



Quantopian

Data Business Meetup
Feb 20, 2013

Quant Finance



Quant Researcher < -- > Quant Developer



Even the simple things are hard.

IAC/InteractiveCorp (IACI)







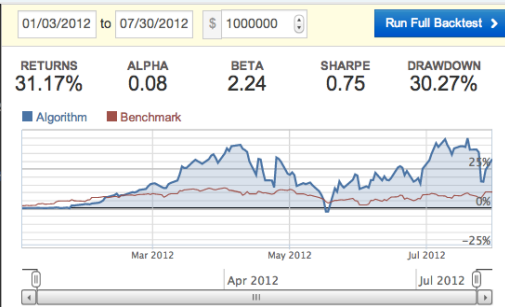
Quantopian is a **cloud trading platform** dedicated to quants.

Quantopian is the first browser-only algorithmic trading platform.

No downloads, no install, no hassle.

Create and backtest algorithms in less than a minute, compared to months of work with existing systems.

```
1 # Put any initialization logic here. The context object will
2 # the other methods in your algorithm.
3 def initialize(context):
4     pass
5
6 # Will be called on every trade event for the securities you
7 def handle_data(context, data):
8     # Implement your algorithm logic here.
9
10    # data[sid(X)] holds the trade event data for that securi
11    # data.portfolio holds the current portfolio state.
12
13    # Place orders with the order(SID, amount) method.
14
15    # TODO: implement your own logic here.
16    order(sid(24), 50)
```



A community of quants that expands the current market. Members range from curious hobbyists and aspiring students to Wall Street professionals.

Quantopian Community

Newest Recent Activity All Threads

New Post

Search



Ernie Chan's "Gold vs. gold-miners" stat arb

John Fawcett, Aug 24, 2012

2735 views 13 replies

feature-Event Study Functionality

Michael Antoniou, 18 hours ago

14 views 0 replies

Different kinds of data sources?

Jonathan Kamens, Nov 2, 2012

155 views 5 replies

Quantopian?

TOP ALGORITHMS

Buy low Sell high 14
gary chan

Ernie Chan's "Gold vs. gold-miners" stat arb 6
John Fawcett

Moving Average Crossover 5

You require internet users to code?

Are you guys crazy?

You need a point and click interface!

40M STEM pros world-wide want to get their hands on the data. That means code.



Real-time code validation, testing, and simulation.

Quantopian

Cloned from "Bull or Bear, not too complicated"

Algorithm Backtest

Save Build Algorithm Help

01/03/2012 to 02/15/2013 \$ 1000000 Daily Run Full Backtest

RETURNS	ALPHA	BETA	SHARPE	DRAWDOWN
32.53%	0.12	0.97	2.04	11.25%

Algorithm 32.53% Benchmark 19.36%

Build Errors Logs Runtime Errors More

```
2012-01-03 handle_data:32 INFO 1000 shares of Security(1637) bought
2012-01-03 handle_data:32 INFO 1000 shares of Security(27558) bought
2012-01-04 handle_data:32 INFO 1000 shares of Security(2190) bought
2012-01-05 handle_data:32 INFO 1000 shares of Security(35920) bought
2012-01-09 handle_data:32 INFO 1000 shares of Security(5729) bought
2012-01-13 handle_data:32 INFO 1000 shares of Security(35902) bought
2012-01-17 handle_data:32 INFO 1000 shares of Security(32146) bought
End of logs.
```

```
1 # Put any initialization logic here. The context object will be passed
2 import numpy as np
3
4 # the other methods in your algorithm.
5 def initialize(context):
6     context.stock=[sid(5729), sid(1637), sid(27558), sid(2190), sid(35920)]
7     context.length = len(context.stock)
8     context.basis = np.zeros(context.length)
9     context.bought = [False]*context.length
10
11
12     context.sell_limit = 1.01 # relative price limit to submit sell order
13
14 # Will be called on every trade event for the securities you specify.
15 def handle_data(context, data):
16     # create numpy array of current security prices
17     current_Prices = np.zeros(context.length)
18     current_Volumes = np.zeros(context.length)
19     last_Prices = np.zeros(context.length)
20     last_Volumes = np.zeros(context.length)
21
22     global last_Volumes #There is no open_volume parameter so last_volume
23
24     for i, stock in enumerate(context.stock):
25         last_Prices[i]=data[stock].open_price
26         current_Prices[i] = data[stock].price
27         current_Volumes[i] = data[stock].volume
28         if not context.bought[i]:
29             last_Volumes[i]=current_Volumes[i]
30             if current_Prices[i]>=context.sell_limit*last_Prices[i] and
31                 order(stock,1000)
32                 log.info("1000 shares of %s bought" %stock)
33                 context.bought[i]=True
```

Your backtester is open-sourced?
Are you guys crazy?

Zipline, a pythonic backtesting library

<http://zipline.io>

quantopian / zipline

[Pull Request](#)

[Unwatch](#)

[Unstar](#)

692

[Fork](#)

66

Code

Network

Pull Requests 4

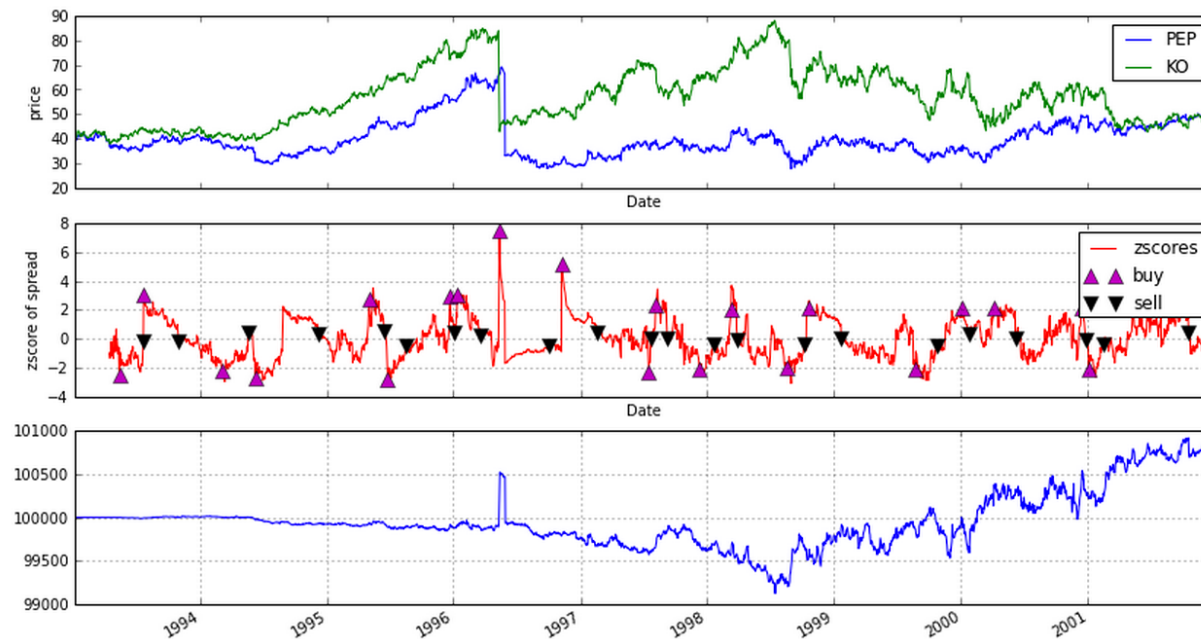
Issues 19

Wiki


Graphs

Settings

Zipline, a Pythonic Algorithmic Trading Library — [Read more](#)



Aspiring Quants need Mentorship

 Quantopian Community

[New Post](#)

Search



Indicator Library

Devon Kyle posted 3 hours ago

17 views 0 replies

TDSequential and other Tom DeMark indicators

Devon Kyle replied 3 hours ago

520 views 2 replies

Max Dama on Automated Trading (pdf)

Devon Kyle replied 3 hours ago

282 views 2 replies

How do I keep this from making my portfolio seriously overdrawn?

James Jack replied 5 hours ago

23 views 1 reply



Bull or Bear, not too complicated

james con replied 14 hours ago

540 views 5 replies

Coursera's Computational Investing Class Begins Feb 22

John Fawcett replied 16 hours ago

788 views 9 replies

Issue Displaying Backtest Results

Rock Fu replied 18 hours ago

52 views 6 replies

Upcoming Webinar: Feb 20

Want to learn how to use Quantopian? Have questions on how to make Quantopian work for you?

Our webinar on **Feb 20, 11am EST** will have a product overview and an extended Q&A session to answer all of your questions.

[Sign Up](#)

TOP SHARED ALGORITHMS

Momentum Trade 48
Zeke Newsom

Mebane Faber Relative Strength Strategy with MA Rule 26
John Chia

 Quantopian

Thanks!

fawce@quantopian.com