Haohan Wang

Follow 272 Followers About

Factor Models Advanced Application

Haohan Wang Jan 6, 2016 · 7 min read

The factor models I used before are mostly fundamental factors which may not be very handy when we would like to add more elements into our model. Here I will explain an algorithm which uses Multi-Factor model to resolve this problem simply. Now, we can actually define our own customized factors and use them as filters or ranking system to allocate stocks into either long or short basket as well as to narrow down the stocks included in your universe.

The 'CustomFactors' that I used here are calculated using pricing, volume, and fundamentals data. By utilizing this method, we allow ourselves to have more freedom to choose whatever factors we want to add into our model so that we can define the inputs and the looking back window length based upon our intuition or research when we are creating the factors and then set them as default for the next step.

The 3 customized factors I defined here are:

- Momentum, which computed over last 200 days sans the most recent 50 days(they are just absolute closing prices instead of moving averages);
- Volatility or *annualized volatility* is the standard deviation of an instrument's yearly logarithmic returns. You can check the formula of annualized volatility here:

across different industries, I calculated the *z-score* of last two yield factors(normalize our factors). At last, we rank the results. (All the data here is from Morningstar fundamentals database and we only use the factor data from the previous day to calculate the score.)

I have explained momentum and fundamentals before. *Volatility* is a relatively new factor here. Based on the concept of volatility, it refers to the amount of uncertainty or risk about the size of changes in a security's value. A higher volatility means that a security's value can potentially be spread out over a larger range of values. This means that the price of the security can change dramatically over a short time period in either direction. Although, the higher the volatility the higher the option price to some point, we are trying to protect our portfolio from high volatility here(high volatility implies high risk). Therefore, we actually keep the volatility in the denominator part, that is to say, the higher the volatility the lower the factor output. We did this is because we want to let all the score of factor follow uniform descending order and we will pick the high-ranking stocks to long. By doing so, we can achieve the lower the volatility(ideal result) the higher the Volatility factor score.
As for the first CustomFactor 'AvgDailyDollarVolumeTraded' which is generated by computing the 20 days average product of close price and trading volume, I will use this factor to set up a security pool from which I can choose the securities to long and short later.

```
class AvgDailyDollarVolumeTraded(CustomFactor):

    inputs = [USEquityPricing.close, USEquityPricing.volume]
    window_length = 20

    def compute(self, today, assets, out, close_price, volume):
        out[:] = np.mean(close_price * volume, axis=0)
```

```
    inputs = [USEquityPricing.close]
    window_length = 200

    def compute(self, today, assets, out, close):
        out[:] = close[-50] / close[0]
```

```
class Volatility(CustomFactor):

    inputs = [USEquityPricing.close]
    window_length = 252

    def compute(self, today, assets, out, close):
        close = pd.DataFrame(data=close, columns=assets)
        # Since we are going to rank largest is best we need to invert the
sdev.
        out[:] = 1 / (np.log(close).diff().std()*((252)**(0.5)))
```

```
class Value(CustomFactor):

    inputs = [morningstar.earnings_ratios.diluted_eps_growth,
              morningstar.valuation_ratios.earning_yield,
              morningstar.valuation_ratios.fcf_yield]

    window_length = 1

    def compute(self, today, assets, out, eps_growth, earning_yield, fcf):
        value_table = pd.DataFrame(index=assets)
        value_table["eps_g"] = eps_growth[-1]
        value_table["ey"] = stats.zscore(earning_yield[-1],axis =0, ddof
=1)
        value_table["fcf"] =stats.zscore(fcf[-1], axis =0, ddof =1)
        out[:] = value_table.rank().mean(axis=1)
```

After setting up all of those factors, we will move into the next step. In this step, I add each factor into our pipeline. Following the normal procedures, we will also set up the commission and leverage here.

the ranking and apply this filter for each of the factor(3 in total) I defined.

I also set up the *stop loss* percentage here which is 98%(or you can interpret as if the loss exceeds 2% we will exit the position; A stop-loss order is designed to limit an investor's loss on a position in a security and we use it as a normal way to limit the maximum drawdown).

Another filter here set up by 'set_screen' through which we will only keep the stocks whose daily dollar trading volume is higher than 10^7 and their 200 day's simple moving average exceeds 5(as the remark above the sma_200, by filtering the stock with the criteria we can screen out penny stocks and low liquidity securities).

```
def initialize(context):
    set_commission(commission.PerShare(cost=0, min_trade_cost=None))
    context.stop_loss_pct = 0.98
    pipe = Pipeline()
    attach_pipeline(pipe, 'ranked_dv')

    pipe.add(Value(), "value")

    pipe.add(Momentum(), "momentum")
    pipe.add(Volatility(), "volatility")

    context.long_leverage = 1
    context.short_leverage = -1

    dollar_volume = AvgDailyDollarVolumeTraded()
    # marketcap = MarketCap()
    top_dv = dollar_volume.top(5000)

    # Screen out penny stocks and low liquidity securities.
    sma_200 = SimpleMovingAverage(inputs=[USEquityPricing.close],
window_length=200)

    pipe.set_screen((dollar_volume > 10**7)&(sma_200 > 5))
```

We rank the 3 customized factors following default descending order and added them into my pipeline afterward. In order to combine 3 factors, we create a 'combo_raw'

Based on the principle of long/short equity strategy, we will only long top 100 stocks and short bottom 100 stocks in terms of their combo_raw scores.

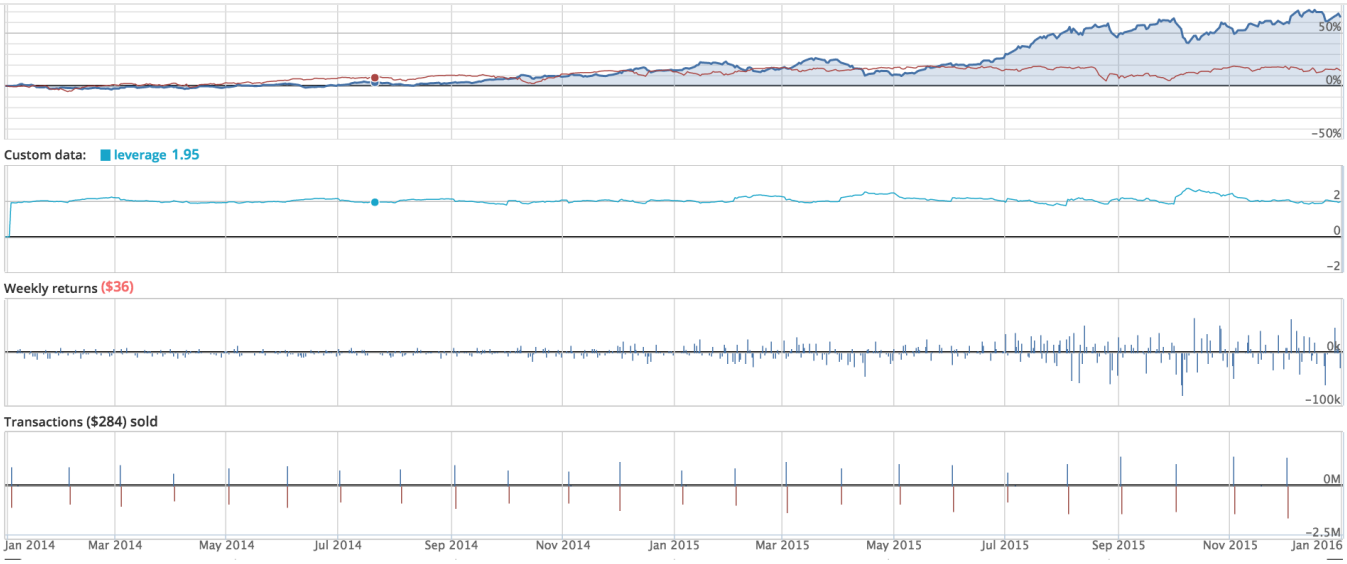
In the end, we will add the union of these two lists(long and short lists) of their indexes together to update my universe.

```
    # combo_raw = (value_rank + momentum_rank + volatility_rank)/3
    combo_raw = value_rank*0.4  + momentum_rank*0.4 + volatility_rank*0.2
    # combo_raw = (value_rank+ volatility_rank)/2

    pipe.add(combo_raw, 'combo_raw')
    pipe.add(combo_raw.rank(mask = top_dv), 'combo_rank')

    schedule_function(func=rebalance,
                      date_rule=date_rules.month_start(days_offset=0),

time_rule=time_rules.market_open(hours=0,minutes=30),
                      half_days=True)
```

```
securities to pass my screen
    # and the colums are the factors which I added to the pipeline
    context.output = pipeline_output('ranked_dv')
    #there are some NaNs in factor 2, I'm removing those
    ranked_dv = context.output.fillna(0)
    ranked_dv = context.output[(context.output.momentum_rank > 1)]
    ranked_dv = context.output[(context.output.value_rank > 0)]

    # Narrow down the securities to only the top 500 & update my universe
    context.long_list = ranked_dv.sort(['combo_rank'],
ascending=False).iloc[:100]
    context.short_list = ranked_dv.sort(['combo_rank'],
ascending=False).iloc[-100:]


update_universe(context.long_list.index.union(context.short_list.index))
```

The screenshot below shows the backtesting outcome from 2013–12–31 to 2015–12–31. You can also check it out from the video that I uploaded.

The total returns within these 2 years are 65.1% compared to the benchmark returns 14.6% and we can tell that our strategy beats the market to some extent. As for the β which is -0.08 in the end, which is good and it also implies that our strategy is hardly influenced by the market. The Sharpe Ratio is 1.68 which means for every unit risk that you are taking you can get 1.68 unit returns back. The only measure that may not be very ideal is MaxDD which is 14.09% and kind of high compared to the total returns.



As I mentioned in my *last article*, we can obtain more information about our backtest by the following analysis.

The information here is very clearly. You can get a more comprehensive picture about your strategy when you go through these tables and graphs such as I am concerned about the *Maximum Drawdown* (*Maximum Drawdown (MDD) is an indicator of downside risk over a specified time period. It can be used both as a stand-alone measure or as an input into other metrics such as "Return over Maximum Drawdown" and Calmar Ratio*)of my strategy. We can see from the analysis below, the worst drawdown happened on 2015–10–01 which is 14.15% along with the rest 4 worst drawdowns.

From the rolling portfolio beta of equity graph, we can tell that our portfolio β is around 0 and the average number is 0 or around 0 too. In terms of the rolling Sharpe Ratio, the average Sharpe ratios are around 2 and we can see that the Sharpe Ratio is above 0 through this period.

The average daily turnover is around 0.1and the average daily trading volume is around 10000. The annual returns of 2015 are significantly higher than ones in 2014.

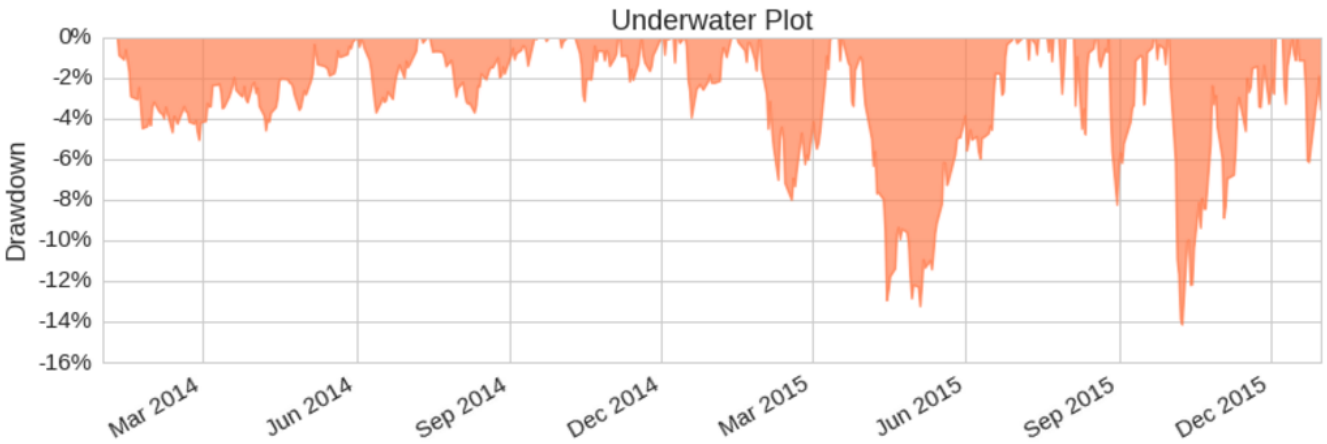
One interesting graph here is Rolling *Fama-French* (you can check my previous article to know more about *Fama-French Factor Model*)Single Factor-Betas which contains 3 factors: *SMB*, *HML*, and *UMD*(UMD (up-minus-down) is the momentum-factor-mimicking portfolio's return). We can see that our portfolio has higher momentum

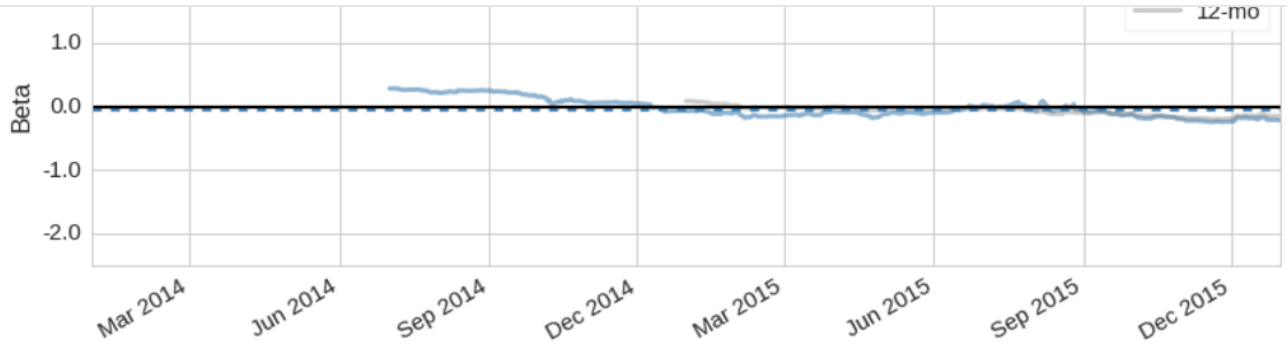
growing over time.(A common way to figure out an asset's factor exposures is to perform a multiple linear regression of an asset's periodic returns (usually monthly) against the returns of long-short factor-mimicking portfolios. Doing so creates a regression-based benchmark that tries to explain the returns of the asset. Unexplainable excess return, or alpha, is often interpreted as evidence of skill or some kind of additional risk not captured by the factor model.)

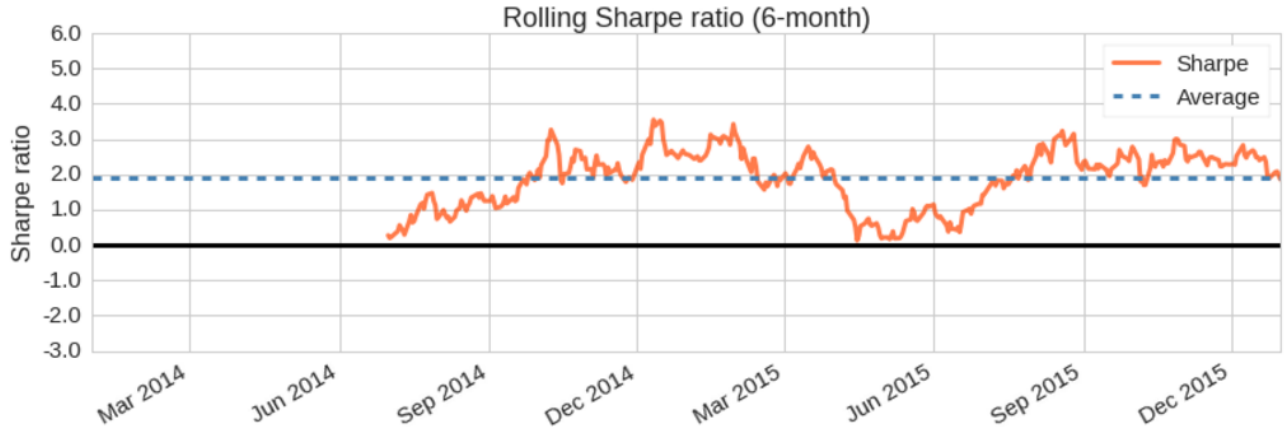
```
100% Time: 0:00:29|###########################################################|
Entire data start date: 2013-12-31
Entire data end date: 2015-12-31


Backtest Months: 24
                    Backtest
annual_return           0.28
annual_volatility       0.18
sharpe_ratio            1.48
calmar_ratio            2.01
stability               0.87
max_drawdown           -0.14
omega_ratio             1.30
sortino_ratio           2.20
skewness               -0.19
kurtosis                2.52
information_ratio       0.06
alpha                   0.27
beta                   -0.07

Worst Drawdown Periods
   net drawdown in %  peak date valley date recovery date duration
0              14.15 2015-10-01  2015-10-09    2015-12-04       47
1              13.25 2015-03-19  2015-05-05    2015-06-30       74
2               8.26 2015-08-26  2015-08-31    2015-09-23       21
4               8.02 2015-01-29  2015-02-17    2015-03-13       32
3               6.17 2015-12-17  2015-12-24           NaT      NaN
```


Cumulative returns
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
Mar 2014
Jun 2014
Sep 2014
Dec 2014
Mar 2015
Jun 2015
Sep 2015
Dec 2015

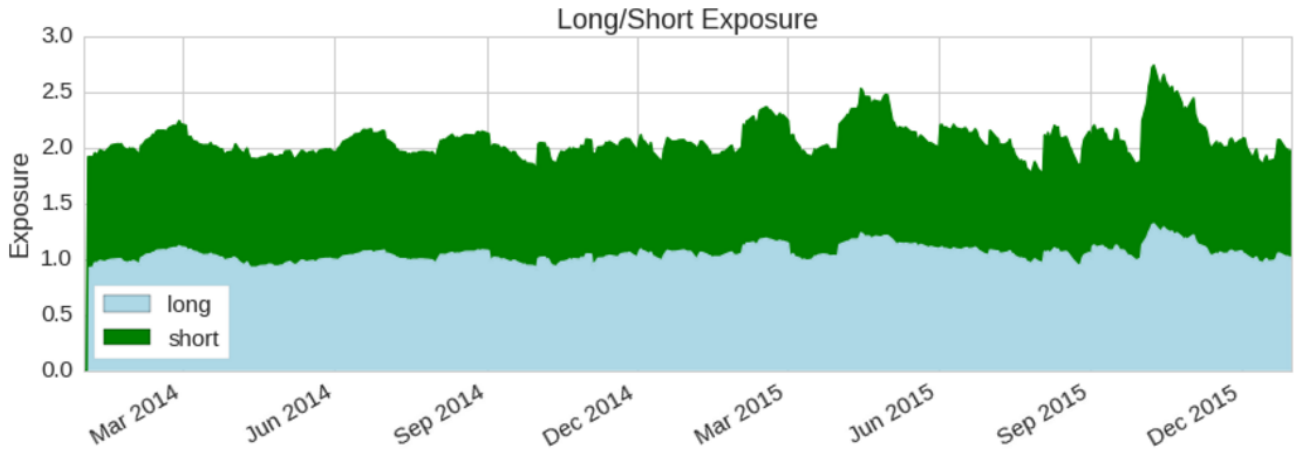

Cumulative return
1.5
1.4
1.3
1.2
1.1
1.0
0.9
Mar 2014
Jun 2014
Sep 2014
Dec 2014
Mar 2015
Jun 2015
Sep 2015
Dec 2015

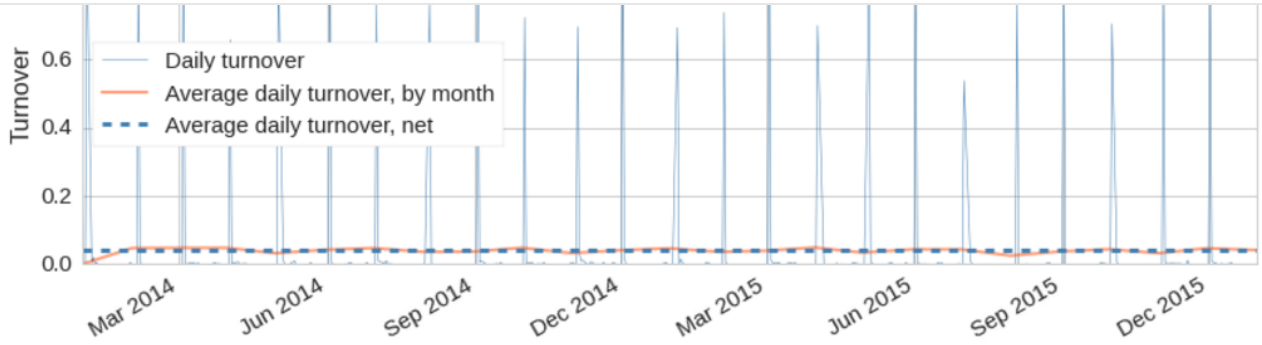

Underwater Plot
Drawdown
0%
-2%
-4%
-6%
-8%
-10%
-12%
-14%
-16%
Mar 2014
Jun 2014
Sep 2014
Dec 2014
Mar 2015
Jun 2015
Sep 2015
Dec 2015

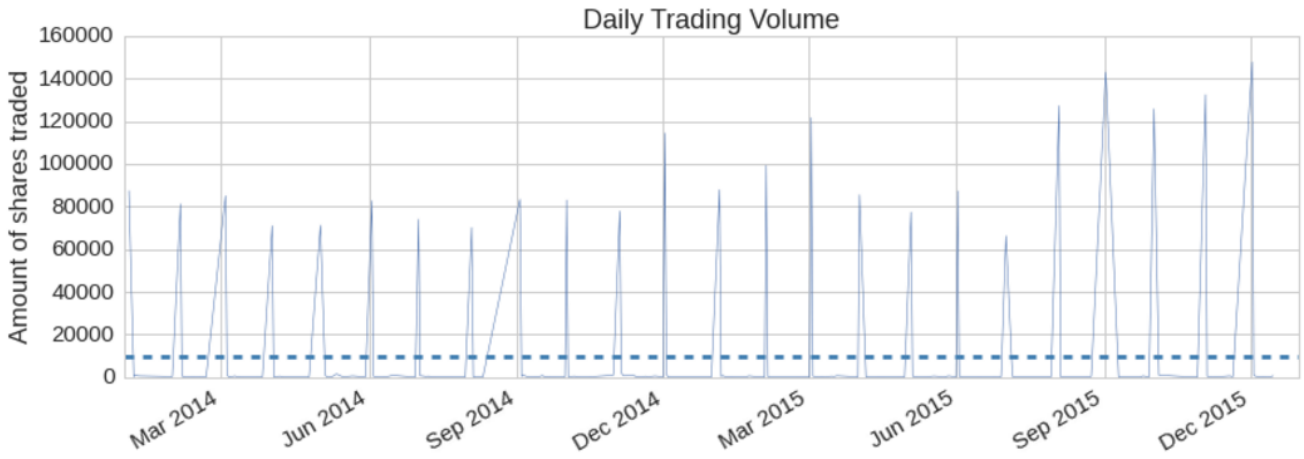

12-mo
Beta
1.0
0.0
-1.0
-2.0
Mar 2014
Jun 2014
Sep 2014
Dec 2014
Mar 2015
Jun 2015
Sep 2015
Dec 2015

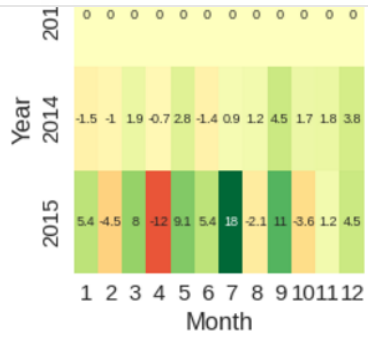

Rolling Sharpe ratio (6-month)
Sharpe
Average
Sharpe ratio
6.0
5.0
4.0
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
Mar 2014
Jun 2014
Sep 2014
Dec 2014
Mar 2015
Jun 2015
Sep 2015
Dec 2015

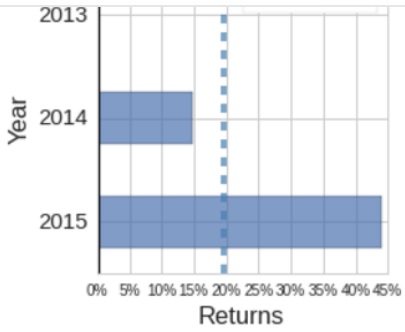

Gross Leverage
2.5
2.0
1.5
1.0
0.5
0.0
Mar 2014
Jun 2014
Sep 2014
Dec 2014
Mar 2015
Jun 2015
Sep 2015
Dec 2015

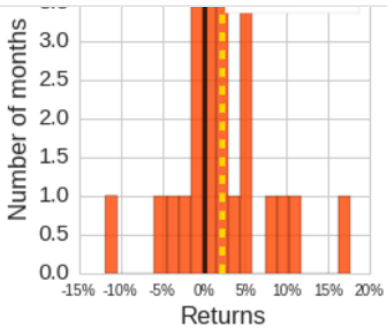

Long/Short Exposure
Exposure
3.0
2.5
2.0
1.5
1.0
0.5
0.0
long
short
Mar 2014
Jun 2014
Sep 2014
Dec 2014
Mar 2015
Jun 2015
Sep 2015
Dec 2015

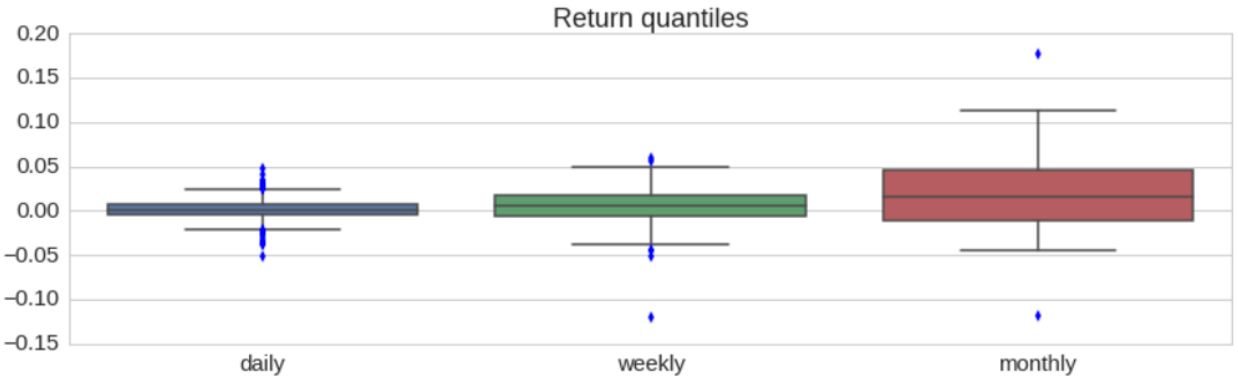

Turnover
0.6
0.4
0.2
0.0
Daily turnover
Average daily turnover, by month
Average daily turnover, net
Mar 2014
Jun 2014
Sep 2014
Dec 2014
Mar 2015
Jun 2015
Sep 2015
Dec 2015

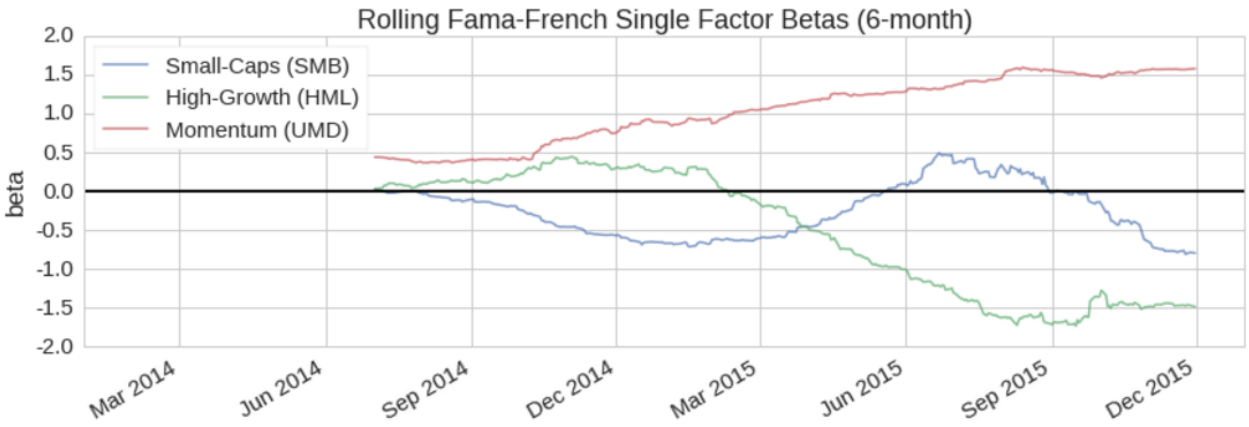

Daily Trading Volume
Amount of shares traded
160000
140000
120000
100000
80000
60000
40000
20000
0
Mar 2014
Jun 2014
Sep 2014
Dec 2014
Mar 2015
Jun 2015
Sep 2015
Dec 2015











Hope you enjoy the content. Please feel free to share your ideas or comments here with me here. Thank you!

Reference:

- *https://en.wikipedia.org/wiki/Volatility_(finance)*

- *http://www.investopedia.com/terms/m/maximum-drawdown-mdd.asp*
- *https://en.wikipedia.org/wiki/Fama%E2%80%93French_three-factor_model*
- *https://en.wikipedia.org/wiki/Carhart_four-factor_model*
- *http://ibd.morningstar.com/article/article.asp?id=636847&CN=brf295,http://ibd.morningstar.com/archive/archive.asp?inputs=days=14;frmtId=12,%20brf295*