

# Generating Alpha

Data Mining Spring 2019

Philip Kiely, Richard Greenbaum, Rudolf  
Hernandez, Alex Foster



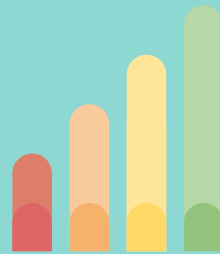
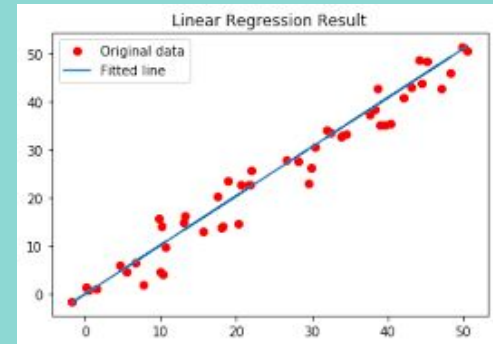
# Project Overview

- Goal was to make a successful stock trading algorithm
- We used Quantopian, an open source quant trading platform
- Our primary task was to create a heuristic that measures the strength of a stock at a given time
  - Allows us to decouple returns from market fluctuations



# Applications of Data Mining

- This project relied heavily on the soft skills emphasized this semester
  - Researching to find the best data
  - Data preprocessing
  - Evaluating model results
- We used python and ipython in Quantopian
- We used a scikit linear regression model for our final algorithm



- 
- 
- 
-



# Alpha Factors



- **Alpha Factor:** A value which could potentially be predictive of the future performance of a stock
- **Analyzing Alpha Factor:** Run through Quantopian's Alphas.
  - Indicates how predictive the alpha factor is.
- We produced **four** alpha factors





# Project Pipeline

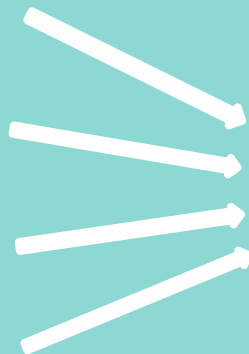
Raw Data

Alpha Factors

Linear Regression

Stock Rating

Ranking/Decision



Long

Short



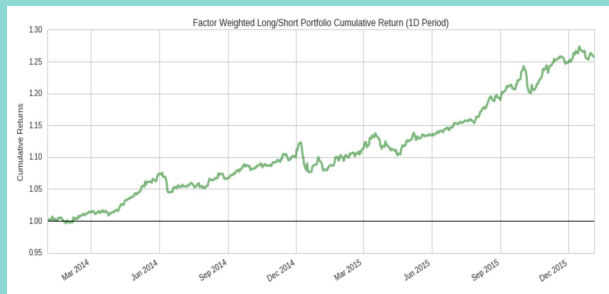
# Sentiment Analysis Alpha Factor

- Goal is to assess public opinion on a stock
- Found two academic papers that achieved success with a sentiment-based trading algorithm
- We used the Trader Mood and Sentdex data sets
- Best version used distance from mean and normalization



# Earnings Analysis Alpha Factor

- Attempts to profit off of the tendency for a security's returns to drift in the direction of an earnings surprise following an earnings announcement
- Well researched phenomenon known as Post-Earnings-Announcement Drift (PEAD)
- Used the Zack's Earnings Surprises dataset supplied by Zack's Investment Research





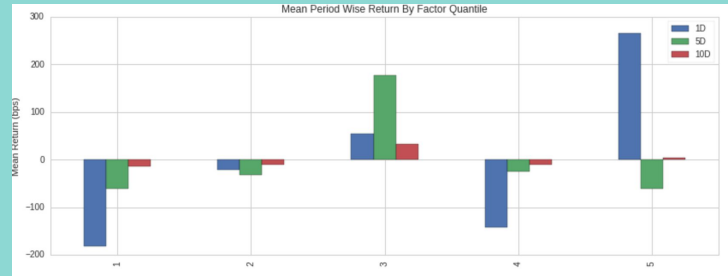
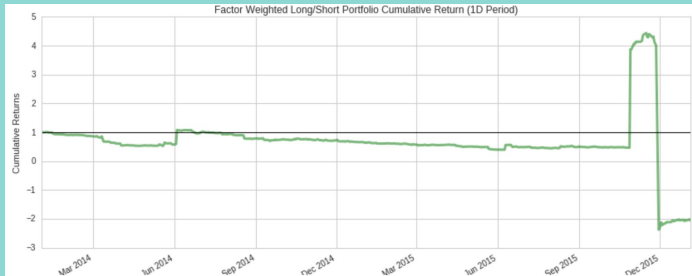
# Crossover Alpha Factor

- (Last day's closing price minus 10 day moving average) divided by 10 day moving average
- Normalized by negative absolute value
- Can be momentum or mean reversion based on normalization



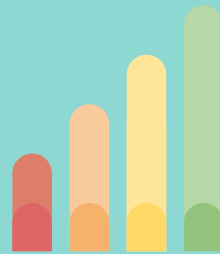
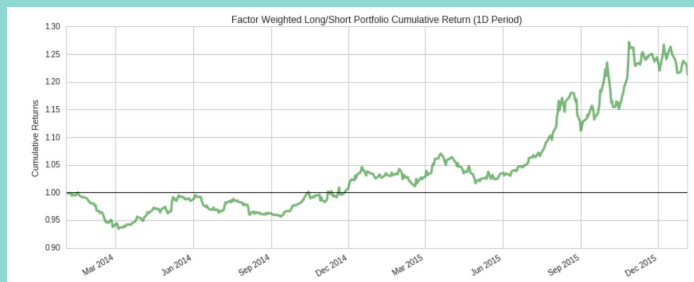
# Volume Momentum Alpha Factor

- Change in volume weighted change in price
- Increased momentum increases variability of returns
- Ended up not being sufficiently predictive to include in the final trading algorithm



# Linear Regression

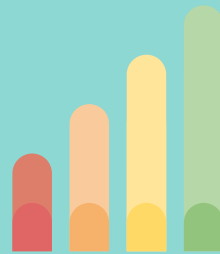
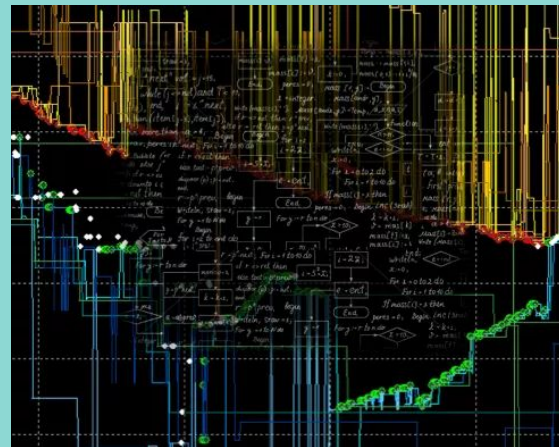
- This model converts our alpha factors into a single value that is used to rank stocks
- We created our own training set with data from 2014-2016
  - Expected values were the stock's 180 day returns
- Final weights were  $3.9\text{e-}6$  sentiment,  $-6.4\text{e-}3$  earnings,  $8.7\text{e-}5$  SMA





# The Algorithms

- Initialize (once)
- Make Pipeline (once)
- Before Trading Start (Daily)
- Rebalance (Daily)

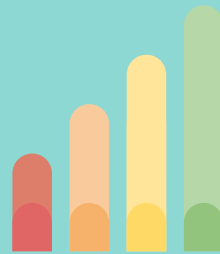




# Testing Environment



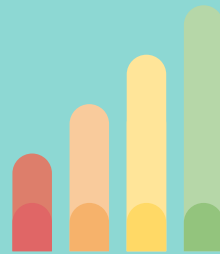
- Backtest on minute historical prices
- Train on 2014-2016, same as alpha factor
- Get detailed performance and risk data
- Test on 2016-2018 after setting hyperparameters
- Test returns may be better than training returns





# Adventures in the Real World

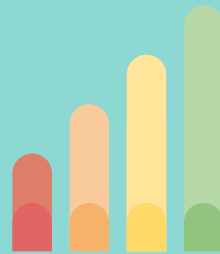
- Long-Short strategies aim to remove market variance from returns
- Ranking Accuracy (Prediction)  $\neq$  Returns
- Real-world constraints like turnover and fees affect returns
- Attempt to meet 2 structural constraints and 7 risk constraints





# Results

Algorithm	Training (2014-2016)	Testing (2016-2018)
Earnings	9.28%	.32%
Sentiment	-18.88%	-19.53%
Crossover	16.71%	-23.16%
Regression	-0.17%	-8.49%





# Future Plans

- Find ways to improve performance
- Code is open source, MIT License
- Quantopian is free to use if you want to trade!





# Questions?

