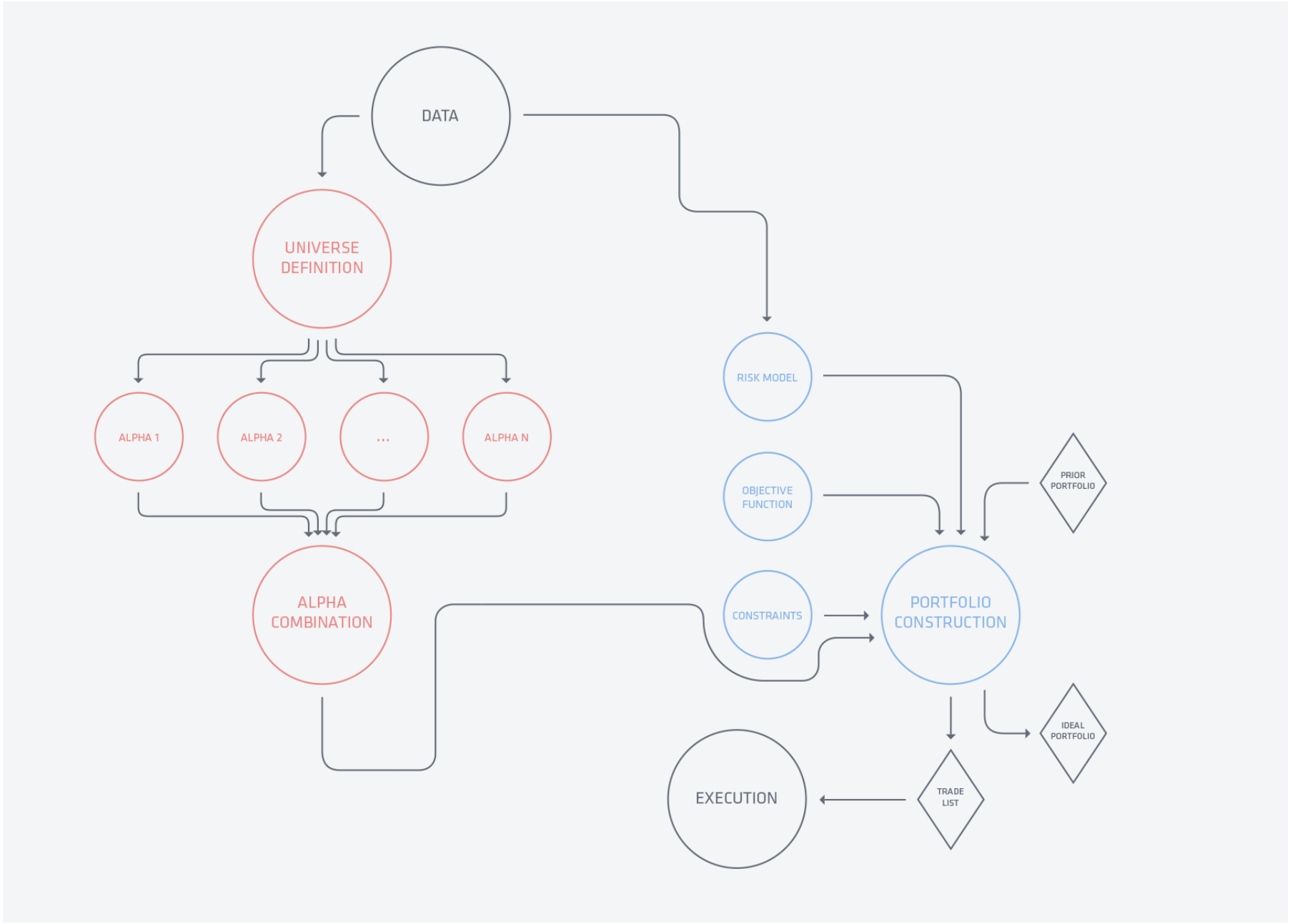
Data Science in 30 minutes: Algorithmic Trading workflow and potential machine learning applications

Jess Stauth PhD, VP Quant Strategy
Thomas Wiecki PhD, Director of Data Science

Quantopian

Disclaimer

This presentation is for informational purposes only and does not constitute an offer to sell, a solicitation to buy, or a recommendation for any security; nor does it constitute an offer to provide investment advisory or other services by Quantopian, Inc. ("Quantopian"). Nothing contained herein constitutes investment advice or offers any opinion with respect to the suitability of any security, and any views expressed herein should not be taken as advice to buy, sell, or hold any security or as an endorsement of any security or company. In preparing the information contained herein, Quantopian has not taken into account the investment needs, objectives, and financial circumstances of any particular investor. Any views expressed and data illustrated herein were prepared based upon information, believed to be reliable, available to Quantopian at the time of publication. Quantopian makes no guarantees as to their accuracy or completeness. All information is subject to change and may quickly become unreliable for various reasons, including changes in market conditions or economic circumstances.

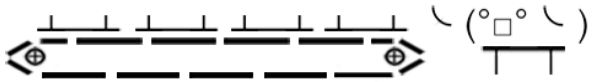

DATA
UNIVERSE DEFINITION
ALPHA 1
ALPHA 2
...
ALPHA N
ALPHA COMBINATION
RISK MODEL
OBJECTIVE FUNCTION
CONSTRAINTS
PRIOR PORTFOLIO
PORTFOLIO CONSTRUCTION
IDEAL PORTFOLIO
TRADE LIST
EXECUTION


DATA

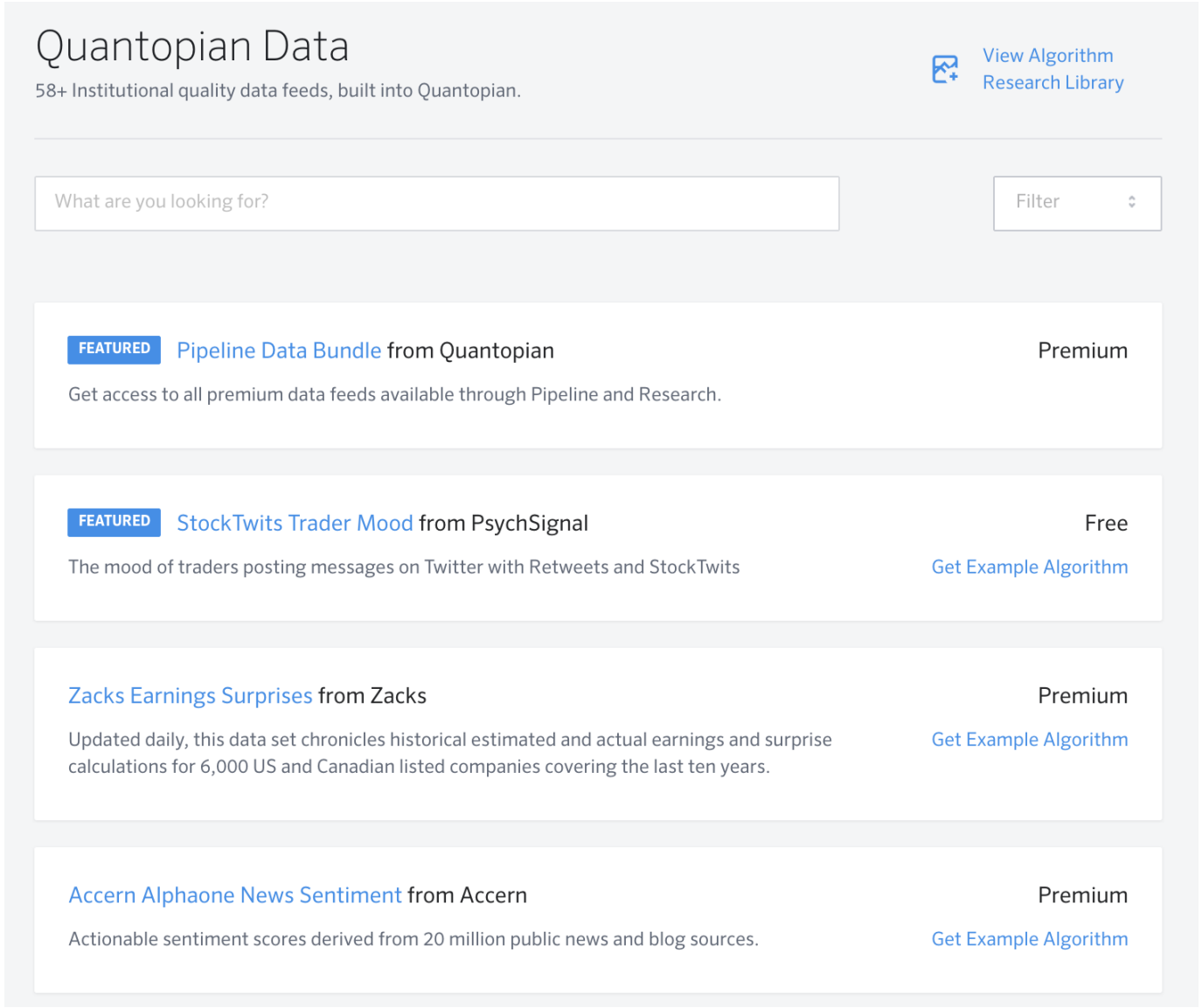
“Which dataset(s) do I think contain information that will help predict future returns?”

Quants  Data

- Market data: price and volume (quote-level, trade-level, minute bars or daily)
- Non-market classical data:
  - Corporate Fundamentals
  - Sell-side estimates
  - Sector classifications
  - Events (corp events, M&A, earnings, etc)
  - Risk model
  - Short Interest
  - Ownership filings
  - …
- ‘New world’ data
  - Tagged or processed news
  - Social media sentiment
  - Satellite imagery
  - Credit card panel
  - …

Quants  Data

- Real-world data is dirty, incomplete and often survivorship-biased.
  - Data cleaning - handle missing values, outliers, errors
  - Normalization / Standardization
  - Symbol mapping / Joining across vendors
  - Buy/Collect/Maintain “Point in Time” data
- Cross-sectional coverage
- Historical coverage
- Sourcing - Cost of purchasing or collecting data has to be less than the expected profit



quantopian.com/data

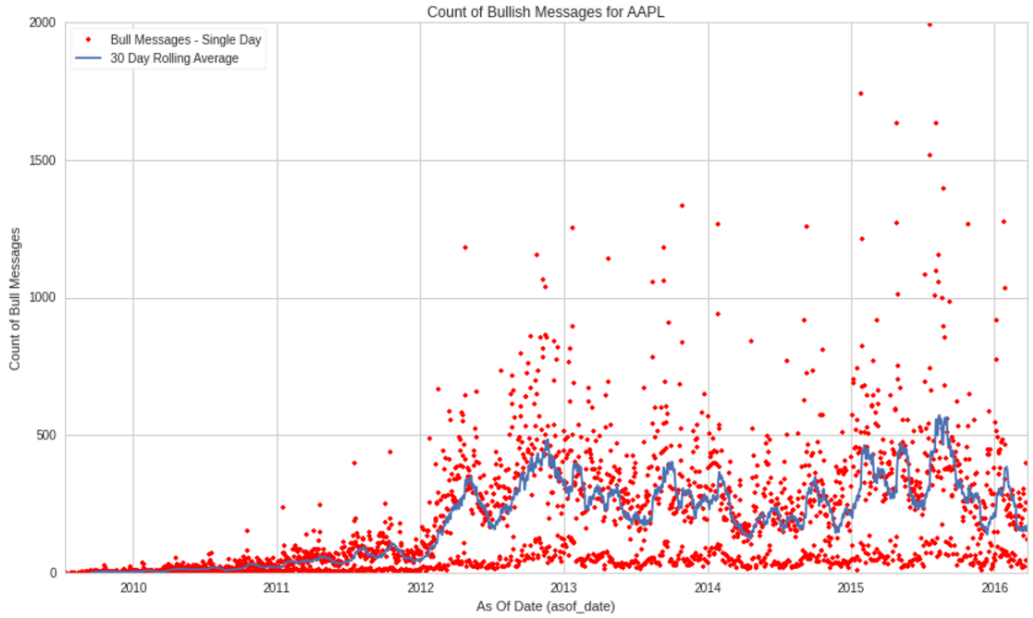
PsychSignal: StockTwits Trader Mood (All Fields)

In this notebook, we'll take a look at PsychSignal's *StockTwits Trader Mood (All Fields)* dataset, available on the Quantopian Store. This dataset spans 2009 through the current day, and documents the mood of traders based on their messages.

We can select columns and rows with ease. Below, we'll fetch all rows for Apple (sid 24) and explore the scores a bit with a chart.

In [5]:
```
# Filtering for AAPL
aapl = dataset[dataset.sid == 24]
aapl_df = odo(aapl.sort('asof_date'), pd.DataFrame)
plt.plot(aapl_df.asof_date, aapl_df.bull_scored_messages, marker='.', linestyle='None', color='r')
plt.plot(aapl_df.asof_date, pd.rolling_mean(aapl_df.bull_scored_messages, 30))
plt.xlabel("As Of Date (asof_date)")
plt.ylabel("Count of Bull Messages")
plt.title("Count of Bullish Messages for AAPL")
plt.legend(["Bull Messages - Single Day", "30 Day Rolling Average"], loc=2)
```

Out[5]: <matplotlib.legend.Legend at 0x7fb5059cee90>



quantopian.com/data


UNIVERSE
DEFINITION

“Over what set of tradable instruments is my returns forecast relevant?”

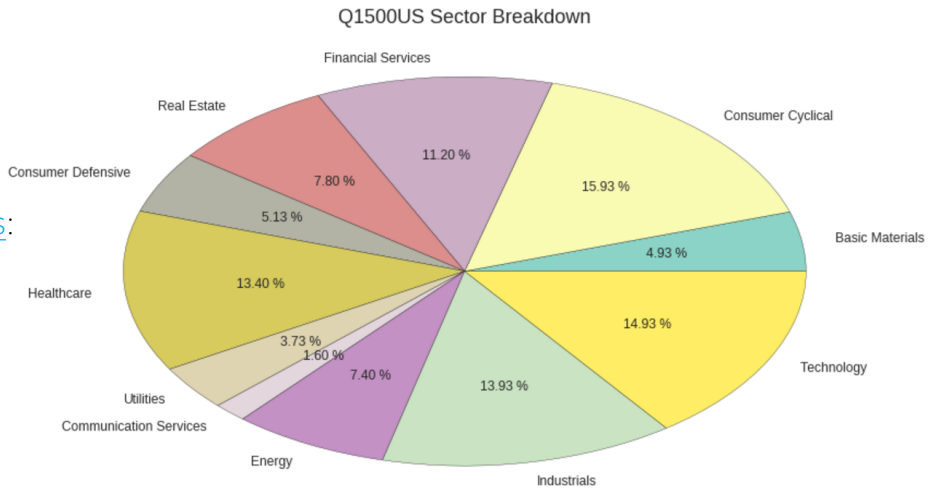
Can I trade it? Should I trade it? Considerations for Defining a Tradable Universe

- Some Practical considerations (aka “Can I trade it?”):
  - Liquidity - e.g. identify a lower bound for the average daily dollar volume.
  - Hard-to-Trade instruments - e.g. eliminate instruments like over-the-counter (OTC) securities
  - Universe size
  - Turnover
- Some Strategy/Style specific considerations (aka “Should I trade it?”):
  - Data coverage - e.g. a strategy that relies on news sentiment may have very limited data for small cap stocks.
  - Sector specific - e.g. a strategy that trades on financial statement data, such as the accruals anomaly doesn’t generalize well to bank stocks.

Alternatives for the lazy:

- Subscribe to an index provider (e.g. Russell, MSCI) $$$
- If you use Quantopian check out the new built-in universes:

```
from quantopian.pipeline.filters import Q1500US
from quantopian.pipeline.filters import Q500US
```



Alpha Discovery

Alpha factors express a predictive relationship between some given set of information and future returns.

Alpha research is an art and a science.

The (vastly simplified) science part:

1. Form a hypothesis
2. Test it.
3. Analyze results*.
4. Reject (or accept) hypothesis
5. Rinse and repeat.

*Quantopian recently launched an open source Python package, **alphalens**, for performance analysis of alpha factors, including:

- Returns Analysis
- Information Coefficient Analysis
- Turnover Analysis
- Sector Analysis

Example Tear Sheet

Example factor courtesy of ExtractAlpha

Returns Analysis

	1	5	10
Ann. alpha	0.085	0.034	0.023
t-stat(alpha)	17.359	15.144	14.664
beta	0.040	0.046	0.046
Mean Daily Return Top Quantile (bps)	8.807	3.844	2.475
Mean Daily Return Bottom Quantile (bps)	-8.179	-3.723	-2.695
Mean Daily Spread (bps)	17.026	7.576	5.200

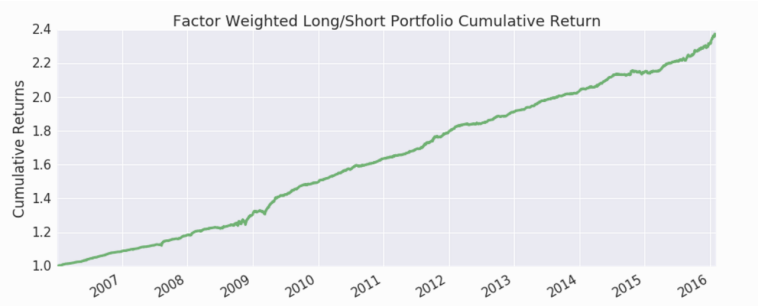
Information Analysis

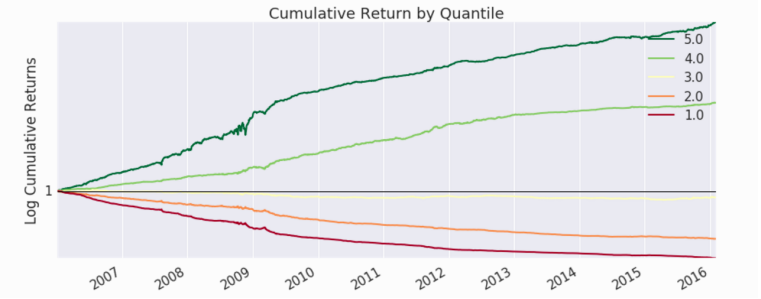
	1	5	10
IC Mean	0.013	0.015	0.016
IC Std.	0.056	0.063	0.063
t-stat(IC)	11.951	12.122	12.327
p-value(IC)	0.000	0.000	0.000
IC Skew	0.121	0.009	0.004
IC Kurtosis	1.703	1.400	1.305
Ann. IR	3.768	3.822	3.887

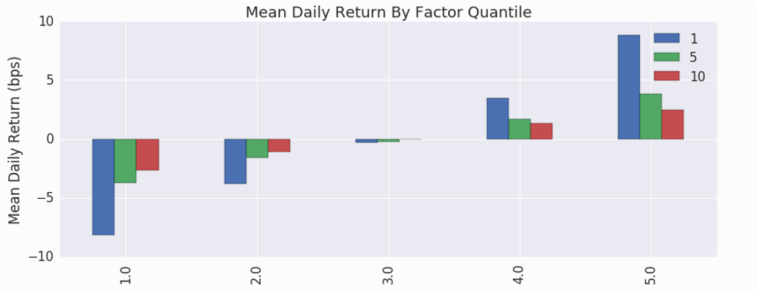
Turnover Analysis

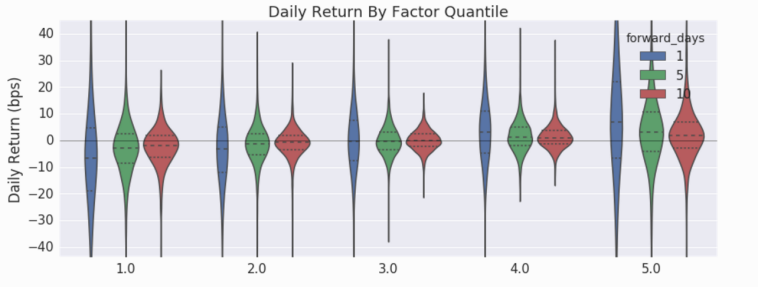
	Top Quantile	Bottom Quantile
Mean Turnover	0.44	0.452

```
Mean Factor Rank Autocorrelation    0.625
dtype: float64
```



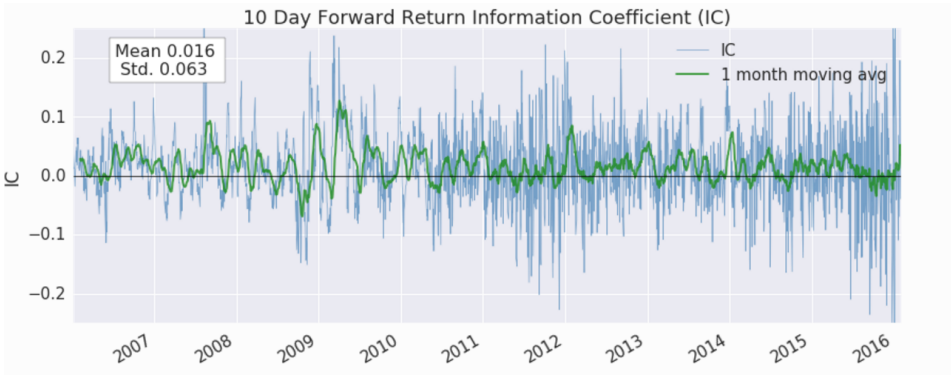


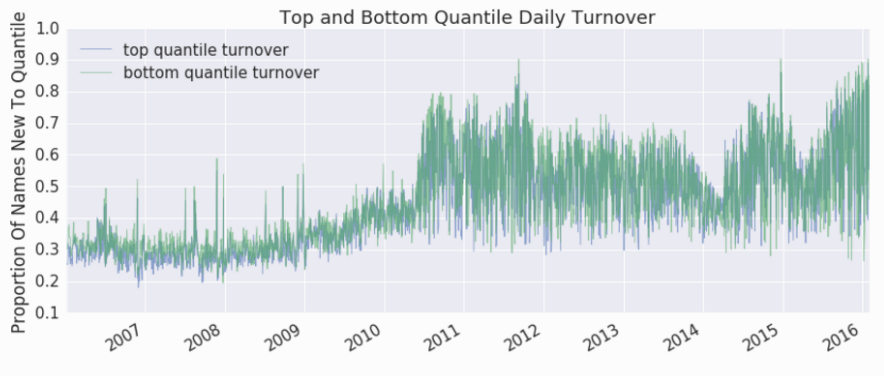


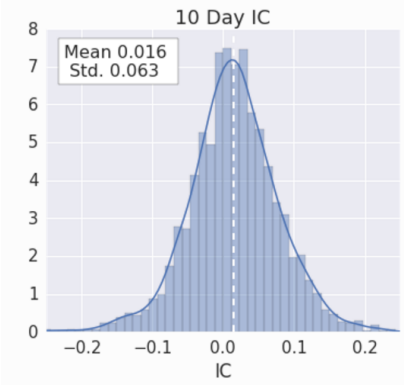


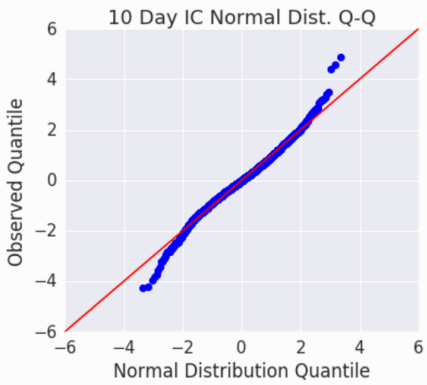
quantopian.github.io/alphalens/

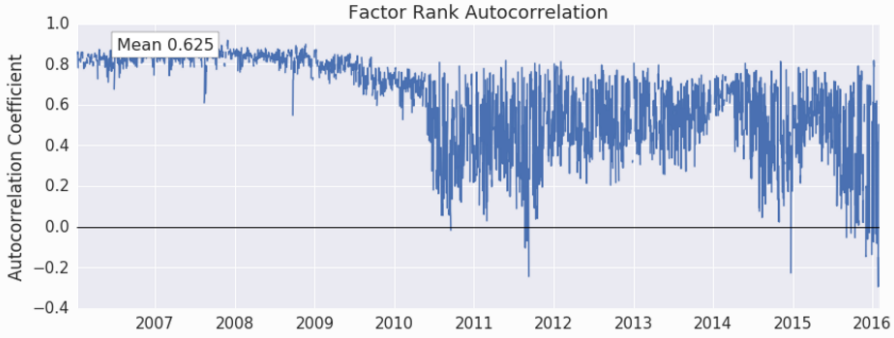
The cumulative returns and mean daily returns presented are gross of fees and do not reflect the deduction of any management fees, performance fees or other expenses. Past performance is not necessarily reflective of future results.

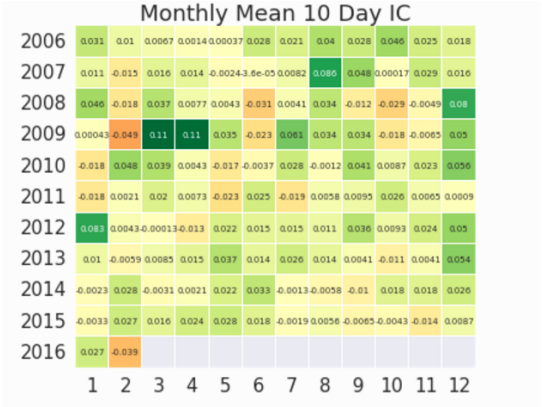

10 Day Forward Return Information Coefficient (IC)
Mean 0.016
Std. 0.063
IC
1 month moving avg
IC

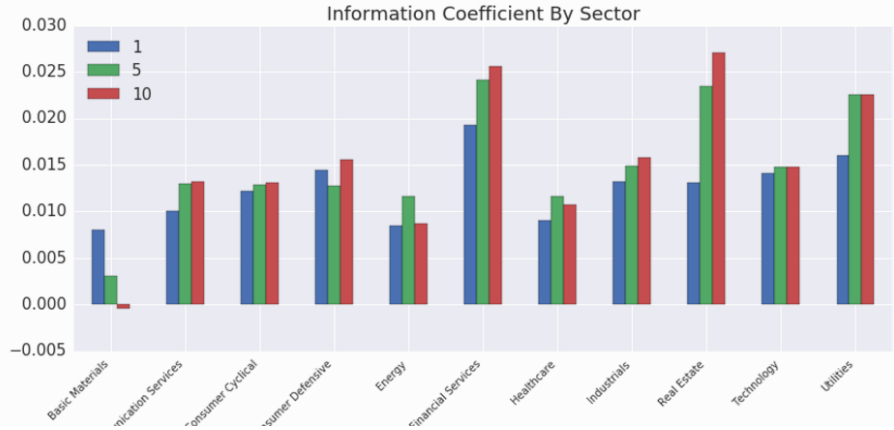

Top and Bottom Quantile Daily Turnover
top quantile turnover
bottom quantile turnover
Proportion Of Names New To Quantile


10 Day IC
Mean 0.016
Std. 0.063
IC


10 Day IC Normal Dist. Q-Q
Observed Quantile
Normal Distribution Quantile


Factor Rank Autocorrelation
Mean 0.625
Autocorrelation Coefficient


Monthly Mean 10 Day IC


Information Coefficient By Sector
1
5
10
Basic Materials
Consumer Cyclical
Energy
Financial Services
Healthcare
Industrials
Real Estate
Technology
Utilities

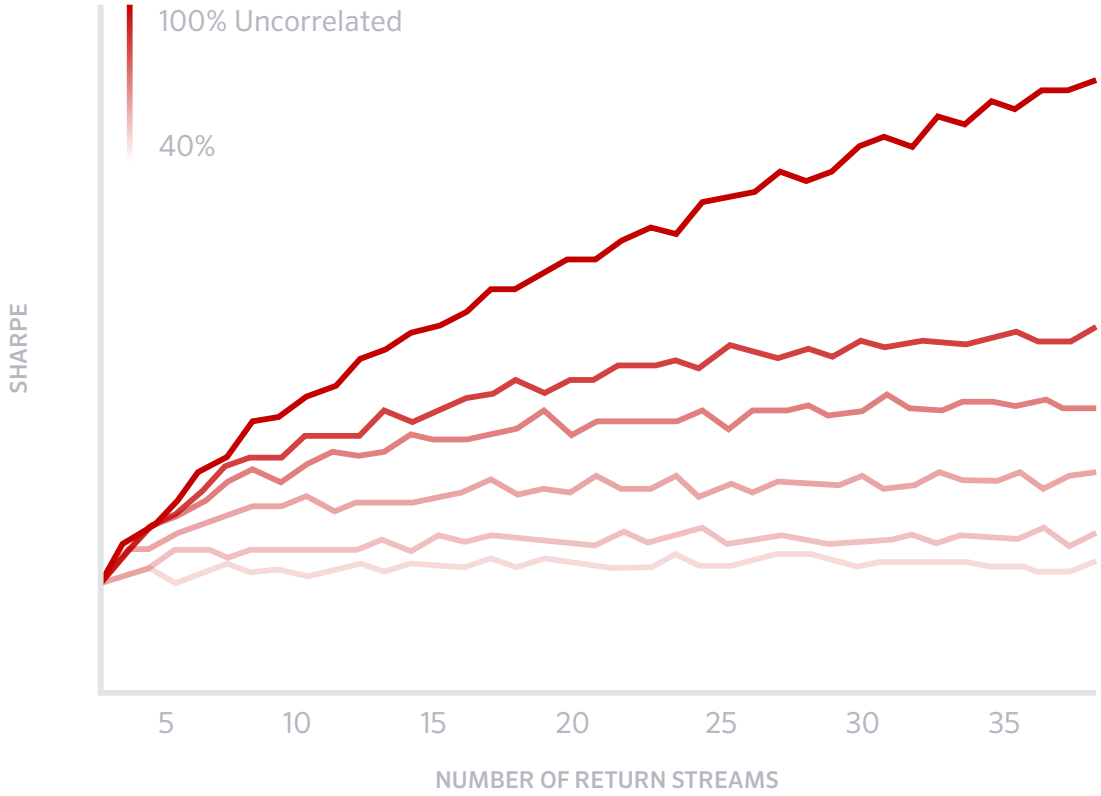
The art part...


ALPHA
COMBINATION

In today's markets, rarely is any single alpha significant enough to be the sole basis of an investment strategy

Why combine alphas?

To the extent you are able to identify alphas with independent returns streams, even a simple linear combination strategy will reap the benefits of diversification.



Sharpe Ratio is a statistical measurement of the risk adjusted performance of a portfolio, and is calculated by dividing a portfolio's excess return over the risk-free rate by the standard deviation of its returns. It shows a portfolio's reward per unit of risk and is useful when comparing two similar investment strategies. As the Sharpe Ratio increases, the better its risk adjusted performance.

How combine alphas?

If we view alpha combination as a classic classification (or regression) problem, Machine Learning is an intuitive choice.

ML is very good at solving and coming up with an alpha combination that is predictive.

Overview

1. Define trading universe.
2. Define alphas.
3. Run pipeline.
4. Split into train and test set.
5. Preprocess data (rank, subsample, align with future returns, impute, scale).
6. Train Machine Learning classifier (AdaBoost from Scikit-Learn[1]).
7. Evaluate Machine Learning classifier on test set.

Download the companion notebook from the DS30 repo, upload and run from Quantopian Research portal, or clone the notebook directly from Thomas Wiecki's original post.

[1] scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html


PORTFOLIO
CONSTRUCTION

Portfolio construction is where the ideal meets reality.

Portfolio construction starts with three basic questions

1. **What risks can you account for?** e.g. market exposure, sector exposure, liquidity risk.
2. **What is your objective function, or what is the stat you want to maximize?** e.g. Sharpe ratio
3. **What practical constraints will you impose?** e.g. a maximum percent of the portfolio should be invested in any single stock.

Combined with your alpha signal, answers to these questions allow you to run a **constrained optimization** to generate your target portfolio.

The Python ecosystem has a number of powerful libraries that can be used for solving general optimization problems:

- scipy.optimize - https://docs.scipy.org/doc/scipy/reference/optimize.html
- CVXOPT - http://cvxopt.org/
- CVXPY - http://www.cvxpy.org/en/latest/

Preview: optimize interface module coming to Q

API Overview

The `optimize` module has three major components in this release:

1. `calculate_optimal_portfolio`, a top-level entrypoint.
2. `Objective` classes, representing functions to be minimized or maximized by the optimizer.
3. `Constraint` classes, representing constraints to be enforced by the optimizer.

Lists of the currently available objectives and constraints can be found under `optimize.objectives` and `optimize.constraints`, respectively.

To run a portfolio optimization, you call `calculate_optimal_portfolio` and provide three values:

- An `Objective` to optimize.
- A list of `Constraints` to enforce.
- A pandas Series containing weights for the current portfolio. The index of the current portfolio series defines the assets that are allowed in the target portfolio.

```
Calculate portfolio weights optimizing an objective subject to constraints.

Parameters
----------
objective : Objective
    The objective function to optimize.
constraints : list[Constraint]
    List of constraints on the output portfolio.
current_portfolio : pd.Series
    A Series containing the current portfolio weights, expressed as
    percentages of the portfolio's liquidation value.

    The index of ``current_portfolio`` defines what assets are available
    for the output portfolio. Assets that are under consideration but not
    currently held should be provided with a weight of 0.
```

```
Returns
-------
optimal_portfolio : pd.Series
    A Series indexed like ``current_portfolio`` containing new portfolio
    weights. Weights should be interpreted in the same way as
    ``current_portfolio``.

Raises
------
InfeasibleConstraints
    Raised when there is no possible portfolio that satisfies the received
    constraints.
UnboundedObjective
    Raised when the received constraints are not sufficient to put an upper
    (or lower) bound on the calculated portfolio weights.
```

Learn more and share feedback: https://www.quantopian.com/posts/request-for-feedback-portfolio-optimization-api


EXECUTION

Implementation questions that can help define your best execution strategy.

- How fast do I need to trade?
- How quickly does the predictive power of the alpha decay?
- Does it make more sense to be passive and execute slowly in the market, or, conversely, does it make more sense to execute aggressively and immediately?

Thank you.

jstauth@quantopian.com

Content heavily borrowed from Q CIO Jonathan Larkin's Blog:
blog.quantopian.com/a-professional-quant-equity-workflow/
And from Thomas Wiecki's Post on Machine Learning (including notebook):
quantopian.com/posts/machine-learning-on-quantopian

Open Source libraries:

Backtester - zipline.io
Alpha Discovery - github.com/quantopian/alphalens
Portfolio Analysis - github.com/quantopian/pyfolio