Machine Learning Trading Strategy Development in Python, using ***zipline*** and ***pyfolio***

Justin Lent
Director, Hedge Fund Development @ Quantopian
justin@quantopian.com

Startup.ML Workshop – San Francisco
May 2016

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, Inc. 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, Inc. 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.

Overview

1. Intro: Who is Quantopian?
   - Provide quants and programmers free tools like ***zipline*** and ***pyfolio***, along with free market data for developing trading algorithms
   - Crowd-sourced quantitative investment manager
     - We make allocations to qualified algorithms, and we share any profits with the author.

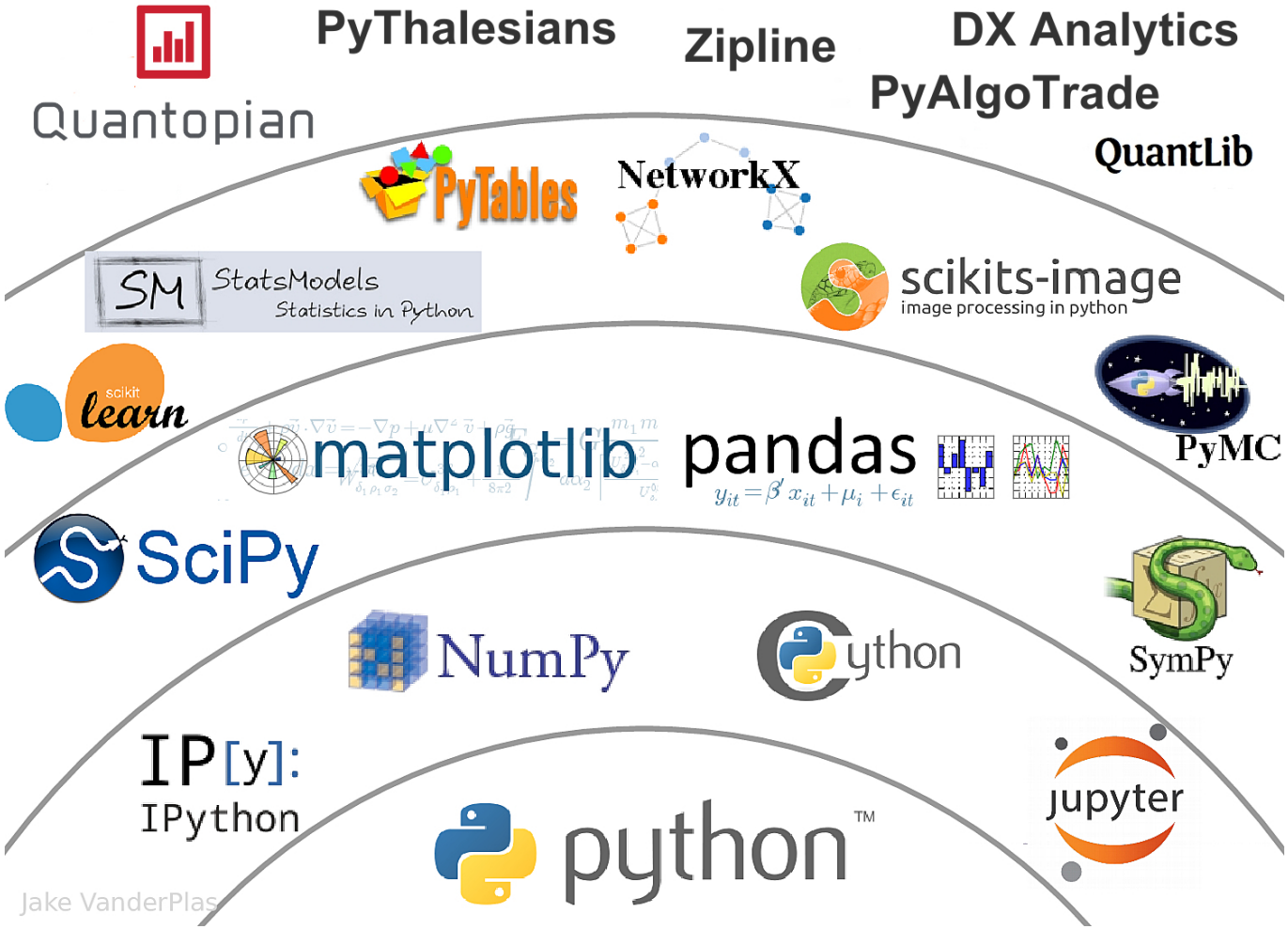
2. Python in Quant Finance

3. What are zipline & pyfolio?
   - Backtester and Portfolio Risk Analysis tools
   - Zipline
     - Open source and free: Apache v2 license
     - https://github.com/quantopian/zipline
     - Tutorial/Docs: http://www.zipline.io/index.html
   - Pyfolio
     - Open source and free: Apache v2 license
     - http://github.com/quantopian/pyfolio
     - Tutorial/Docs: http://quantopian.github.io/pyfolio/

Why use Python for Quant Finance?

- Python is a general purpose language
- No hodge-podge of perl, bash, R, matlab, fortran, Excel
- Gives us access to a vibrant, rapidly expanding ecosystem of tools…
- *Very easy to learn*

The Quant Finance PyData Stack



- Source: [Jake VanderPlas: State of the Tools]
  - (https://www.youtube.com/watch?v=5GlNDD7qbP4)

Zipline + pyfolio

- ***Zipline***: open-source backtester by Quantopian

- Powers Quantopian.com
  - Various models for transaction costs and slippage.
  - Web based IDE for creating and deploying trading algorithms

- Hosted ipython notebook research server
  - Ad-hoc data analysis. We provide market data.
  - Pull in strategy backtest results from the Web IDE and use ***pyfolio***

Using zipline & pyfolio stand-alone

- Installation
- Use Anaconda to get a Python system with the full PyData ecosystem.
- You can conda install the zipline package which includes pyfolio.
  - *conda install -c Quantopian zipline*
  - More info: https://conda.anaconda.org/quantopian
- Just want pyfolio? *pip install pyfolio*

- Import / Usage (typical)
  - Zipline: import *TradingAlgorithm* class, and individual zipline specific API functions (based on specific usecase)

```
from zipline import TradingAlgorithm
from zipline.api import order_target, record, symbol, history, add_history, order_target_percent
from zipline.api import schedule_function, date_rules, time_rules, order, get_open_orders, get_datetime
from zipline.api import set_slippage, set_commission
from zipline.api import slippage
from zipline.api import commission

from zipline.utils import tradingcalendar
```

  - Pyfolio

```
import pyfolio as pf
```

What is zipline?

Zipline Examples

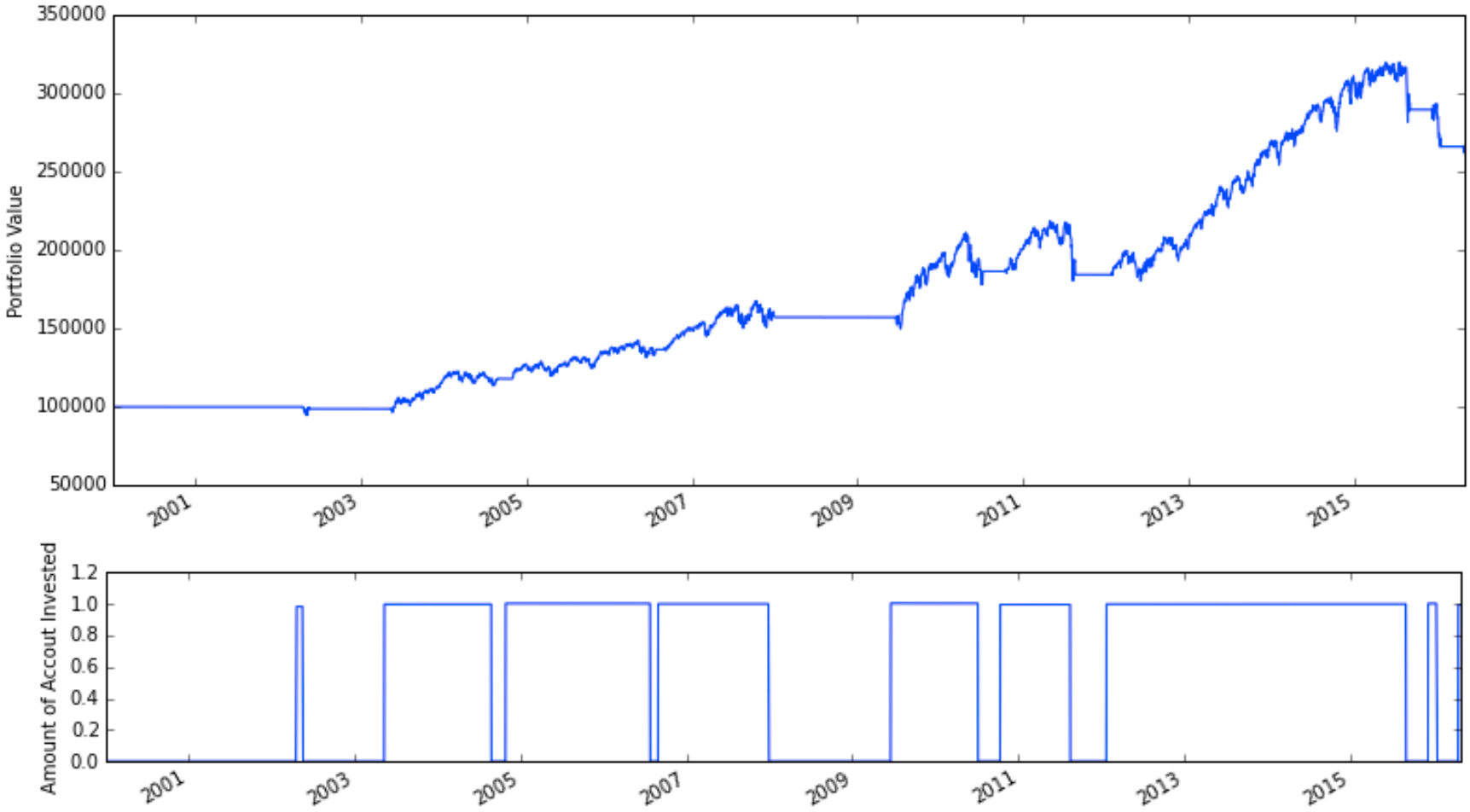
- Simple Example
  - ipython notebook, running locally
  - “Hello World” algorithm: 50-day MA/200-day MA crossover

- Machine Learning Example
  - Quantopian IDE + *pyfolio* in Quantopian’s hosted ipython notebook server
  - Inspired by algorithm shared in the Quantopian community forum
    - https://www.quantopian.com/posts/machine-learning-support-vector-regression

- If we have time...
  - Visualize sensitivity of a strategy to variation in input parameter values
    - Pair trading example using Gold and Oil ETF’s
      - https://www.quantopian.com/posts/sensitivity-analysis-aka-parameter-optimization-of-pair-trade-input-parameters
  - Zipline + TensorFlow
    - Dr. Erk Subasi, QuantCon 2016 Talk:
      - "Honey, I Deep-Shrunk the Sample Covariance Matrix!”
    - https://github.com/erksubasi/AutoencoderCovShrinkage/blob/master/QuantCon2016.ipynb

Zipline: Simple Example

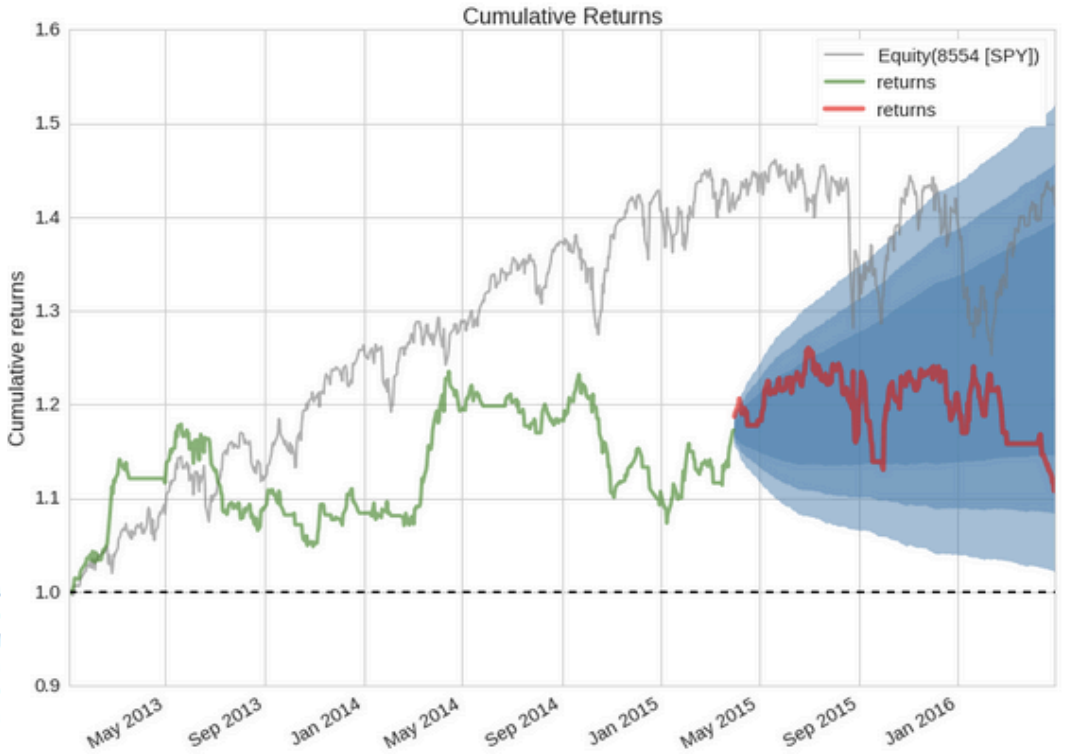
- From the Zipline Tutorial: http://www.zipline.io/beginner-tutorial.html#ipython-notebook
- The Hello World of trading strategies
  - Buy a stock when its 50-day moving average crosses above its 200-day moving average
  - Sell the stock when its 50-day MA falls back below its 200-day MA



Zipline: Machine Learning Example

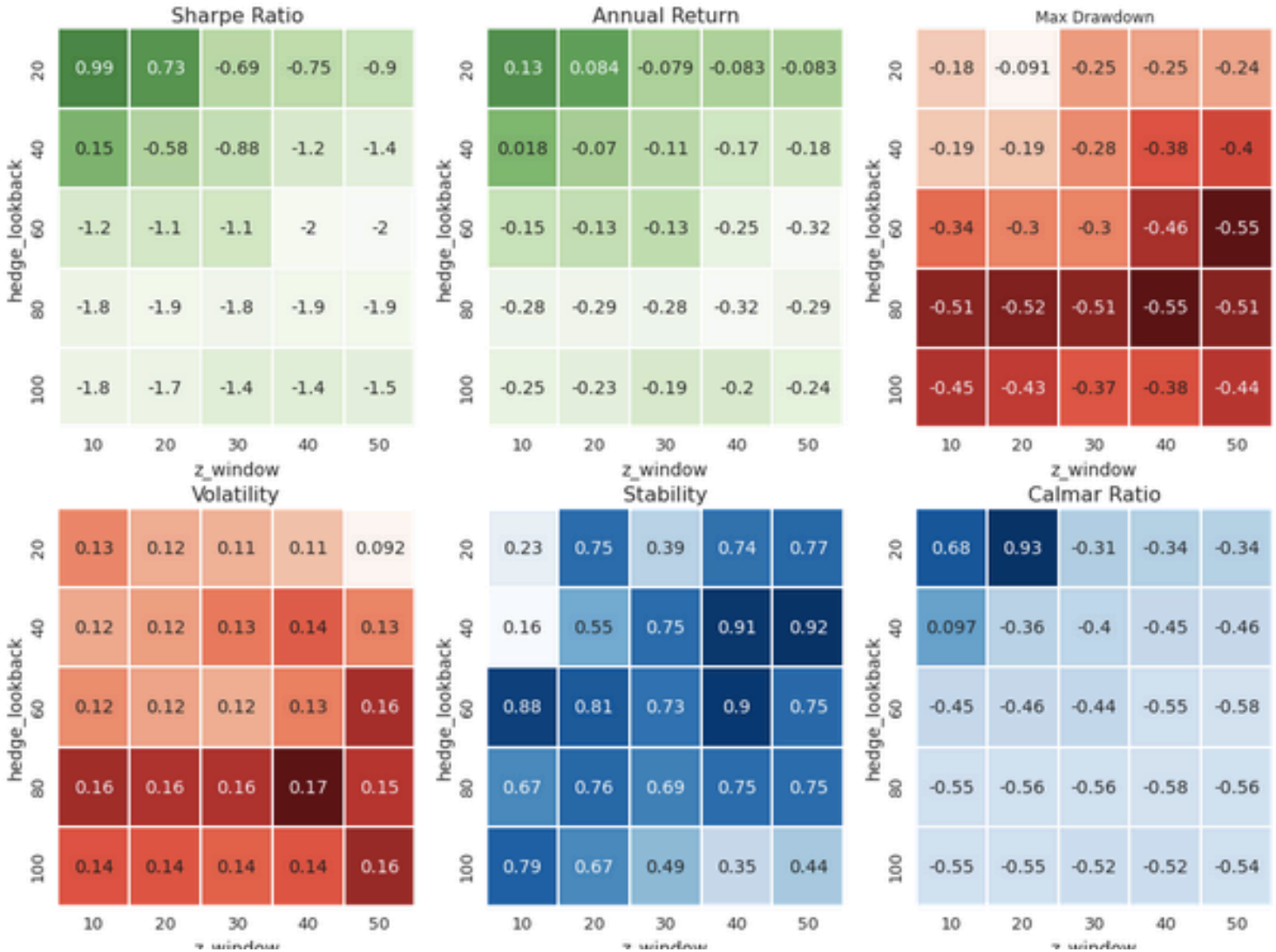
- Inspired by algorithm shared in the Quantopian community forum
  - https://www.quantopian.com/posts/machine-learning-support-vector-regression
- Train SVM on 5 simple price/volume features (open/high/low/close/volume)
  - Train using trailing 21-day (1-month) window, and predict whether the stock will be up or down the next day
    - Go Long or Short based on the prediction
  - Risk Management: If trade loses more than 1%, exit the trade.
    - Since we're using SPY (the SP500 ETF) in this example, a 1% move is somewhat sizeable
- ***Pyfolio*** analysis: For example purposes, I set the out-of-sample date to be right after the forum post was made

```
Entire data start date: 2013-01-03
Entire data end date: 2016-04-29


Out-of-Sample Months: 12
Backtest Months: 26
                    Backtest  Out_of_Sample  All_History
annual_return           0.07          -0.06         0.03
annual_volatility       0.09           0.13         0.11
sharpe_ratio            0.82          -0.43         0.35
calmar_ratio            0.56          -0.51         0.24
stability               0.21           0.20         0.41
max_drawdown           -0.13          -0.12        -0.13
omega_ratio             1.16           0.92         1.07
sortino_ratio           1.25          -0.65         0.52
skewness                0.41           0.72         0.57
kurtosis                1.99           5.47         4.87
information_ratio      -0.04          -0.02        -0.03
alpha                   0.08          -0.05         0.05
beta                   -0.05          -0.16        -0.10

Worst Drawdown Periods
   net drawdown in %  peak date valley date recovery date duration
0              13.11 2014-04-15  2015-01-08    2015-06-29      315
1              12.07 2015-07-01  2016-04-28           NaT      NaN
2              11.07 2013-05-21  2013-10-30    2014-03-28      224
3               2.21 2013-03-06  2013-05-01    2013-05-07       45
4               1.45 2014-04-07  2014-04-09    2014-04-10        4
```



Model Sensitivity to Input Parameter Values

- Pair trading example using Gold and Oil ETF's
  https://www.quantopian.com/posts/sensitivity-analysis-aka-parameter-optimization-of-pair-trade-input-parameters



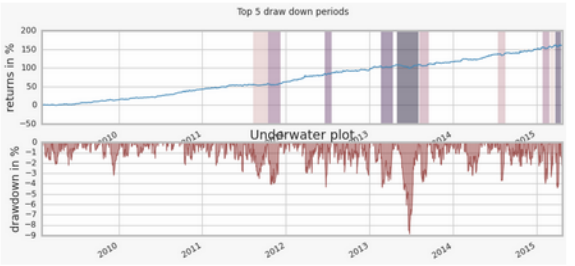
More details about ***pyfolio***

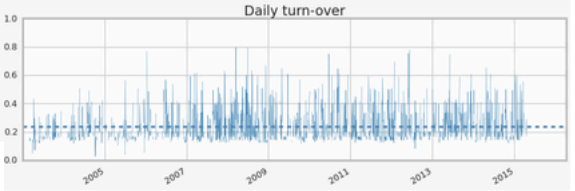
“Tearsheets”

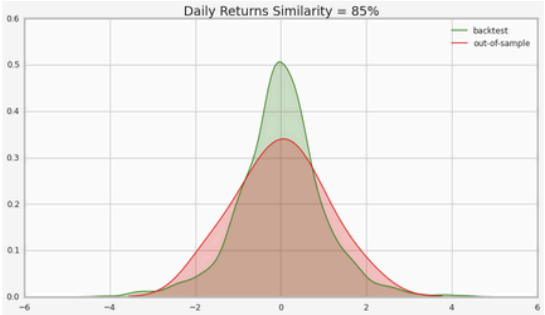
- Collection of tables and plots.

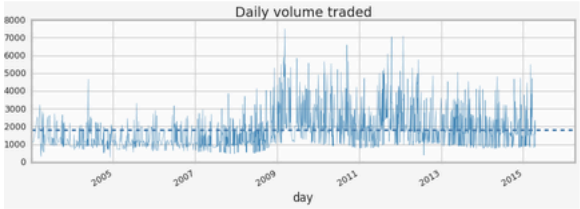
Visualizations

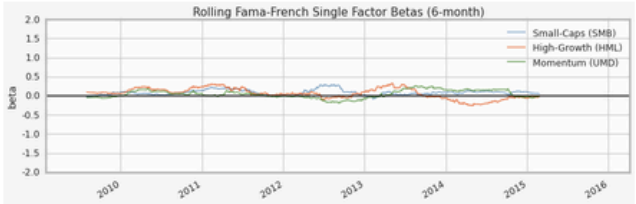
- Daily returns of a stock, or trading strategy
- Positions
- Transactions
- Periods of market stress
- *Bayesian risk analyses

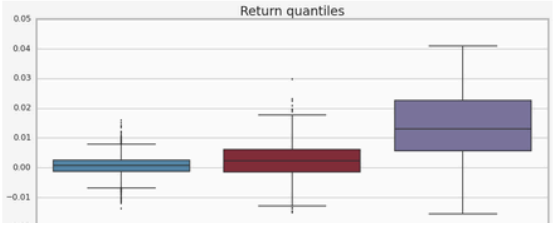


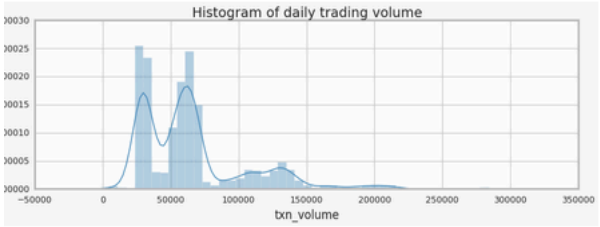




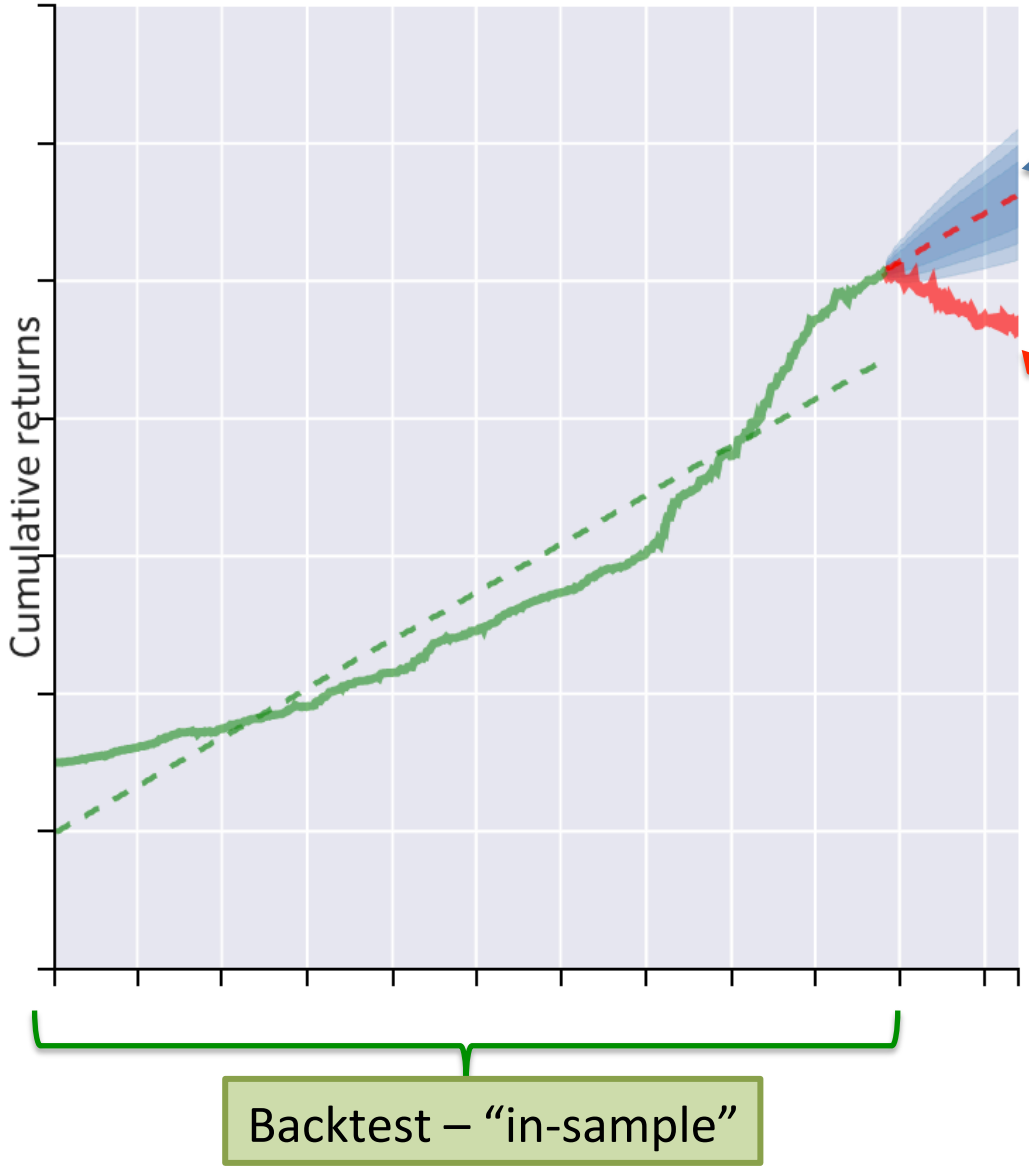








Backtest vs. Out-of-Sample Analysis



What is the Cone?

"*Cone*":
Projected expectations and "Risk bands" based on the backtest in-sample performance of the strategy

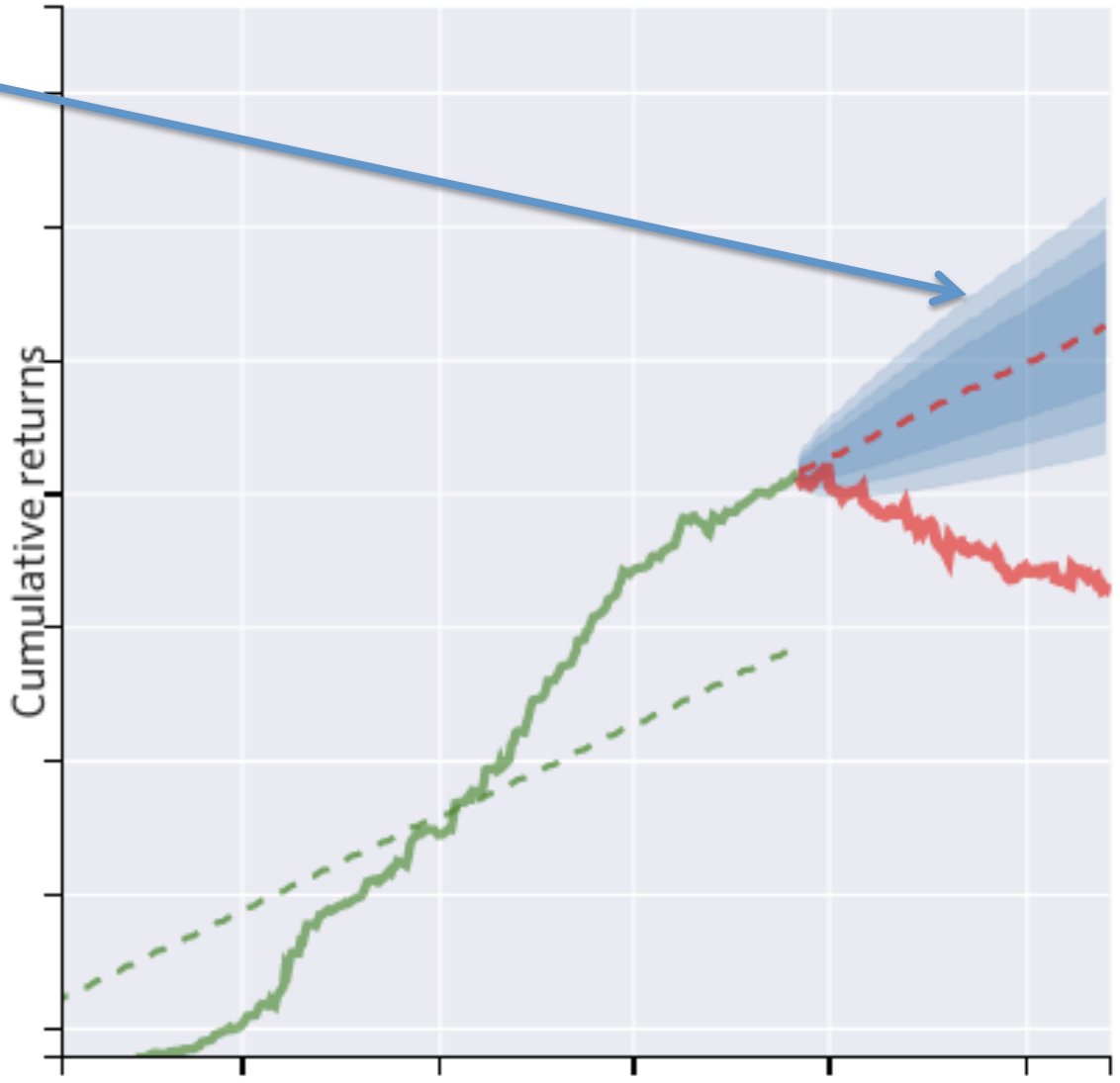
Green = Backtest, in-sample

Red = Live Trading, out-of-sample

Blue cone = volatility bands

- 1.0, 1.5, 2.0 stdevs

Consider exiting trading strategy if it starts trading outside of the -2.0 stdev region of the cone



Summary

- ***Zipline*** backtester can be used standalone or in the Quantopian IDE
  - Support for testing and trading futures contracts is coming

- ***pyfolio*** bundles various useful portfolio analyses tools and includes Bayesian modeling functionality beyond what was presented today
  - Can be used with Zipline/Quantopian developed strategies or simply on a CSV file of daily returns

- Quantopian's Jess Stauth, PhD. "Using pyfolio" webinar: https://www.youtube.com/watch?v=-VmZAlBWUko

- ***pyfolio*** is still young -- please contribute: https://github.com/quantopian/pyfolio/labels/help%20wanted

Thank you.
Questions?

justin@quantopian.com

Disclaimer

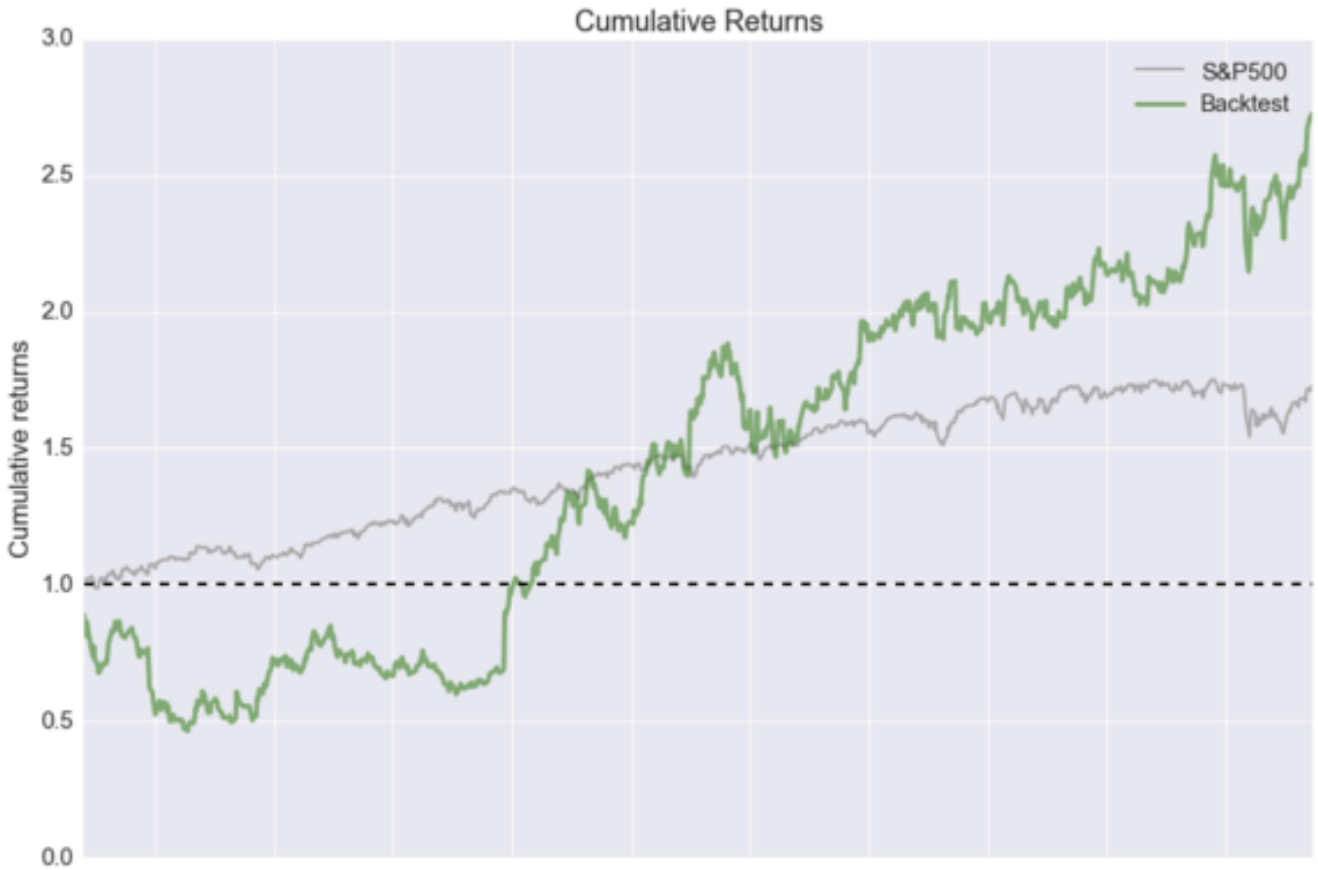
This presentation is for informational purposes only and does not constitute an offer to sell, a solicitation to buy, or a recommendation for any security; nor does it constitute an offer to provide investment advisory or other services by Quantopian, Inc. ("Quantopian"). Nothing contained herein constitutes 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, Inc. 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, Inc. 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.

Appendix

Pyfolio Tearsheet Components

In [6]:
```
pf.create_returns_tear_sheet(stock_rets)
```

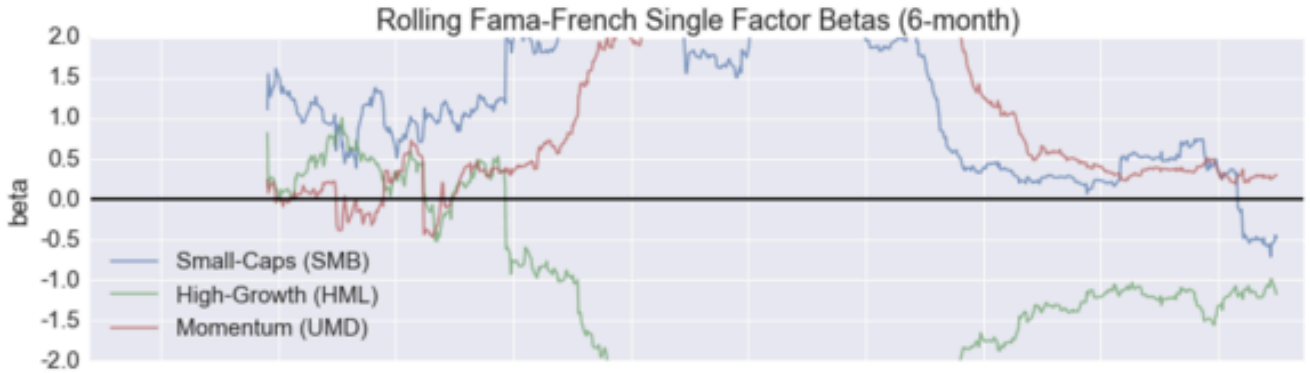
```
Entire data start date: 2012-05-21
Entire data end date: 2015-10-28


Backtest Months: 41
                     Backtest
annual_return            0.38
annual_volatility        0.44
sharpe_ratio             0.88
calmar_ratio             0.80
stability                0.88
max_drawdown            -0.48
omega_ratio              1.18
sortino_ratio            1.42
skewness                 1.74
kurtosis                19.32
alpha                    0.22
beta                     1.01

Worst Drawdown Periods
   net drawdown in %  peak date valley date recovery date duration
1              47.90 2012-05-21  2012-09-04    2013-07-25      309
2              22.06 2014-03-10  2014-04-28    2014-07-24       99
3              17.34 2013-10-18  2013-11-25    2013-12-17       43
0              16.57 2015-07-21  2015-08-24    2015-10-19       65
4               9.20 2015-03-24  2015-05-12    2015-06-23       66


2-sigma returns daily    -0.053
2-sigma returns weekly   -0.108
```


Cumulative Returns
S&P500
Backtest
Cumulative returns
3.0
2.5
2.0
1.5
1.0
0.5
0.0


Cumulative returns volatility matched to benchmark.
S&P500
Backtest
Cumulative returns
1.8
1.6
1.4
1.2
1.0
0.8

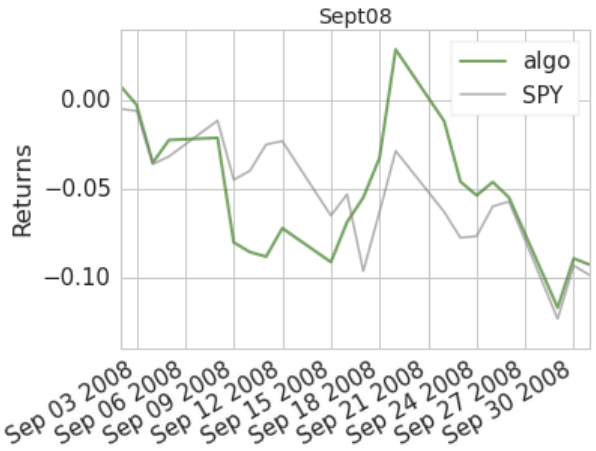

Rolling Portfolio Beta to S&P 500
6-mo
12-mo
Beta
2.0
1.0
0.0
-1.0
-2.0

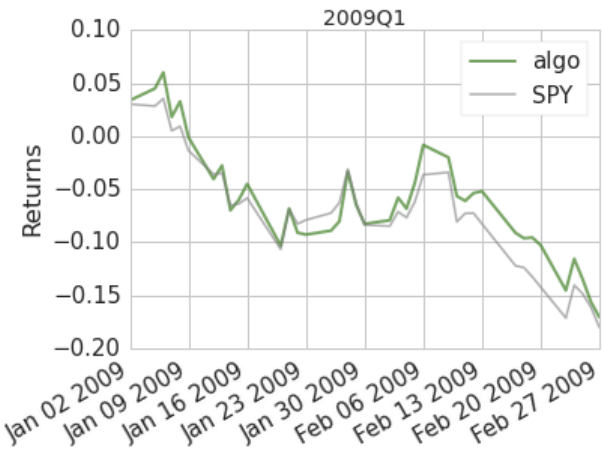

Rolling Sharpe ratio (6-month)
Sharpe
Average
Sharpe ratio
6.0
5.0
4.0
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0

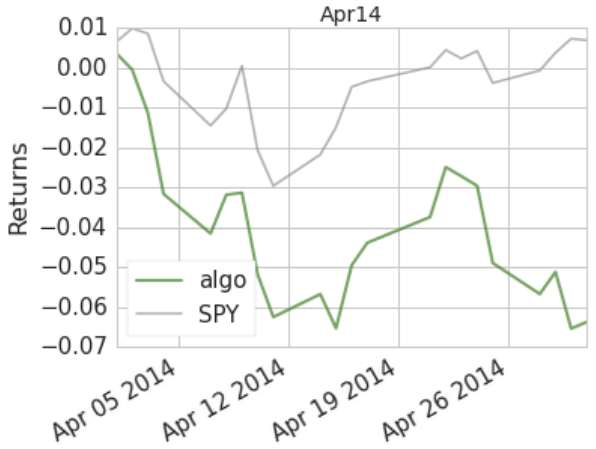

Rolling Fama-French Single Factor Betas (6-month)
beta
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
-2.0
Small-Caps (SMB)
High-Growth (HML)
Momentum (UMD)

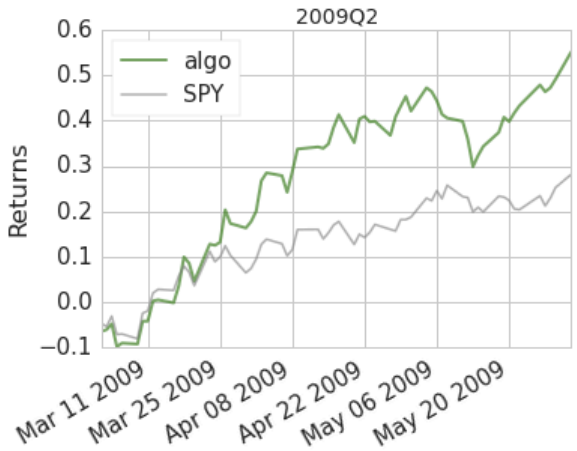
Performance during Market Stress Events

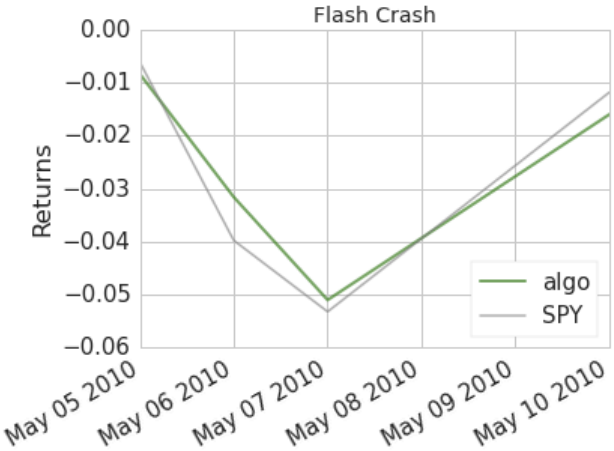
Pyfolio contains 15-20 pre-defined market stress periods so you can easily see how well your strategy performs during crisis events

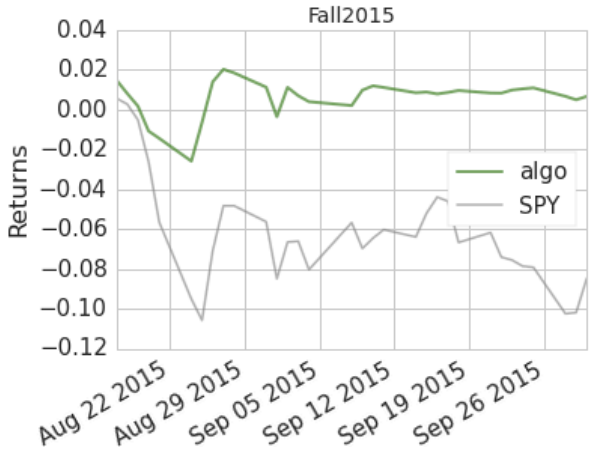


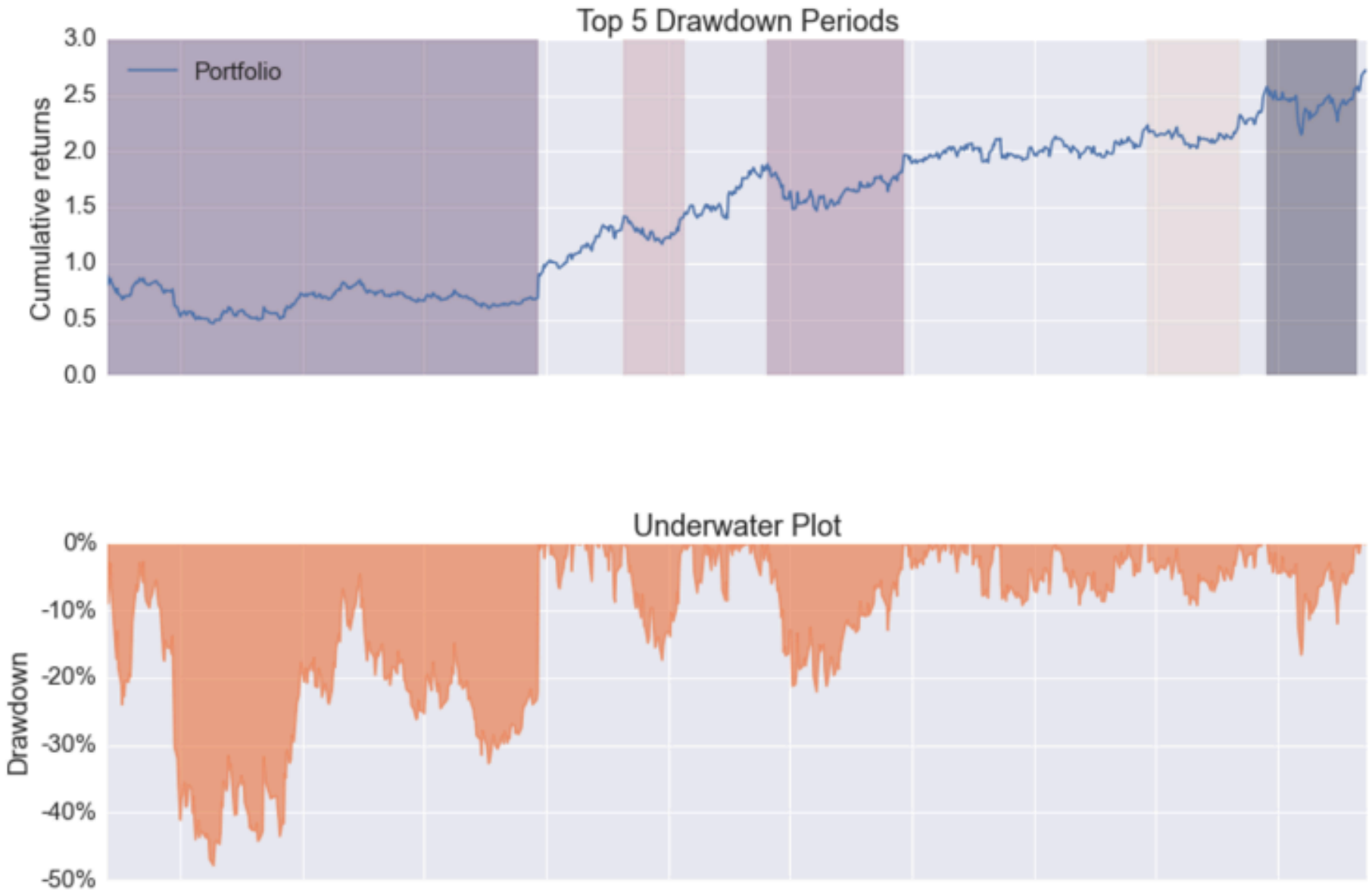


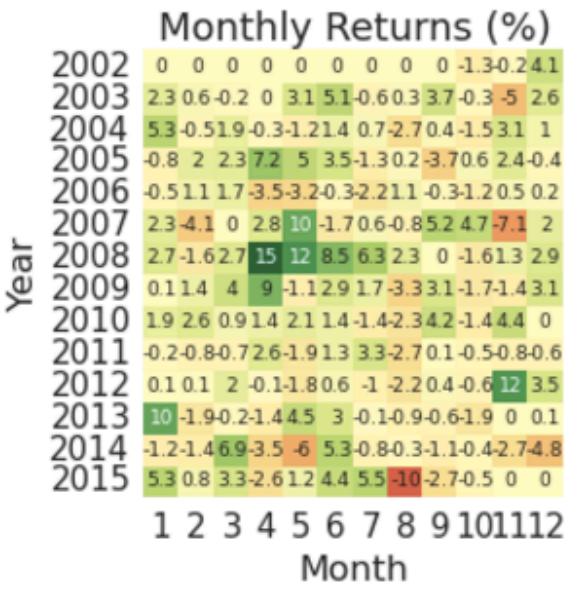




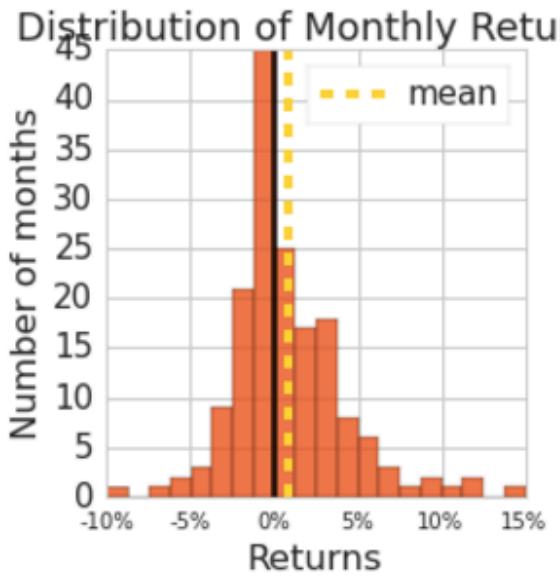


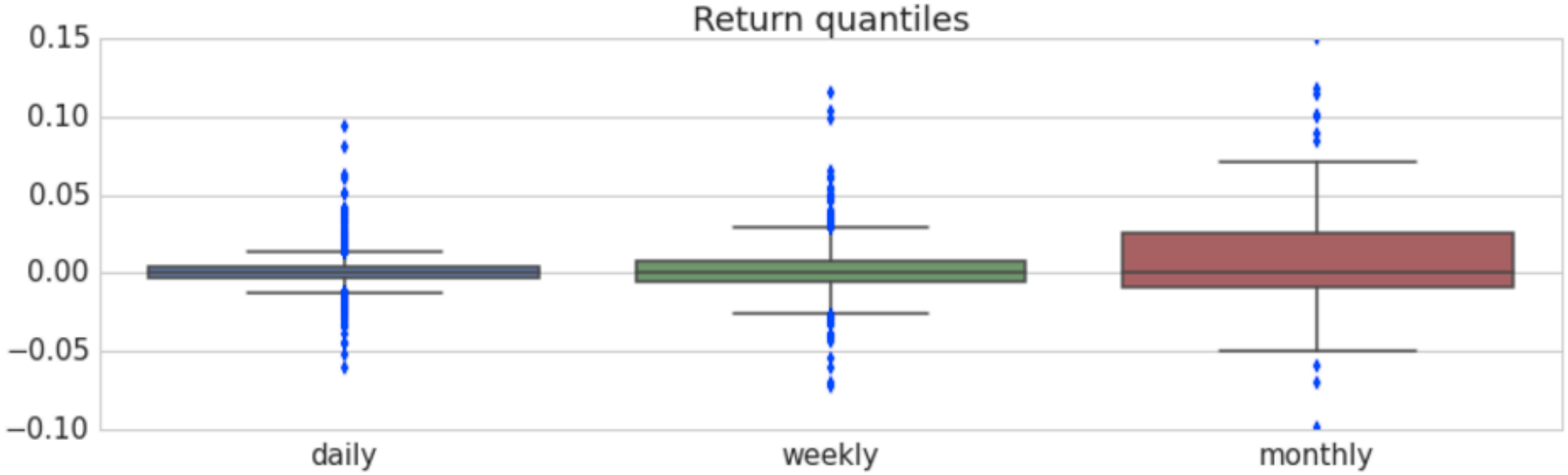



Top 5 Drawdown Periods
Portfolio
Cumulative returns
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Underwater Plot
Drawdown
0%
-10%
-20%
-30%
-40%
-50%

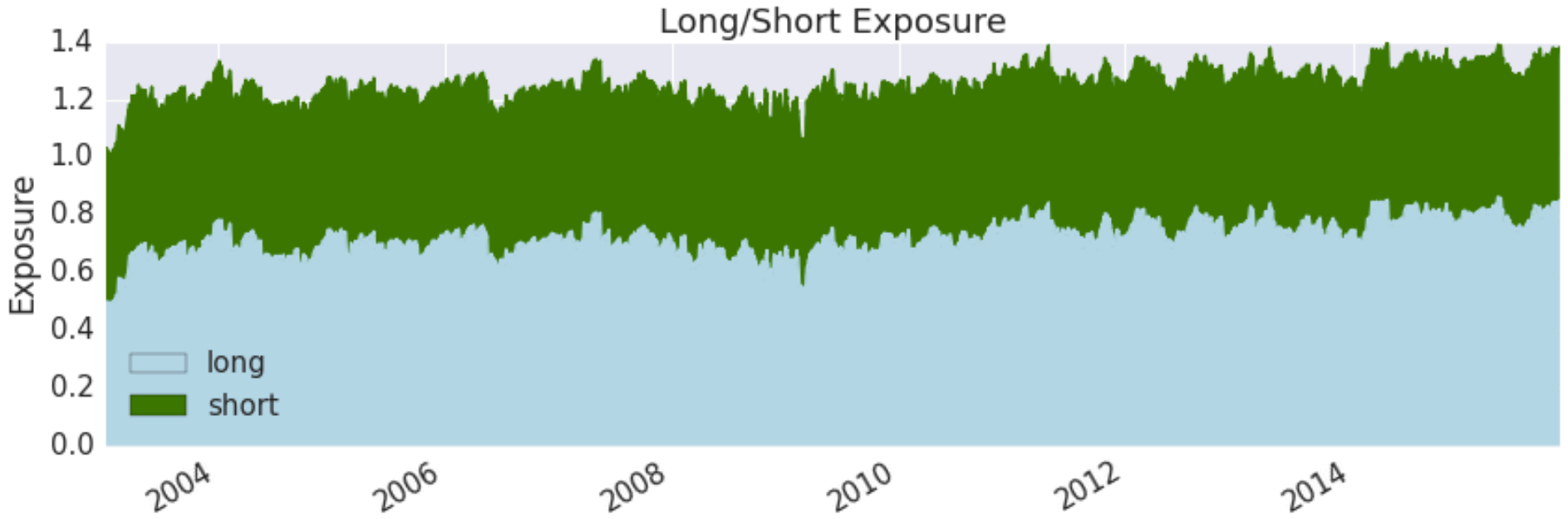

Monthly Returns (%)
Year
Month
2002 0 0 0 0 0 0 0 0 0 -1.3 -0.2 4.1
2003 2.3 0.6 -0.2 0 3.1 5.1 -0.6 0.3 3.7 -0.3 -5 2.6
2004 5.3 -0.5 1.9 -0.3 -1.2 1.4 0.7 -2.7 0.4 -1.5 3.1 1
2005 -0.8 2 2.3 7.2 5 3.5 -1.3 0.2 -3.7 0.6 2.4 -0.4
2006 -0.5 1.1 1.7 -3.5 -3.2 -0.3 -2.2 1.1 -0.3 -1.2 0.5 0.2
2007 2.3 -4.1 0 2.8 10 -1.7 0.6 -0.8 5.2 4.7 -7.1 2
2008 2.7 -1.6 2.7 15 12 8.5 6.3 2.3 0 -1.6 1.3 2.9
2009 0.1 1.4 4 9 -1.1 2.9 1.7 -3.3 3.1 -1.7 -1.4 3.1
2010 1.9 2.6 0.9 1.4 2.1 1.4 -1.4 -2.3 4.2 -1.4 4.4 0
2011 -0.2 -0.8 -0.7 2.6 -1.9 1.3 3.3 -2.7 0.1 -0.5 -0.8 -0.6
2012 0.1 0.1 2 -0.1 -1.8 0.6 -1 -2.2 0.4 -0.6 12 3.5
2013 10 -1.9 -0.2 -1.4 4.5 3 -0.1 -0.9 -0.6 -1.9 0 0.1
2014 -1.2 -1.4 6.9 -3.5 -6 5.3 -0.8 0.3 -1.1 -0.4 -2.7 -4.8
2015 5.3 0.8 3.3 -2.6 1.2 4.4 5.5 -10 -2.7 -0.5 0 0
1 2 3 4 5 6 7 8 9 10 11 12


Annual Returns
Year
Returns
mean
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
-20% -10% 0% 10% 20% 30% 40% 50% 60% 70%

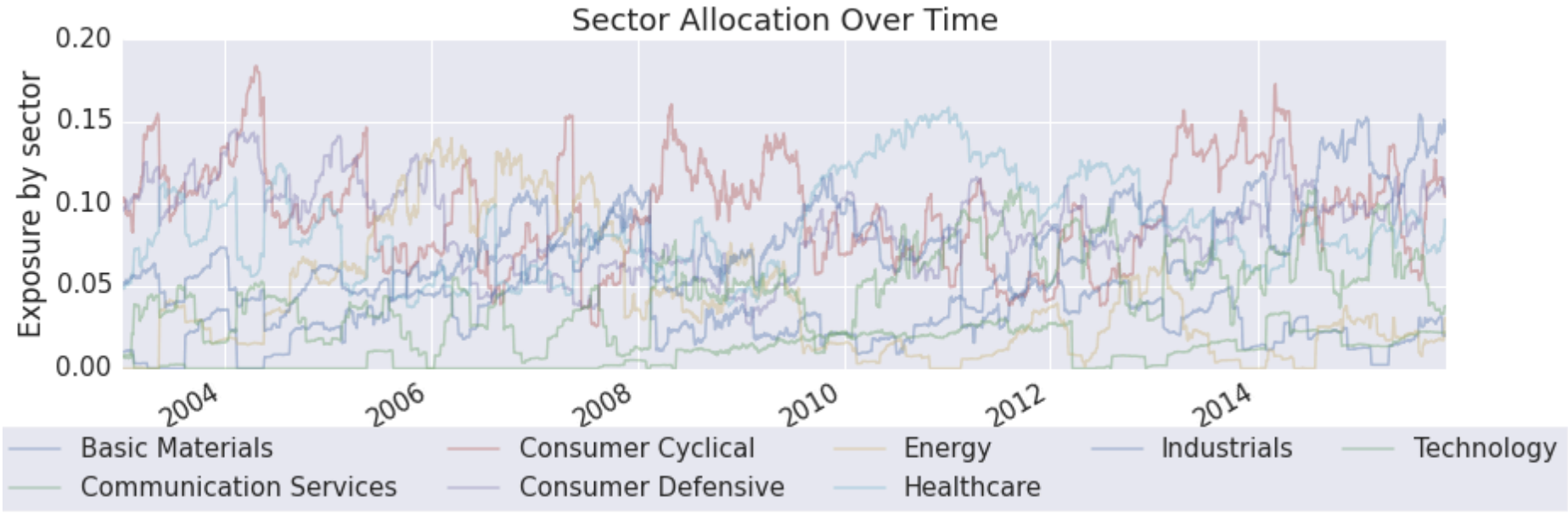

Distribution of Monthly Returns
Number of months
Returns
mean
0 5 10 15 20 25 30 35 40 45
-10% -5% 0% 5% 10% 15%

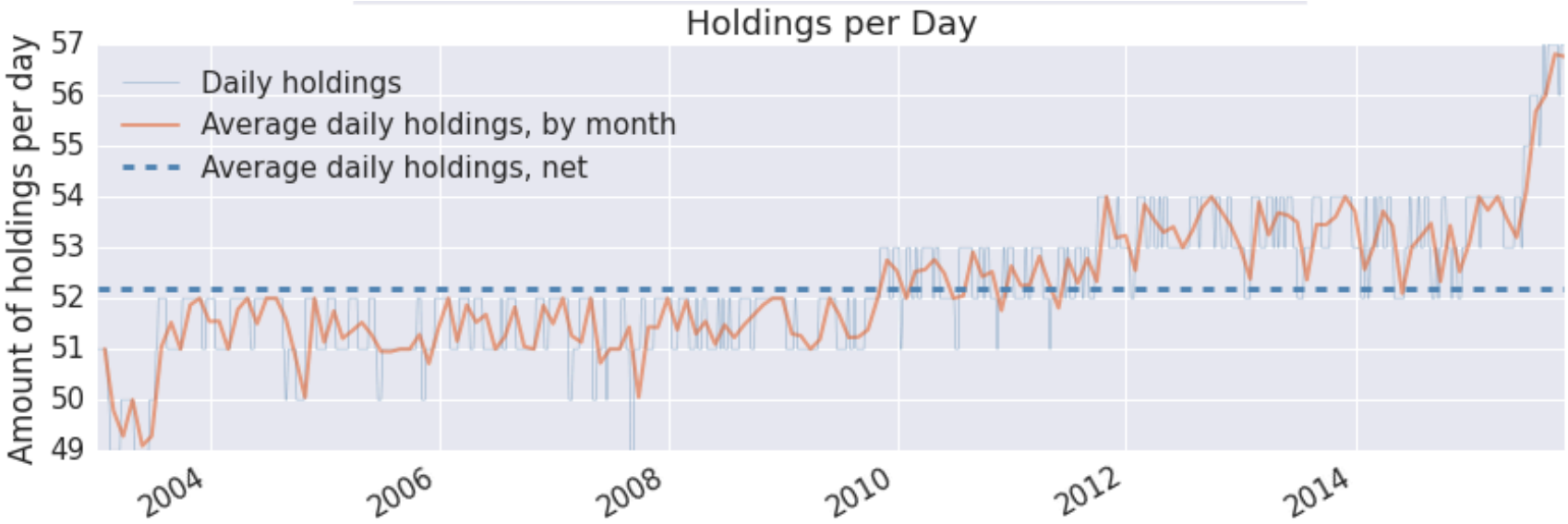

Return quantiles
0.15
0.10
0.05
0.00
−0.05
−0.10
daily
weekly
monthly

Long/Short Exposure over Time



Sector Exposure over Time



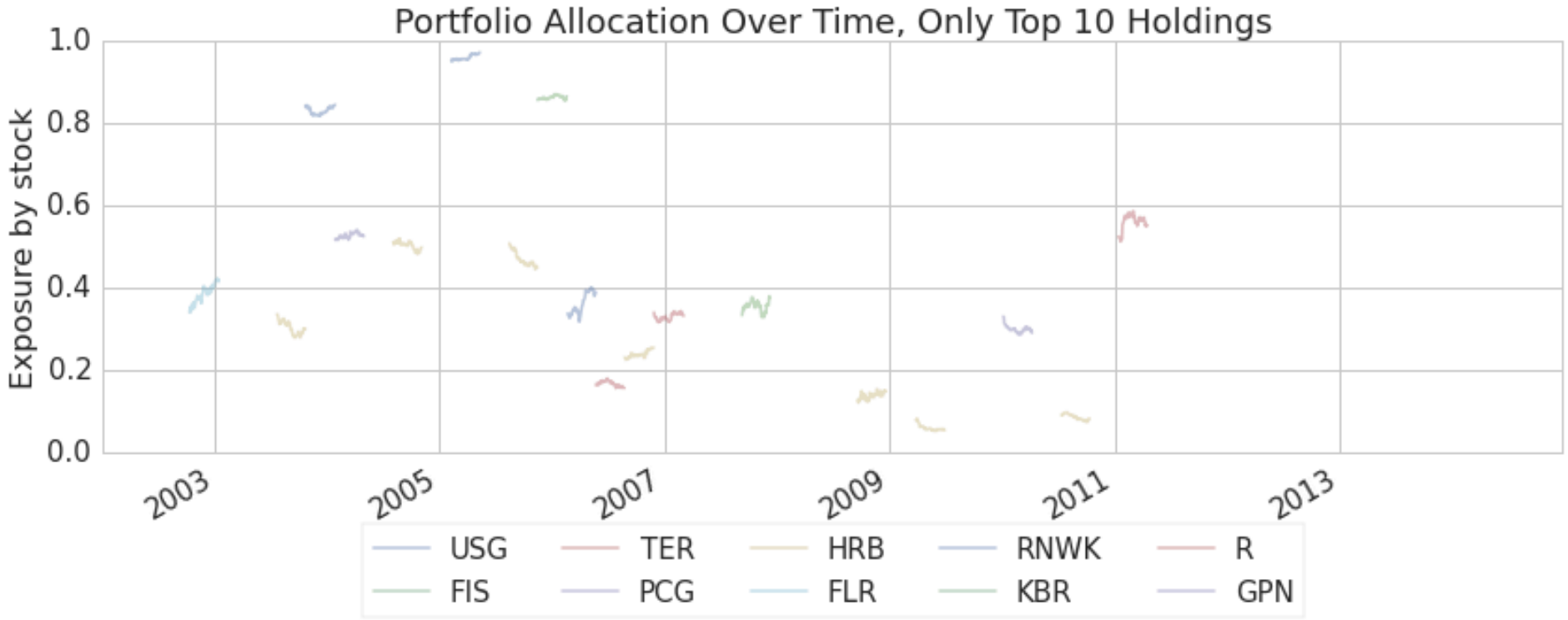


Single Stock Concentration

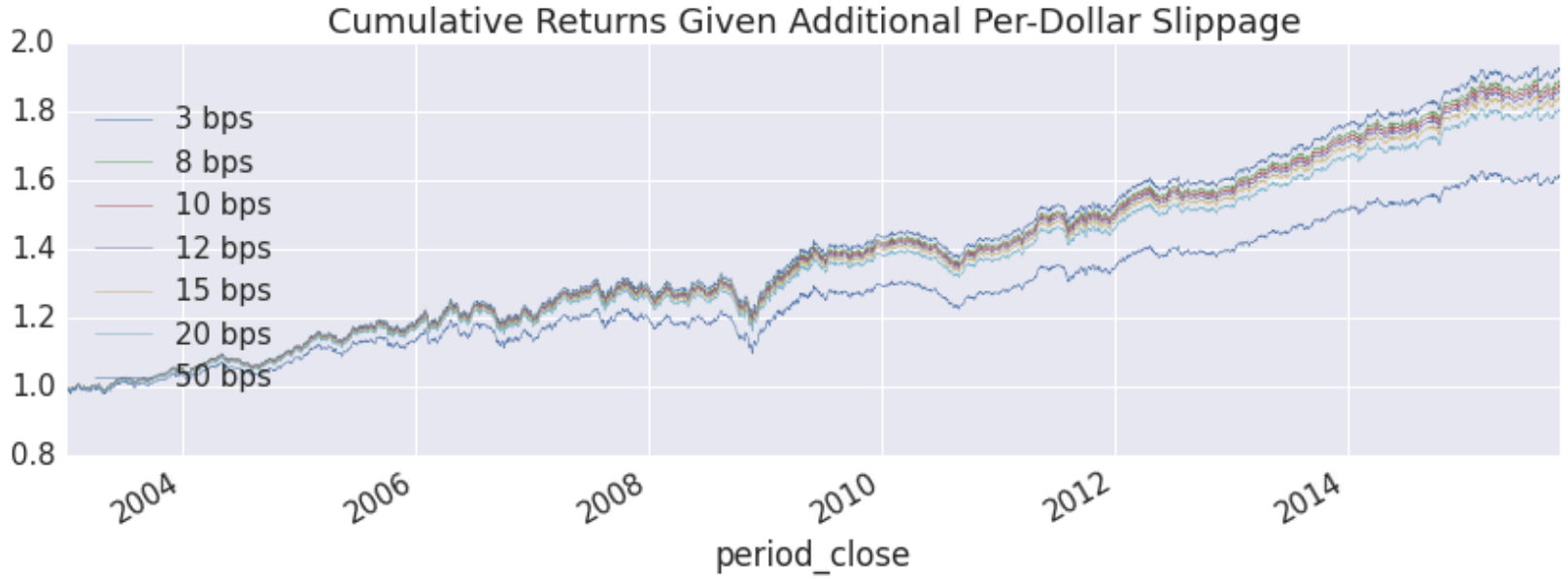
```
Top 10 long positions of all time (and max%)
[u'USG' u'FIS' u'TER' u'PCG' u'HRB' u'FLR' u'RNWK' u'KBR' u'R' u'GPN']
[ 0.972  0.87   0.586  0.54   0.52   0.423  0.401  0.379  0.343  0.33 ]
```

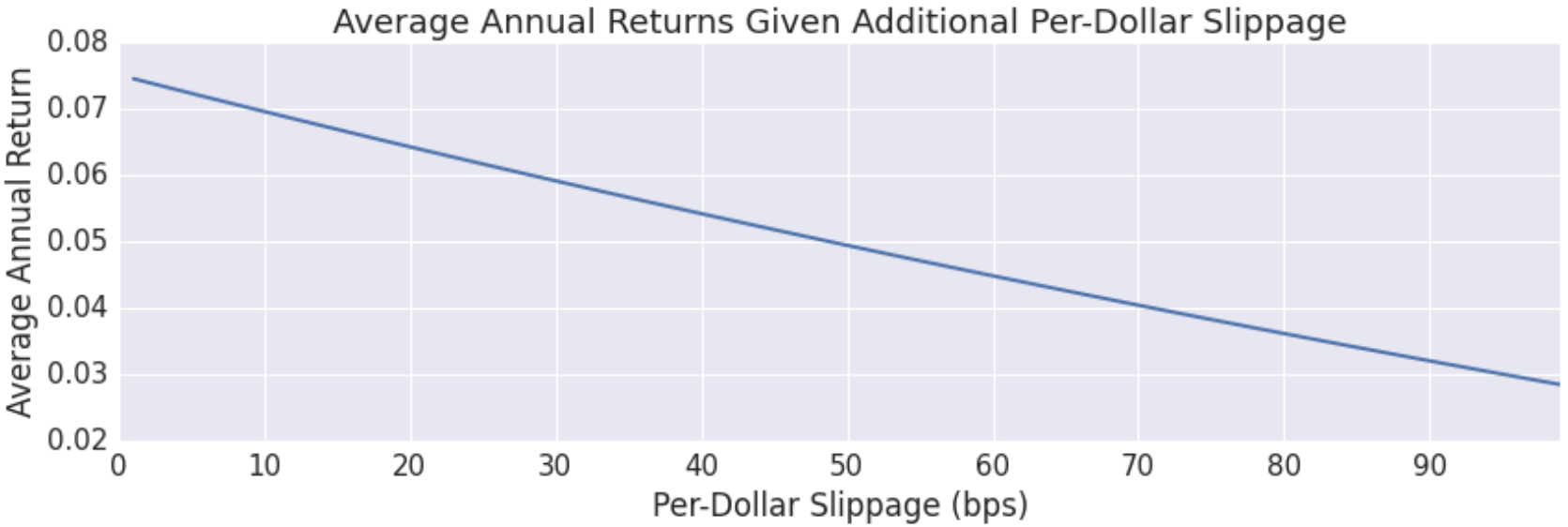
Red Flag, at 1 point, USG was 97% of the portfolio!

```
Top 10 short positions of all time (and max%)
[]
[]
```



Slippage and Transaction Cost Sensitivity

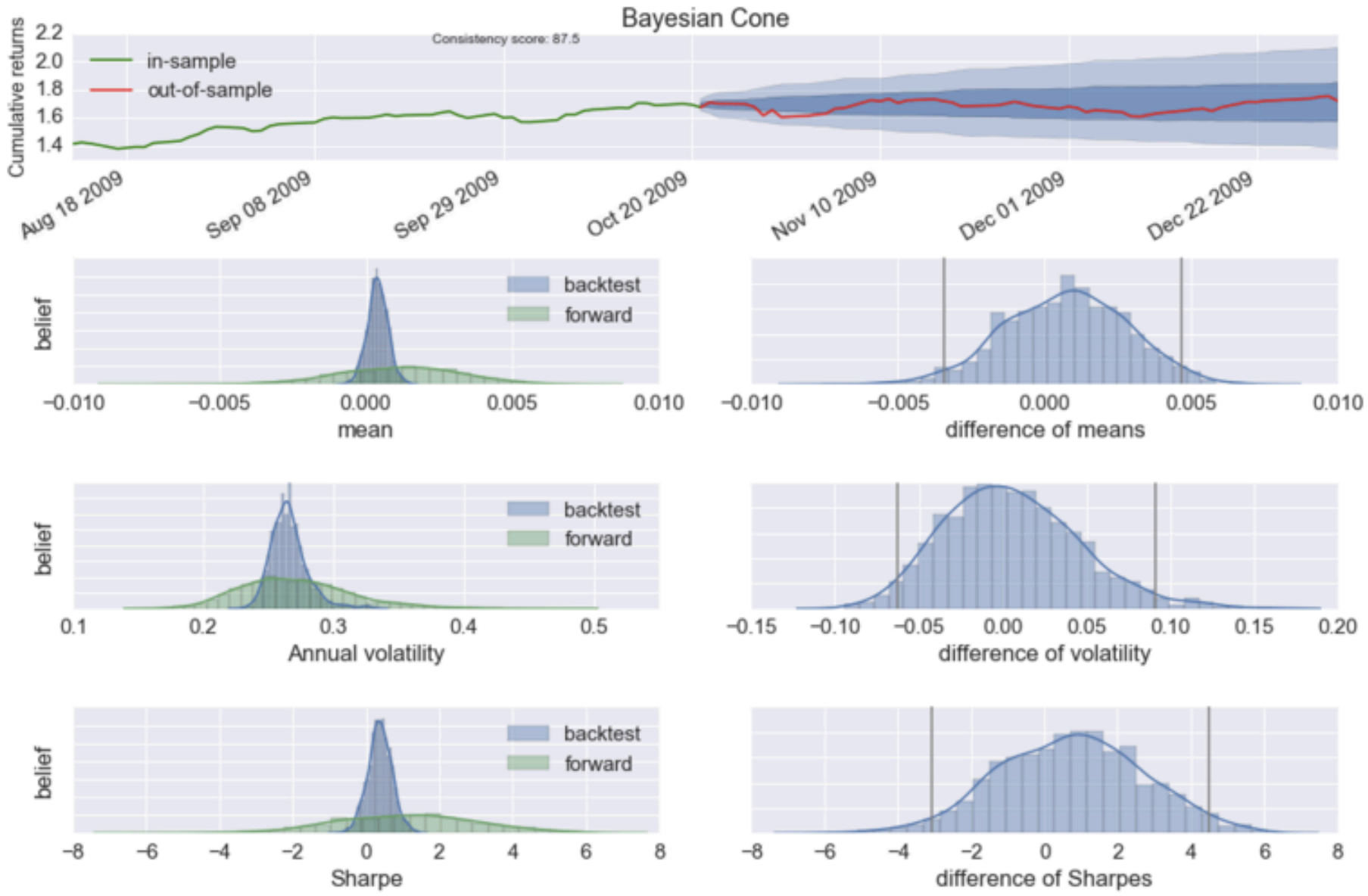




Bayesian analysis in pyfolio

- Sneak-peek into ongoing research.
- Primary focus is comparing backtest (in-sample) and forward-test (out-of-sample; OOS).
- Sophisticated statistical modeling takes *uncertainty* into account.
- Uses T-distribution to model returns (instead of normal).
  - Addresses 'fat-tail' nature of financial returns
- Relies on PyMC3.
  - Python module for Bayesian statistical modeling and model fitting which focuses on advanced Markov chain Monte Carlo fitting algorithms.

Modeling Trading Strategy Uncertainty with Bayesian Analysis



More Info on Bayesian Analysis

Accompanying blog post:
http://blog.quantopian.com/bayesian-cone/

Bayesian Methods for Hackers:
http://camdavidsonpilon.github.io/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/

Our Data Scientist's blog (Thomas Wiecki, PhD)

- twiecki.github.io
- Active developer of PyMC3:
  http://pymc-devs.github.io/pymc3

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, Inc. 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, Inc. 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.