**Part – I:**

**Project Proposal:**

I try to simulate an alpha using a formulaic expression within Websim. The Alpha is as follows:

*(-1 \* rank(covariance(rank(close), rank(volume), 5)))*

The expression means that we heavily utilize the function ‘rank’

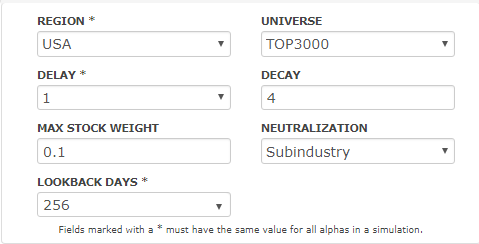
That is, rank the values of x among all instruments, and returns float numbers equally distributed between 0.0 and 1.0.  
For example, given 6 stocks with close price [20.2, 15.6, 10.0, 5.7, 50.2, 18.4], Rank(close) returns [0.8, 0.4, 0.2, 0.0, 1.0, 0.6]

We then use covariance, that is, Covariance of the values in vectors x and y for the past n days. Here, n is 5, and we then negate that.

This formulaic expression is then run over a given US equity population and a simulation is generated.

**Explanation:**

**Data used:**



The above screenshot depicts the Websim settings used for the simulation. From this, it is clear that the Region for which the simulation was carried out was US, and the universe of stocks was Top3k.

**Methodologies:**

Delay refers to the availability of data relative to decision time, which was set to 1, this means that our strategy is conservative, and that there is a lag of a day.

A decay of 4 is used. Decay performs a linear decay function over the past n days by combining today’s value with previous days’ decayed value.

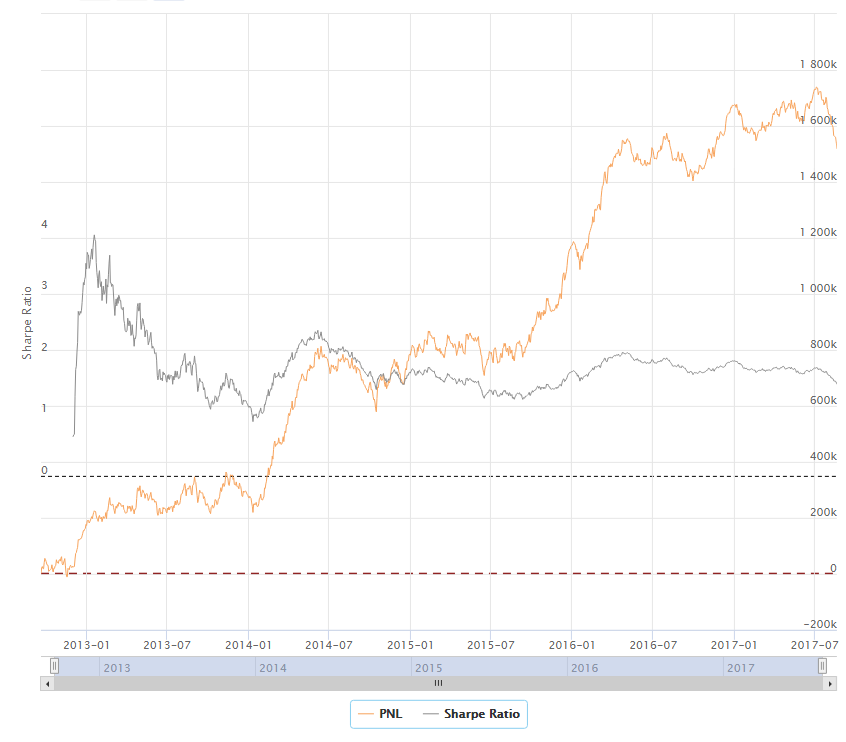
The maximum weight for each stock in the overall portfolio is set to 0.1, that is, 10%.

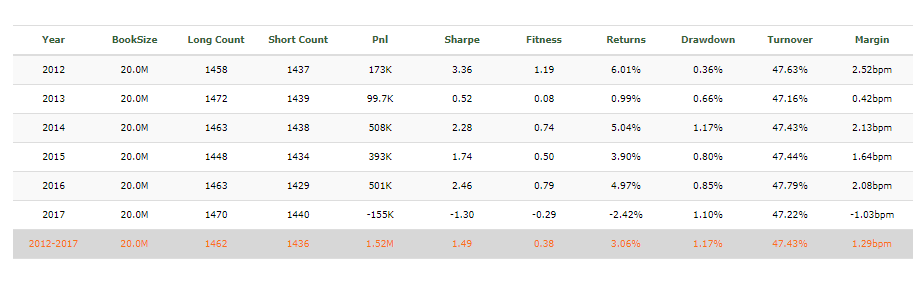
Neutralization = Industry or Subindustry, means that all the instruments in the Alpha vector are grouped into smaller buckets corresponding to industry or sub-industry and neutralization is applied separately to each of the buckets.

Loopback days is the number of prior days’ data to look at when running the Alpha for each day, which is set to 256.

**Observations / Performance Metrics:**

The simulation is run for 5 years, and the following is the backtest output:





As one can observe, the sharpe ratio is positive throughout for all the years, except for 2017. Overall, the average sharpe ratio for 5 years, between the year 2012 to 2017 is **1.49** which means that this alpha is decent due to its positive sharpe ratio.

**Potential Improvements:**

One improvement could be the fact that we could take co-variance of high and volume, instead of close, and play around with the ‘n’ value of co-variance to give a better yielding alpha.

**Part – II:**

**Project Proposal:**

I’ve tried to create a strategy based on moving averages (fast and slow moving averages), where slow moving average has a window of 50 ticks and fast moving average has a window of 10 ticks.

The data used is Bitcoin-USD data, that is, it is not an equity related data.

I’ve used the Zipline library from Quantopian in order to perform my backtest.

**Explanation:**

**Installation:**

1. Please make sure that the interpreter is **Python version 2.7.** The script will not behave as intended if the installation is even a version higher. Only applicable for PCs that have a Windows installation
2. To install all the relevant python packages, use the command on the command prompt as ‘pip install –r requirements.txt’ after extracting the zip folder
3. To run the file, on a command prompt type ‘python fp.py’
4. NOTE: The zipline library is heavy and the program takes quite some time to run.

**Data used:**

The data used is Bitcoin-USD minute-by-minute tick data. The minute-by-minute data indicates that the data is of high quality.

**Assumptions:**

Zero brokerage and slippage is considered. This does not indicate a real life scenario as there is considerable brokerage and slippage affecting the strategy.

The trading calendar does not have any holidays and is a 24x7 calendar.

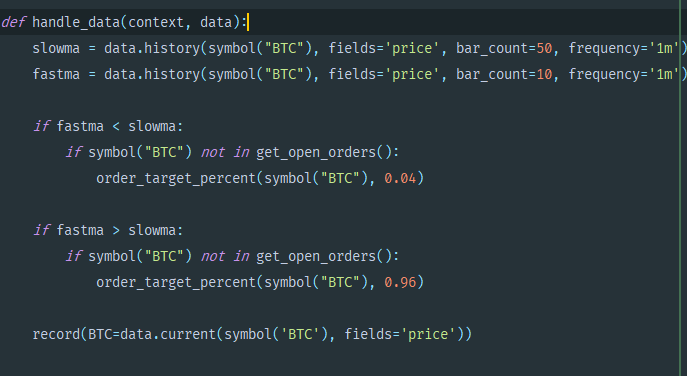
**Methodologies:**

Data is cleansed and UNIX time is converted to 1 minute datetime in pandas dataframes.

The method or strategy utilized for alpha generation is a fast and slow moving average, where the fast moving average is of 10 ticks or minutes and the slow moving average is of 50 ticks or 50 minutes.

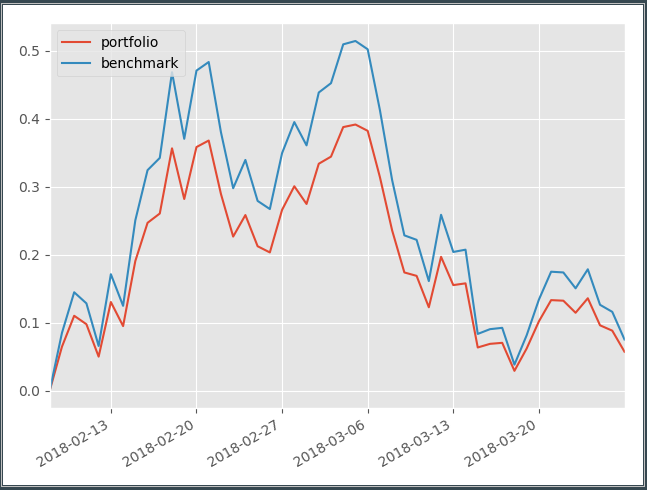
A buy signal is sent when the fast moving average is above the slow, and a sell signal is sent when the slow moving average is above the fast.

All this is handled with the helper function ‘handle\_data’ provided by zipline.



**Observations / Performance Metrics:**

The portfolio, on an average, seems to be underperforming the benchmark.



**Potential Improvements:**

The strategy could potentially be changed to consider different bands of Moving averages and generating a Buy or Sell signal based on logic of each band. For example, a fast, medium and slow moving average, and a strategy that generates buy-sell signals based on these and the previous buy-sell signal. This could potentially improve our returns.