Equity(8554 [SPY])
Backtest
1.20
1.10
1.00
0.90
0.80
0.70
0.60
0.50
Cumulative returns
2007-11
2008-01
2008-03
2008-05
2008-07
2008-09
2008-11
2009-01
2009-03

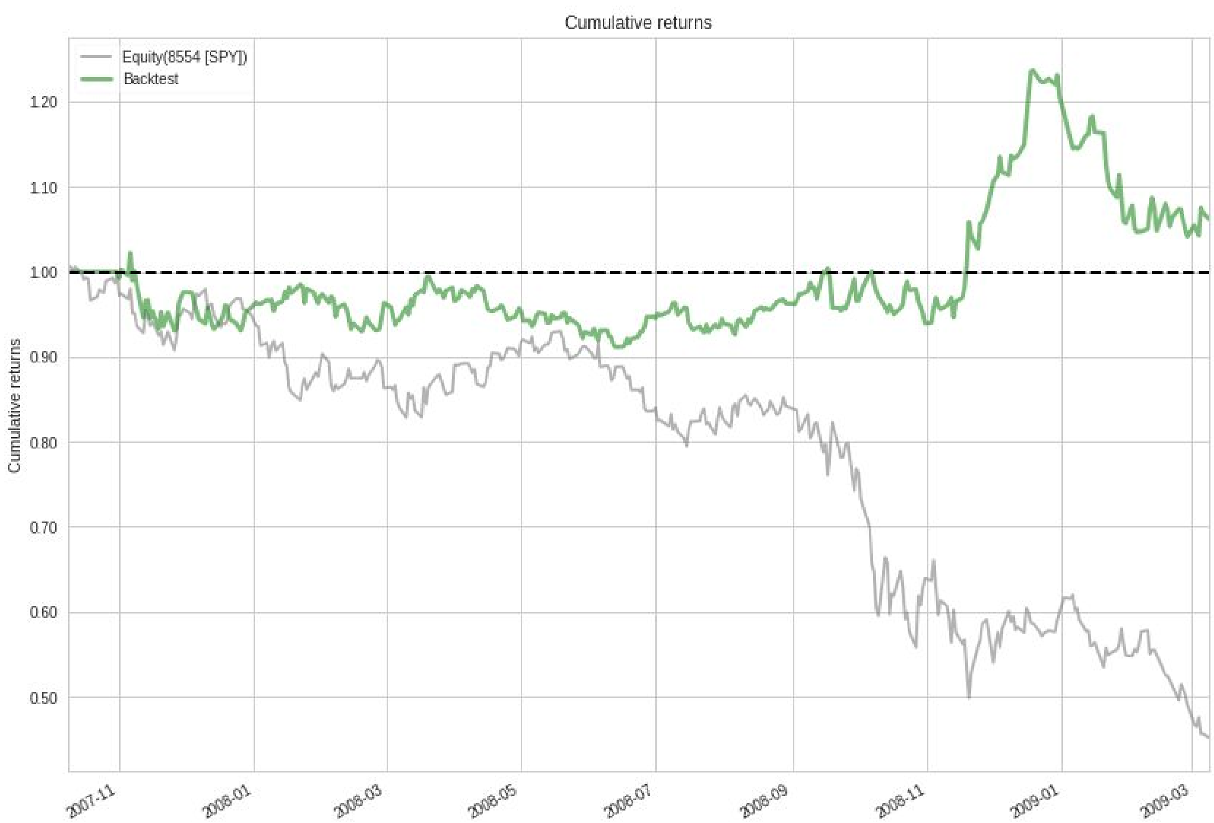

Cumulative returns
Equity(8554 [SPY])
Backtest
1.20
1.10
1.00
0.90
0.80
0.70
0.60
0.50
Cumulative returns
2007-11
2008-01
2008-03
2008-05
2008-07
2008-09
2008-11
2009-01
2009-03

__Figure 3.__ Backtest comparing the Cumulative Returns of the benchmark (Grey line) and the strategy (Green line) – Top: Base rotation strategy; Bottom: Improvised strategy

	Annual Return	**Cumulative Return**	**Volatility**	**Sharpe**	**Drawdown**
Base	4.33%	6.17%	18.66%	0.85	-15.79%
Improvised	4.33%	6.17%	18.66%	0.85	-15.79%

__Table 3.__ Backtest performance on base and improvised strategies

During this bear period, the backtest results of both the base sector rotation strategy and the combined strategy with alpha are identical. When investigating further into the transactions that occurred, the ETFs in our universe did not obtain significant and positive alphas throughout this time range.

5.4 Bull period from 2009 to 2017

Figure 4. *Backtest comparing the Cumulative Returns of the benchmark (Grey line) and the strategy (Green line) – Top: Base rotation strategy; Bottom: Improvised strategy*

	Annual Return	**Cumulative Return**	**Volatility**	**Sharpe**	**Drawdown**
Base	7.68%	81.28%	17.02%	2.38	-24.299%
Improvised	7.63%	80.55%	17.57%	2.45	-24.299%

Table 4. *Backtest performance on base and improvised strategies.*

5.5 Overview

Overall, the results here are only marginally better than the base strategy, and therefore we are unable to determine whether there is any clear advantage in augmenting the base strategy with our improvement.

6 Discussion

6.1 Untackled Limitations

While the use of Good Alpha in this particular setting does help earn a marginally higher sharpe, some limitations were still left untackled. For example, as evident in the second bull market backtest period, Sector ETF Rotation does not always beat the market. We suspect this is due to the fact that with using the 1-month mean daily return in the scoring function, one will constantly be chasing the recent performance only (The Lazy Trader, 2015), which could lead to suboptimal portfolios in unstable markets. One could potentially rotate into a sector that has performed well recently (within the last month), which immediately begins to decrease in value.

For example, one possible way to improve on this is to include the 3-month return as part of the scoring as this ensures one should require sustained evidence of an upwards trend before they invest into it.

6.1 Future Modifications

We would like to explore different combinations of this scoring function using machine learning or perhaps incorporation other risk factors into the creation of an even broader diversified portfolio.

Good Alpha strategy can be tested as the sole trading signal. This can be compared to the momentum trading (tactical allocation of assets) and S&P index (buy and hold benchmark). This would have helped us to estimate the advantage gained by the Good Alpha as the sole trading signal. However due to Quantopian community services being discontinued, we were unable to do so.

7 Conclusion

Our findings suggest that the combination of the sector rotation and Good Alpha was successful in the bull markets, and had no effect during the bear markets. Supplementing tactical asset allocation approaches with a factor that strategically allocates assets has proven successful in determining the right sectors that we should rotate into, however the results remain statistically insignificant. There are many areas that require further research such as the scoring function and deciding on the weights based on trading signals to ensure the success of this trading strategy in practise.

8 References

Barone, A. (2019). *What Is Momentum Trading?* Investopedia. https://www.investopedia.com/trading/introduction-to-momentum-trading

Beers, B. (2020). *Sector Rotation.* Investopedia. https://www.investopedia.com/articles/trading/05/020305.asp

Berkin, A. L., & Swedroe, L. E. (2016). *Your Complete Guide to Factor-Based Investing: The Way Smart Money Invests Today.* Buckingham.

Blume, M. E. (1984). The use of "alphas" to improve performance. The Journal of Portfolio Management, 11(1), 86-92.

Bossaerts, P., & Yang, W. (2015). *Using alpha to generate alpha*. Working Paper.

Carlson, B. (2017). *Momentum Investing Has a Place in Portfolios.* Bloomberg. https://www.bloomberg.com/opinion/articles/2017-10-24/momentum-investing-has-a-place-in-portfolios

Cohn, M. (2014). *Developing A Rotation Strategy Using Highly Diversified ETFs - Part III.* Seeking Alpha. https://seekingalpha.com/article/2107713-developing-a-rotation-strategy-using-highly-diversified-etfs-part-iii

Dybvig, P. H., & Ross, S. A. (1985). The analytics of performance measurement using a security market line. The Journal of finance, 40(2), 401-416.

Unknown (2016). *ETF Rotation systems - ranking evaluation with Excel.* The Lazy Trader. http://www.the-lazy-trader.com/2016/08/evaluate-momentum-etf-rotation-systems-with-excel-tool.html

Unknown (2015). *ETF Rotation Systems to beat the Market - American Equities.* The Lazy Trader. http://www.the-lazy-trader.com/2015/01/etf-rotation-systems-to-beat-market-American-Equities.html

9 Appendix

9.1 Code

Please note: *This code is consistent with the quantopian framework for backtesting, and may need to be adapted to function on another backtesting platform).*

```
"""

FNCE30010  - Algorithmic Trading
Semester 2 - 2020

Sidakpreet Mann
Pei Chi Hung
Chan Jie Ho

Base Strategy - Sector rotation ETF strategy
                Trades based on 30 day returns and volatility
                Timing the market

                 Overall, can see that this strategy rotates around all sectors
=> depends on estimations, includes sectors that are doing well and decrease
sectors not doing well
              Favours larger caps
              Average-high MSCI ESG Fund Rating for most (except XLE), (A
rating is around 6.5. Note: "Highly rated funds consist of companies that tend
to show strong and/or improving management of financially relevant
environmental, social and governance issues. These companies may be more
resilient to disruptions arising from ESG events.")

                Implemented by Eric Bell on Quantopian.
                https://www.quantopian.com/posts/etf-rotation-strategy


Improvement   - Good Alpha strategy
                Regressing 30 day returns of ETFs on portfolio returns
                Strategic Allocation

"""
from quantopian.algorithm import attach_pipeline, pipeline_output
from quantopian.pipeline import Pipeline
import quantopian.pipeline.filters as Filters
import quantopian.pipeline.factors as Factors
from quantopian.pipeline.factors import DailyReturns
import pandas as pd
```

```
import numpy as np
from quantopian.pipeline.data.builtin import USEquityPricing
from quantopian.pipeline.factors import CustomFactor
import statsmodels.api as sm


def initialize(context):
# Consumer Staples (essential g&s) Select Sector SPDR ETF (TICKER: XLP)
#  Consumer  Discretionary  (non-essential  g&s)  Select  Sector  SPDR  ETF  (TICKER:
XLY)
# Energy Select Sector SPDR ETF (TICKER: XLE)
# Financials Select Sector SPDR ETF (TICKER: XLF)
# Health Care Select Sector SPDR ETF (TICKER: XLV)
# Industrials Select Sector SPDR ETF (TICKER: XLI)
# Materials Select Sector SPDR ETF (TICKER: XLB)
# Technology Select Sector SPDR ETF (TICKER: XLK)
# Utilities Select Sector SPDR ETF (TICKER: XLU)


    context.portfolio_30_day_returns = []

    context.spy = sid(8554)
    context.shy = sid(23921)
    context.stock= [sid(19659),
                   sid(19662),
                   sid(19655),
                   sid(19656),
                   sid(19661),
                   sid(19657),
                   sid(19654),
                   sid(19658),
                   sid(19660)]

    # ===========================
    # GOOD ALPHA IMPROVEMENT

    # Rebalance first day of every month, 1 hour after the market open.
        schedule_function(record_day_start_port_value,  date_rules.every_day(),
time_rules.market_open())
    # ===========================


                    schedule_function(rebalance,     date_rules.month_start(),
time_rules.market_open(hours=1))
                       schedule_function(close,      date_rules.month_start(),
```

```
time_rules.market_open())

    # Record tracking variables at the end of each day.
                  schedule_function(count_positions,    date_rules.every_day(),
time_rules.market_close())
                                                                               #
schedule_function(trail_stop,date_rules.every_day(),time_rules.market_open())


    #====================================
    # GOOD ALPHA IMPROVEMENT

     # Adjust array of daily returns of our portfolio to maintain a 30-day
window
          schedule_function(record_day_end_port_value,  date_rules.every_day(),
time_rules.market_close())
    #====================================

    # Create our dynamic stock selector.
    attach_pipeline(make_pipeline(context), 'myPipe')


# GOOD ALPHA IMPROVEMENT
def record_day_start_port_value(context, data):
    # record the portfolio value at the start of the day
    context.day_start_port_val = context.portfolio.portfolio_value

# GOOD ALPHA IMPROVEMENT
def record_day_end_port_value(context, data):
    # record portfolio value at the end of the day
    context.day_end_port_val = context.portfolio.portfolio_value

     daily_return  =  (context.day_end_port_val  -  context.day_start_port_val)  /
context.day_start_port_val

    if len(context.portfolio_30_day_returns) == 30:
        # adjust the array of portfolio daily returns
        context.portfolio_30_day_returns = context.portfolio_30_day_returns[1:]
+ [daily_return]
    elif len(context.portfolio_30_day_returns) < 30:
        context.portfolio_30_day_returns.append(daily_return)


def make_pipeline(context):
    """
```

```
     A function to create our dynamic stock selector (pipeline). Documentation
on
    pipeline can be found here: https://www.quantopian.com/help#pipeline-title
    """

    universe=Filters.StaticAssets(context.stock)

    # Factor of yesterday's close price.
    day20_ret=Factors.Returns(inputs=[USEquityPricing.close], window_length=21,
mask=universe)
                     day3mo_ret=Factors.Returns(inputs=[USEquityPricing.close],
window_length=63, mask=universe)

                     day6mo_ret=Factors.Returns(inputs=[USEquityPricing.close],
window_length=126,mask=universe)
                     day1yr_ret=Factors.Returns(inputs=[USEquityPricing.close],
window_length=252, mask=universe)

    volatility=Factors.AnnualizedVolatility(mask=universe)

     # THIS SCORING METHOD IS AN IMPROVEMENT THE SAME AUTHOR MADE IN A MORE
RECENT STRATEGY
      # THIS MODIFICATION WAS MODE TO MAKE IT CONSISTENT WITH THE OTHER
SUCCESSFUL SECTOR ROTATION STRATEGIES
    WEIGHT1=0.7
    WEIGHT2=0.3

    score=((WEIGHT1*day3mo_ret)+(WEIGHT2*volatility))
    score_rank=score.rank(ascending=False)


    best=(score_rank<=2)
    pipe = Pipeline(
        columns = {
            'Score': score,
            'Score_Rank' : score_rank
        },
        screen = (best)
    )
    return pipe

def before_trading_start(context, data):
    """
    Called every day before the market opened.
    """
```

```
    # ==============================================
    # GOOD ALPHA IMPROVEMENT
     # records only the regressions where the good-alpha is significant (each
day)
    context.regs = {}

    # Retrieve the previous 30 previous trading prices before each trading day
     etf_list = symbols('SPY', 'XLY', 'XLE', 'XLF', 'XLV', 'XLI', 'XLB', 'XLK',
'XLU')

    prices = data.history(etf_list,
                          fields='price',
                          bar_count=31,
                          frequency='1d')

    daily_rets_30_days = (prices.pct_change()).to_dict()

    for key, val in daily_rets_30_days.items():

        # don't do anything for the first 30 days
        if len(context.portfolio_30_day_returns) < 30:
            break

        # remove the first day  as pct changes, first = nan
        temp = sorted(val.items())[1:]

         # this is the 30 day window of data points for daily returns for each
security
        security_30daily_returns = [x[-1] for x in temp]

          # regress the asset on the currently stored 30 data points of daily
returns of
        # our portfolio to get the GOOD ALPHA
        y = security_30daily_returns
        x = context.portfolio_30_day_returns
        x2 = sm.add_constant(x)

        est = sm.OLS(y, x2)
        est2 = est.fit()

        alpha, beta = est2.params
        palpha, pbeta = est2.pvalues

        # record only the significant alphas
```

```
        if (palpha < 0.1):
            context.regs[key] = {'alpha': alpha, 'palpha': palpha}

    # ==============================================

    context.output = pipeline_output('myPipe')
    context.longs = context.output.index.tolist()


# BASE STRATEGY
# My Friend Blue helped me restructure the weight so I did not double leverage
on Safety Months
def close(context, data):
    for stock in context.portfolio.positions:
        if not data.can_trade(stock):
            continue
        order_target(stock, 0)
        log.info("Sell" + str(stock))


def rebalance(context, data):

    # BASE STRATEGY COMPONENT
    spy_200 = data.history(context.spy, "price", 200, "1d")
    spy_mavg = spy_200.mean()
    spy_price = data.current(context.spy, "price")
    if data.can_trade(context.shy) and spy_price < spy_mavg:
        # Do a check to make sure no positions before this 1.0 all-in
        order_target_percent(context.shy, 1.0)
        log.info("Safety Month")
        return

    # ========================================
    # GOOD ALPHA IMPROVEMENT

    for stock in context.longs:

        if not data.can_trade(stock):
            continue
        if get_open_orders(stock):
            continue

        order_target_percent(stock, 1.0 / len(context.longs))
                      #   order_target_value(stock,   context.portfolio.cash   /
len(context.longs))
```

```
        log.info("Buy" + str(stock))

    # if alpha is negative, reduce weight in this security
    for sec in context.portfolio.positions.keys():
        if sec in context.regs:
            if context.regs[sec]['alpha'] < 0:

                log.info("Negative alpha, reducing weight by 15%")
                position_size = context.portfolio.positions[sec].amount
                order(sec, -0.15 * position_size)

    # if cash is -ve, reduce exposure to entire portfolio
    if context.portfolio.cash < 0:
        for sec in context.portfolio.positions:
            order_target_percent(sec, 0.8 / len(context.portfolio.positions.keys()))
        return

    # if good alpha is +ve and a certain ETF has not been bought by base strategy
    # set its target weight to 5% of current total portfolio value
    for sec in context.regs:

        if context.regs[sec]['alpha'] > 0 and sec not in context.portfolio.positions:
            order_target_percent(sec, 0.05)

        elif context.regs[sec]['alpha'] > 0 and sec in context.portfolio.positions:
            current_pos_size = context.portfolio.positions[sec].amount
            current_alloc = current_pos_size / context.portfolio.value
            order_target_percent(sec, current_alloc + 0.05)
    # ========================================


# BASE STRATEGY COMPONENT
def count_positions(context, data):
    longs = 0
    for position in context.portfolio.positions.values():
        if position.amount > 0:
            longs += 1

    record(longs=longs)
```