Technology: Python + Quantopian

Exploring Price and Earnings Momentum in the Tech Sector

Saloni Tripathi

April 23, 2019 || Discrete Time Models in Finance || Professor Mikhail Smirnov

Introduction: What is Momentum Trading?

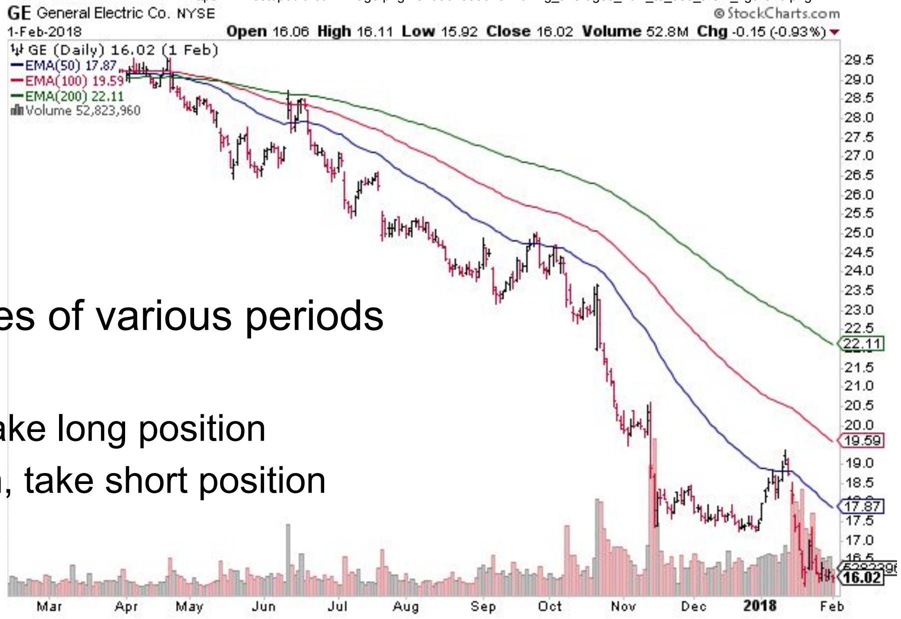
- Physics analogy: object with momentum is likely to continue its path
- Capture profits by investing in stocks likely to continue their trajectory
- Need to determine both trajectory and likelihood of continuation
- With this model, investors would enter a trade when the momentum begins to accelerate and exit their trade position when the momentum begins to fade
- Growth stocks with notable trading volumes offer favorable conditions for momentum trading, as they offer strong trends and often the potential of the price to either rise or fall can be 50%
    - Tech stocks

Price and Earnings Momentum Trading

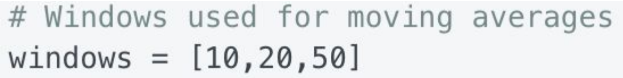
- **Price Momentum** refers to a stock’s tendency to continue its increase or decrease in prices
- **Earnings Momentum** refers to an acceleration or deceleration in corporate Earnings Per Share (EPS) growth relative to previous fiscal quarters or years which likely correlates to an acceleration or deceleration in stock price
- Growth trajectory of earnings tends to continue for several quarters/years
- Indicators: Volume, Moving Averages, Sentiment, Volatility, RSI, etc.

Price Momentum Model

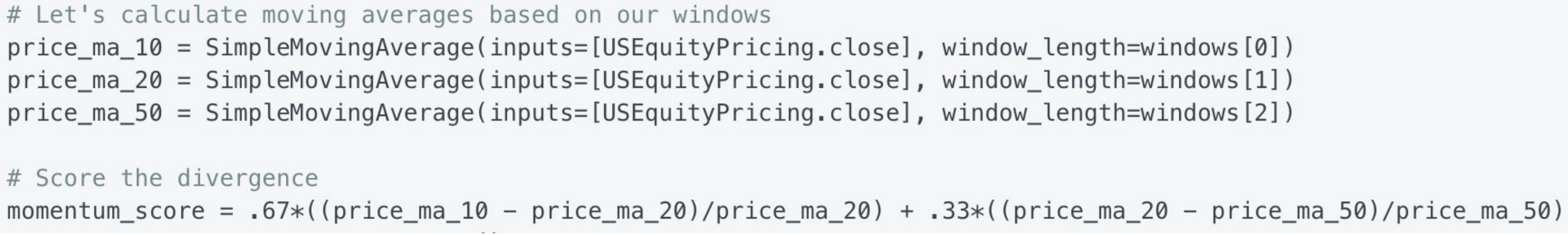
https://i.investopedia.com/image/png/1519964535878/moving_averages_how_to_use_them_figure2b.png



- Signal to trade = increasing trade volumes
- Signal to trade = diverging price moving averages of various periods
  - Windows used (10, 20, 50)
  - MA10 > MA20 > MA50 => upward momentum, take long position
  - MA10 < MA20 < MA50 => downward momentum, take short position



```
# Windows used for moving averages
windows = [10,20,50]
```



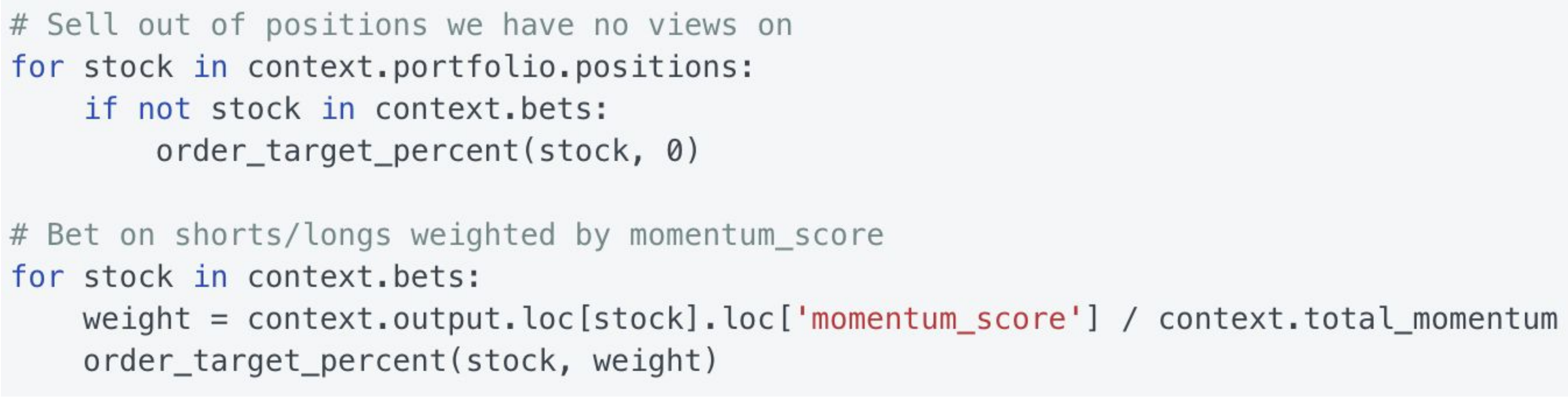
```
# Let's calculate moving averages based on our windows
price_ma_10 = SimpleMovingAverage(inputs=[USEquityPricing.close], window_length=windows[0])
price_ma_20 = SimpleMovingAverage(inputs=[USEquityPricing.close], window_length=windows[1])
price_ma_50 = SimpleMovingAverage(inputs=[USEquityPricing.close], window_length=windows[2])

# Score the divergence
momentum_score = .67*((price_ma_10 - price_ma_20)/price_ma_20) + .33*((price_ma_20 - price_ma_50)/price_ma_50)
```

- Upward momentum tends to get positive score, downward momentum negative score
- Magnitude determined by divergence of moving averages

Model Considerations and Position Management

- Volume, useful both as a signal and to ensure liquidity
  - Tech stocks with nontrivial trading volumes: Average dollar volume > $5mil
  - Increasing trading volumes => more investors joining => more likely trend continuation
- Stocks with reasonably high trading prices (ex: avoid penny stocks)
- We want to sell positions when price divergence weakens
- Weight of investment in each position is dependent on momentum_score

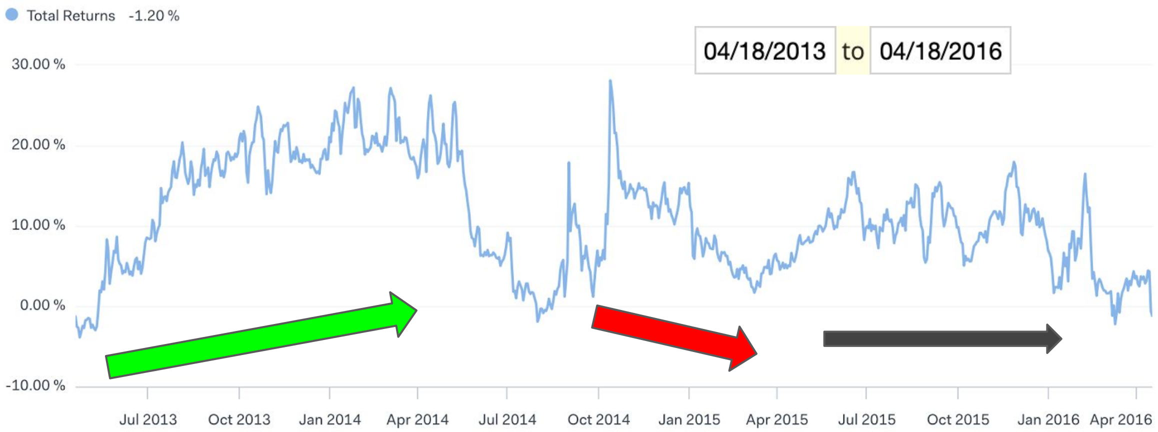


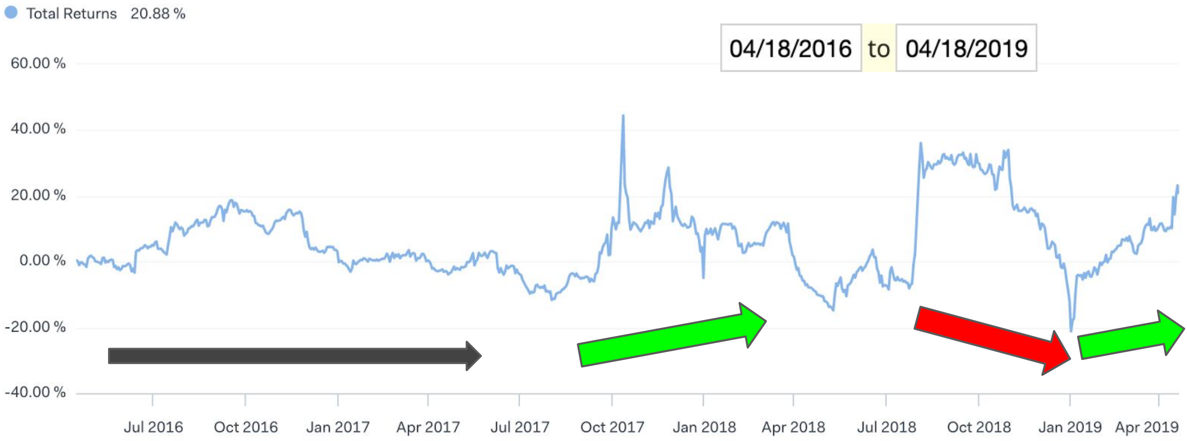
```
# Sell out of positions we have no views on
for stock in context.portfolio.positions:
    if not stock in context.bets:
        order_target_percent(stock, 0)

# Bet on shorts/longs weighted by momentum_score
for stock in context.bets:
    weight = context.output.loc[stock].loc['momentum_score'] / context.total_momentum
    order_target_percent(stock, weight)
```

Results & Takeaways

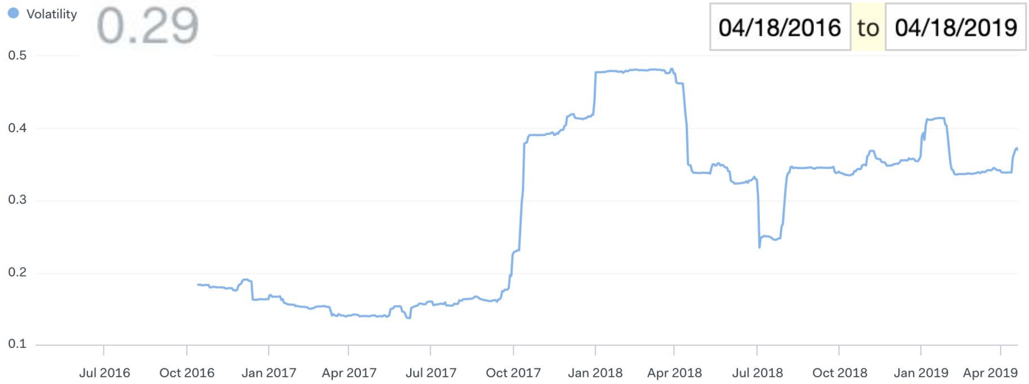
- 2013-2016 = 1.20% loss
- 2016-2019 = 20.88% gain
- SPY benchmark returned 43% 48%
- Fairly random results, but overall not a terrible strategy; there seem to be periods of gains and losses
- Future work: indicators for market conditions that correspond with gains and losses, fine-tuning parameters and including other momentum signals

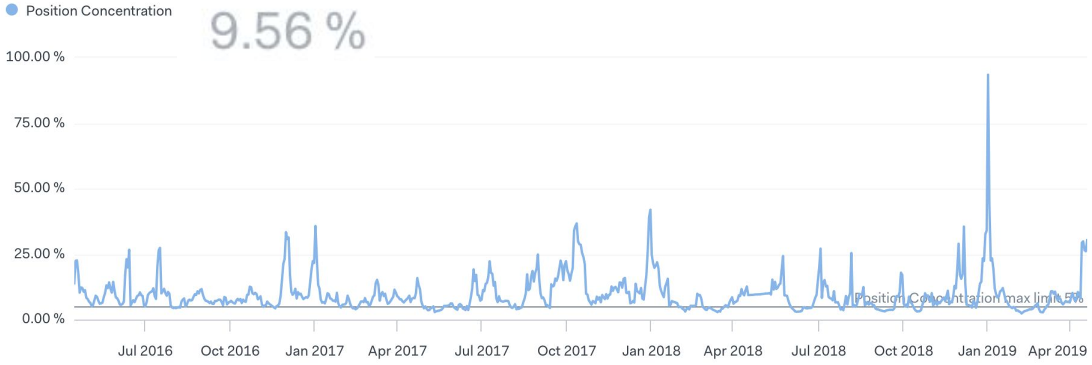


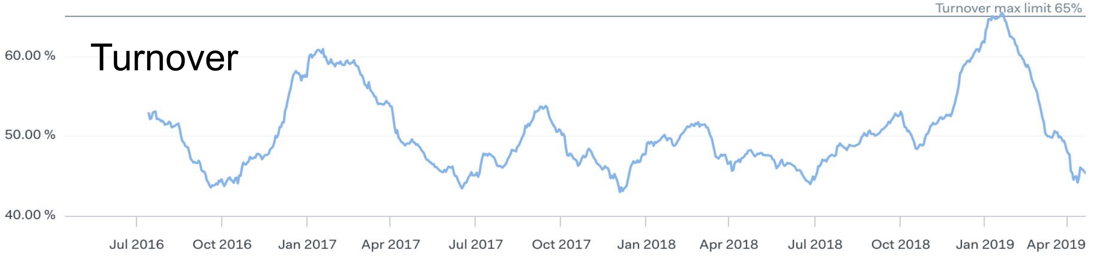


Results & Takeaways

- Relatively high volatility
- Typically invested in various long/short positions (only fully invested in one bet at one point in time)
- Fairly high turnover (rate at which assets are being bought and sold within the portfolio), which will hurt more when commission fees incorporated







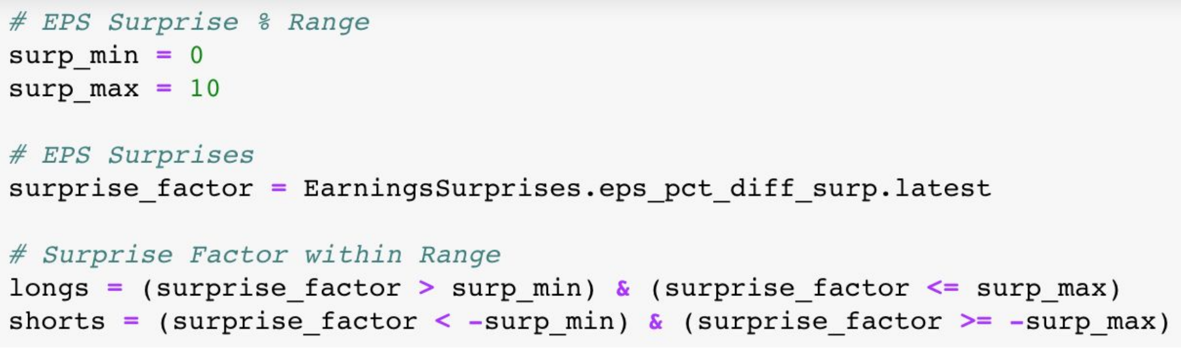
Post-Earnings Announcement Drift (PEAD) Model

- There is a common belief that earnings momentum and price momentum are strongly connected
- While investigating the effects of earnings announcements on stock prices, we came across an interesting effect called Post-Earnings Announcement Drift (PEAD)
- **PEAD:** Earnings announcements that cause a positive/negative surprise will lead to positive/negative cumulative abnormal returns following the announcement
- First proposed by Ray Ball and Philip R. Brown

Source Paper: Brandt, Kishore, Santa-Clara, Venkatachalam: Earnings Announcements are Full of Surprises

Model Considerations and Position Management

- **EPS Surprise %** = ((Actual EPS - Estimated EPS) / Absolute Estimated EPS) *100
- Buy stocks with Earnings Surprise the day after Earnings Announcement
    - 10% positive/negative surprise
- Hold stocks for 45 days
- Monitor their performance every day:
    - If over 10% profit sell
    - If over 5% loss sell

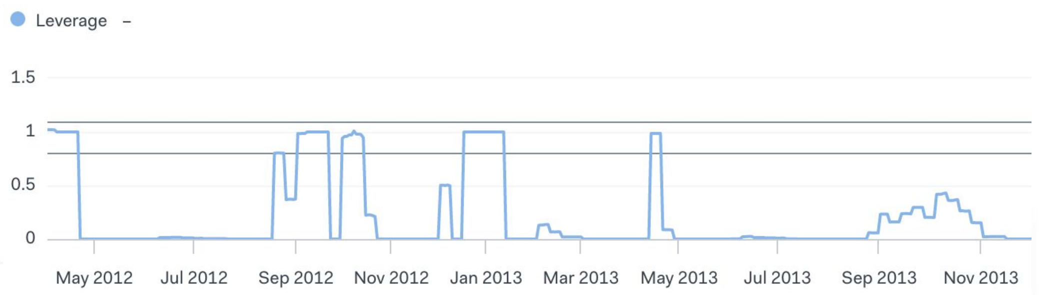


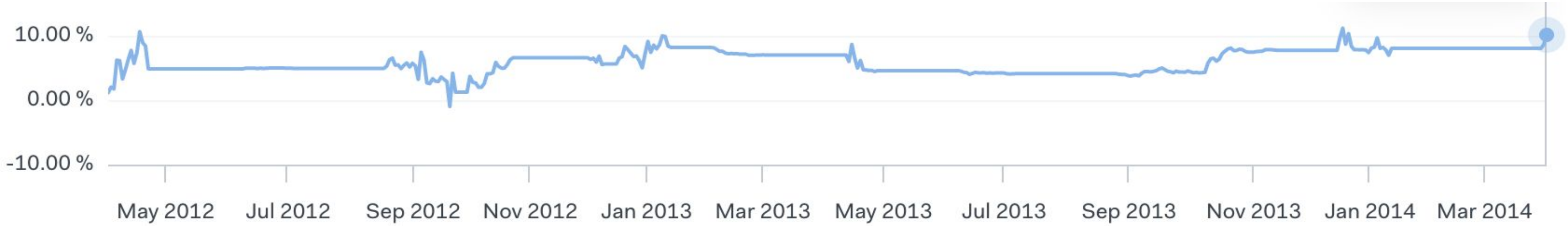
```
# EPS Surprise % Range
surp_min = 0
surp_max = 10

# EPS Surprises
surprise_factor = EarningsSurprises.eps_pct_diff_surp.latest

# Surprise Factor within Range
longs = (surprise_factor > surp_min) & (surprise_factor <= surp_max)
shorts = (surprise_factor < -surp_min) & (surprise_factor >= -surp_max)
```

Results & Takeaways





- 10.14% total returns over two years
- Short-term volatility is seemingly correlated with the surprise factor
- PEAD Trading generally has high volatility
- By limiting the EPS Surprise % and profits and losses on each position, we attempted to reduce the volatility of our model
- Higher percentage surprises:
  - Lead to sudden leaps that usually return to equilibrium within a day or two

Sentiment and IPOs

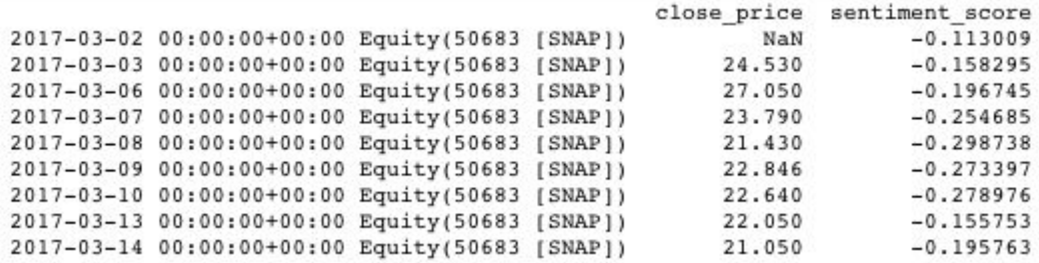
- Sentiment analysis is the automated process of understanding an opinion about a given subject from written or spoken language
  - Subproblem of NLP
- StockTwits captures sentiment for us, via #bullish and #bearish tweets
  - They calculate a percentage and assign companies a sentiment score
- We think we can use SA as a basis for IPO trading strategy

Hypothesis: when there is a new stock entering the exchange as an IPO, there's no historical data for that company on the exchange

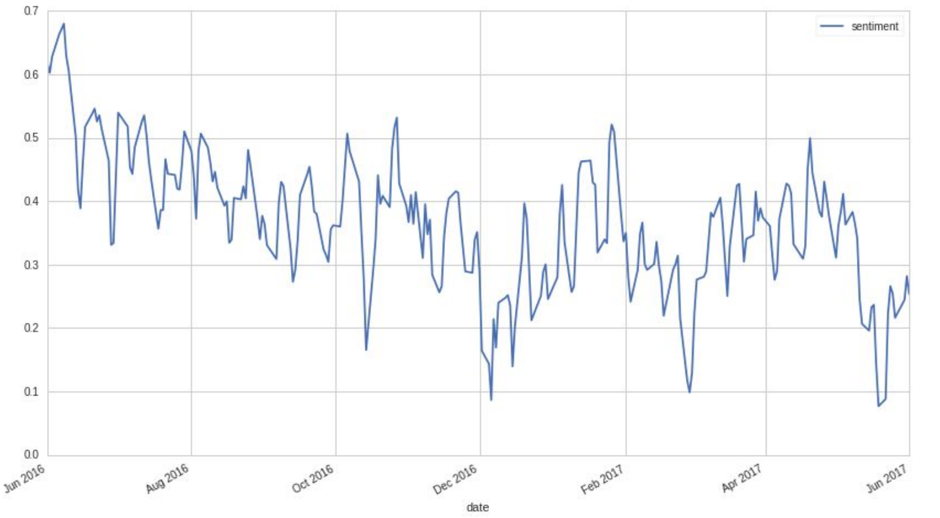
- we think that the price movements over the first three months are going to be heavily due to public sentiment
- want to be able to capture sentiment in the market and use that as a signal to create a long/short strategy

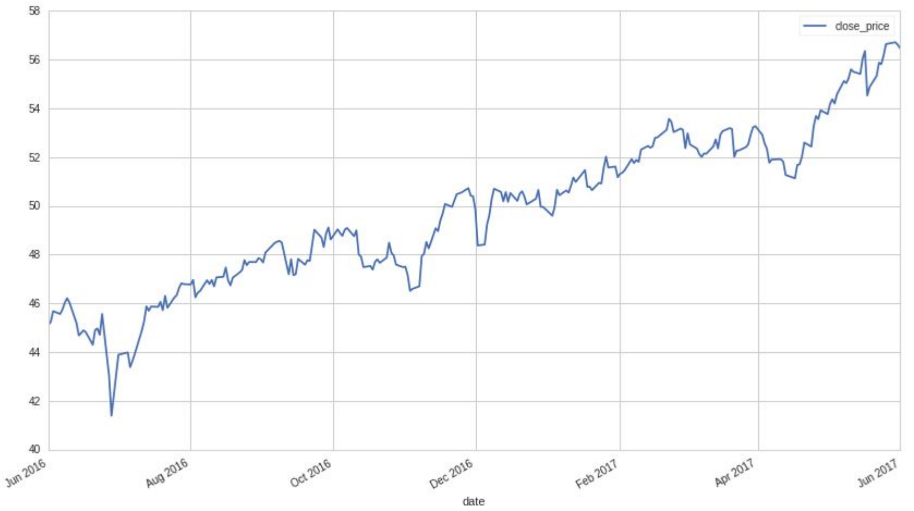
How do we do this?

- Come up with a daily sentiment score for the entire tech sector of stocks → basically like an average of the sentiment for that specific stock, over a trailing 3 day window
- Ex.) if we have 500 total tech stocks in the sector, then we will have 500 different sentiment scores PER DAY
  Then take those 500 and compute the average of them for every single day
  --> gives us one score that summarizes the overall sentiment towards the technology sector, for that day.
- We chart that score over time



```
                                                   close_price  sentiment_score
2017-03-02 00:00:00+00:00 Equity(50683 [SNAP])          NaN        -0.113009
2017-03-03 00:00:00+00:00 Equity(50683 [SNAP])       24.530        -0.158295
2017-03-06 00:00:00+00:00 Equity(50683 [SNAP])       27.050        -0.196745
2017-03-07 00:00:00+00:00 Equity(50683 [SNAP])       23.790        -0.254685
2017-03-08 00:00:00+00:00 Equity(50683 [SNAP])       21.430        -0.298738
2017-03-09 00:00:00+00:00 Equity(50683 [SNAP])       22.846        -0.273397
2017-03-10 00:00:00+00:00 Equity(50683 [SNAP])       22.640        -0.278976
2017-03-13 00:00:00+00:00 Equity(50683 [SNAP])       22.050        -0.155753
2017-03-14 00:00:00+00:00 Equity(50683 [SNAP])       21.050        -0.195763
```





- For every day of the year, we have 2 numbers: avg sentiment and avg closing price for the entire sector
- We chart both of them over time to see if there is a correlation b/w the two
    - i.e. does rise/dip in the sentiment chart reflect a subsequent rise/dip in average closing price

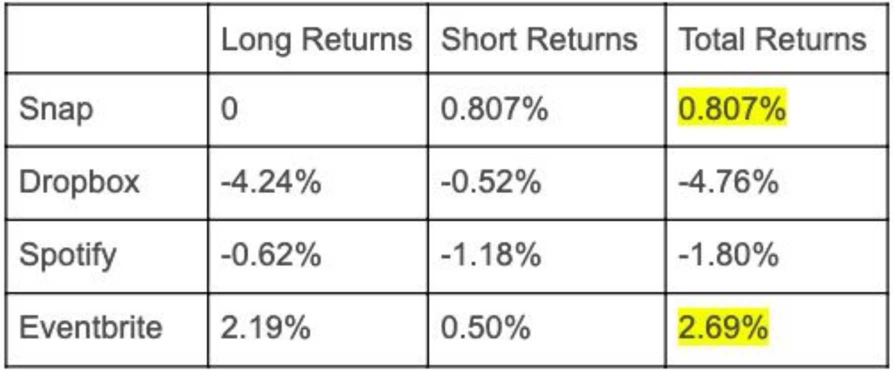
Simple long/short strategy using the above research:

- Buy a new IPO stock when the sentiment is higher than a threshold value
- Short when the sentiment is lower than a threshold
  - (thresholds chosen experimentally)
- Keep position for 3 day periods, then sell (using 3 day sentiments because the effects of public sentiment are short term)
- We calculate and report the average returns, only trading within the first 3 months of a stock’s issuance onto the exchange

→ Conclusions

We found that two of the 4 IPOs under study produced positive returns. In the case of Snapchat, we saw consistently negative sentiment, which is why we see mostly short positions taken. We believe that the consistency in the sentiment over the 3 month period helped produce the positive returns. Since Snapchat is also a very public and polarizing consumer software product, we expected its performance to be more closely tied to sentiment, than perhaps dropbox, which is less talked about in the general media. We believe this explains the rather poor performance when applying this strategy for dropbox and spotify.

While Snap, Dropbox, and Spotify have seen large fluctuations in their earnings growth over their first year (often over 100% growth year over year) Eventbrite has not. It may be the case that slower growth stocks perform better with this long/short strategy we've implemented. This would be an interesting area of further study.



	Long Returns	Short Returns	Total Returns
Snap	0	0.807%	0.807%
Dropbox	-4.24%	-0.52%	-4.76%
Spotify	-0.62%	-1.18%	-1.80%
Eventbrite	2.19%	0.50%	2.69%