Democratized Trading Using Quantopian

ISB | 2018 | FinTech

Trading Algorithm

On Quantopian, a trading algorithm is a Python program that defines two special functions: `initialize()` and `handle_data()`. `initialize()` is called when the program is started, and `handle_data()` is called once per minute during simulation or live-trading in events that we'll refer to as 'bars'. The job of `initialize()` is to perform any one-time startup logic. The job of `handle_data()` is to decide what orders, if any, should be placed each minute.

The following is an example of an algorithm that allocates 100% of its portfolio in AAPL:

```
def initialize(context):
    # Reference to AAPL
    context.aapl = sid(24)

def handle_data(context, data):
    # Position 100% of our portfolio to be long in AAPL
    order_target_percent(context.aapl, 1.00)
```

Clone

To run this example algorithm, create a copy by clicking the "Clone" button in the upper right hand corner. Run a backtest by clicking "Build Algorithm" (top left) or "Run Full Backtest" (top right).

https://www.quantopian.com/tutorials/getting-started

Modeling Intro

- Finance: https://www.quantopian.com/research/notebooks/Tutorials%20and%20Documentation/2.%20Getting%20Started%3A%20Financial%20Modeling.ipynb
- Handing data: https://www.quantopian.com/research/notebooks/Tutorials%20and%20Documentation/3.%20Tutorial%20-%20Manipulating%20Data%20with%20Pandas.ipynb
- Backtesting: https://www.quantopian.com/research/notebooks/Tutorials%20and%20Documentation/6.%20Tutorial%20-%20Analyzing%20Backtest%20Results.ipynb

A Laundry List of Risk Measures for complex trading and complex FIs

Complex structures are harder to manage because of:

- Multiple strategies.
- Long-short positions.
- Dynamic strategies.
- Widespread use of derivatives.
- Multilayered portfolios and delegated/sub-advisory portfolios.

Risk Measures for Complex Funds

(1) *Absolute excess return (AER)*: $\mu - r_f$.
   (a) As complex funds are often defined as absolute return vehicles, investors in complex funds would do well to rank them by their AERs.
   (b) Many complex funds are beta neutral, and, therefore, excess return is the correct metric for the assessment of such portfolios.
   (c) Investors in a pool of complex funds should also examine AER correlations across funds to gain insights into the extent of diversification afforded by each asset class. When choosing a pool of complex funds, the covariance matrix of AERs of the funds will provide insights into the relatedness of subportfolio alphas.

(2) *Total risk* σ. As complex funds take on especially high levels of idiosyncratic risk, total risk is one of the measures of risk that needs to be examined carefully. Total risk is the standard deviation of returns of the fund and is easy to compute from the time series of returns of the fund. Combining this risk measure with AER provides the Sharpe ratio.

Higher-Order Moments

(3) *Higher-order moments of returns*

(a) Complex funds invest in deal structures (long/shorts, etc.) rather than just asset classes, and hence, their portfolios may not be well diversified.

(b) Complex funds undertake dynamic strategies, and along with the lack of diversification, they imply that the return distribution of the fund will deviate substantially from Gaussian.

(c) Given this, higher-order moments such as skewness and kurtosis become important, and should be computed and examined. In many instances, deal structures will lead to very low standard deviation, and the risk is hidden in the skewness and kurtosis of returns. Investors are naturally averse to negative skewness and excess kurtosis.

(d) An example of a deal in which higher-order moments are important is one where there is convergence trade implemented with a long/short position. Most days, this position will result in no gain or loss, and on the occasional day the leveraged position will result in a huge gain or loss. The standard deviation of such a series of returns is very small, often close to zero, but the kurtosis of these returns will be very large, given the size of the outlier returns compared to the usual daily move.

Gaming Risk Measures

(e) As another example, consider a portfolio comprising the S&P500 index. Such a portfolio has returns that may be closer to Gaussian than most others. Take such a portfolio and short 20% of its face value in call options on the index. This results in premiums being collected, pushing the mean return higher, and also reduces the effective stock position, lowering the standard deviation, and results in a higher Sharpe ratio. Is this a free lunch? No, the risk is buried in the fact that the short call position has made the skewness of returns of the net portfolio much more negative than before. In this way funds can game the Sharpe ratio by hiding risk in the higher moments. Therefore, it is necessary, especially with levered portfolios arising from long/short trades, or from positions in derivatives, to examine carefully all the higher-order moments as well.

Semi-variance

(4) *Semi-variance*. (a) Semi-variance is the variance of returns on one side of the return distribution. Usually semi-variance uses only those observations that fall below the mean return (though the median may also be used).

(b) Semi-variance of returns is a measure of downside risk. It offers additional information over and above what may be seen in the variance itself. It also has a close relationship to skewness and is therefore one way of examining higher-order moment effects.

Turnover

(5) *Turnover*. (a) There is documented evidence that mutual funds that trade more within their style category tend to underperform relative to other funds. Elton, Gruber, Das and Hlavka (1993) show that this is true across categories as well.

(b) Turnover for any period is measured as the dollar value of the portfolio that is traded relative to the size of the portfolio at the beginning of the period.

(c) Excess turnover may be a symptom of either trading on no information, or result in extra transaction costs that reduce the returns of the portfolio. Either way, it is costly.

(d) When constructing a portfolio of complex funds, it may be useful to rank them by turnover, so as to control for excess trading costs.

Leverage

(6) *Leverage*. (a) Ranking complex funds by leverage is a way to segment them by risk. The higher the leverage, the greater the importance of the higher-order moments of returns in risk management.

(b) The Sharpe ratio changes as the relative amounts of systematic and idiosyncratic risk vary. This may be seen with a simple example. Given a \$1 investment, let w_m be invested in systematic factors (the market) with expected excess return r_m and standard deviation σ_m. Let w_i be invested in deals comprising pure idiosyncratic risk with expected excess return r_i and standard deviation σ_i. The remaining money $(1 - w_m - w_i)$ is invested in the risk free asset. Given that the two risky returns are orthogonal, the Sharpe ratio is:

$$\text{Sharpe ratio} = \frac{w_m r_m + w_i r_i}{\sqrt{w_m^2 \sigma_m^2 + w_i^2 \sigma_i^2}}$$

We varied w_m and w_i both from 0 to 5, and reworked the Sharpe ratio. Figure 1 shows that the Sharpe ratio can change rapidly in some ranges when leverage changes. Hence, in addition to sorting funds by leverage, we also need to determine how quickly their leverage is changing. If funds compute their leverage on a daily basis, then a plot of the time series will provide a good sense of the swings in leverage. Also, the simple standard deviation of daily leverage will be a useful measure of changing risk, as well as a measure of how useful the Sharpe ratio might be in evaluating any given fund.

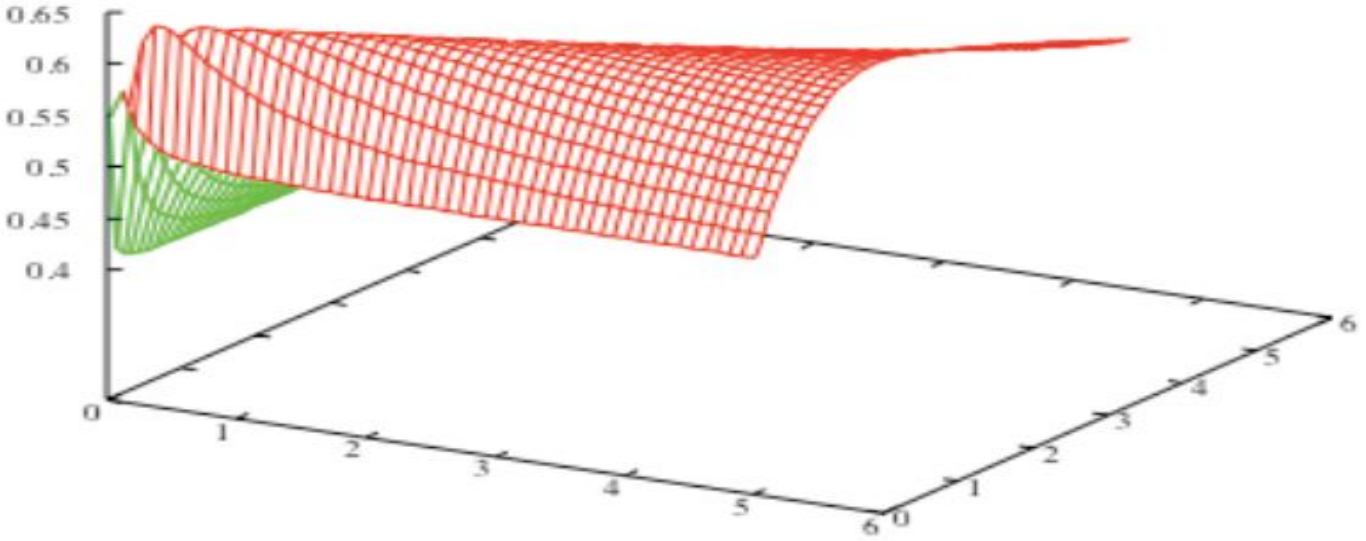


FIGURE 1. Sharpe ratio when systematic and idiosyncratic components are varied in weights. When the sum of the portfolio weight values on the x-axis (w_i) and y-axis (w_m) exceeds 1.0, the fund is borrowing money to make longer investments in risky assets, and is incurring extra leverage.

Factor Exposures

(7) *Factor exposures*. (a) Factor decompositions of the returns from complex funds offer insights into the risk in their portfolios. Sensitivity may be measured to (i) standard linear factors, and (ii) dynamic option-like factors. The former is less useful given the market-neutral focus of complex funds, and hence, factors of the latter type, where the specific deal nature of the fund is tracked are preferred.

(b) Fung and Hsieh (2001) point out that a style-based linear model may be an appropriate approach in analyzing the returns of hedge funds. They posit a simple model:

$$R_i = \alpha + \sum_k \beta_k \, SF_k + e_i$$

where the β_ks are the factor loadings and the *style factors* are denoted SF_k. A style-based model may be thought of as an asset class model extended to account for specific deal structures that are undertaken by hedge funds. Hence, using the terminology of Fung and Hsieh, the term "style" is used to connote both (i) "location" or asset class, and (ii) "strategy" or dynamic trading model.

Hedge Fund Exposures

(c) Details of appropriate factors to be used are detailed in two papers: Fung and Hsieh (1997) and Fung and Hsieh (2001). In the former paper, they use the following asset classes as factors: (i) three equity classes: MSCI U.S. equities, MSCI non-U.S. equities, and IFC emerging market equities, (ii) two bond classes: JP Morgan U.S. government bonds and JP Morgan non-U.S. government bonds, (iii) cash, i.e. the 1-month eurodollar deposit, (iv) commodities, i.e. the price of gold, and (v) currencies - the Federal Reserve Trade Weighted Dollar Index. This results in eight "location" factors.

(d) To obtain the style "strategy" factors, Fung and Hsieh (1997) use factor analysis on a set of hedge funds and commodity funds to obtain five distinct factors. At this point the approach is purely statistical, but these five factors are then identified with the following hedge fund categories: (i) Systems/Opportunistic, (ii) Global/Macro, (iii) Value, (iv) Systems/Trend Following, and (v) Distressed.

Extreme Value Risk

(8) *Extreme-value risk*. (a) The nature of complex fund trading strategies results in return distributions with greater tail risk. Hence, fitting the fund's returns to extreme-value distributions will offer useful insights for the purpose of risk management. This is one approach to quantifying the extent of deviation from Gaussian return distributions.

(b) Excellent descriptions of the use of extreme value theory (EVT) in finance are available from Embrechts, Klüppelberg, and Mikosch (1997) and Dowd (1999). EVT is about the statistical properties of the extreme values from a distribution, i.e., analyzing its tails. It focuses on the maxima or minima of distributions, or on the k highest/lowest values of the distribution.

Extreme Value Theorem

(c) Extreme Value Theorem: the distribution of extreme observations of X converges asymptotically to:

$$H(x;\xi,\mu,\sigma) = \begin{cases} \exp\left[-\left(1+\xi\frac{(x-\mu)}{\sigma}\right)^{-1/\xi}\right] & \xi \neq 0 \\ \exp\left[-e^{(x-\mu)/\sigma}\right] & \xi = 0 \end{cases}$$

where $x \in (-\infty, +\infty)$. The parameter μ is the mean and σ is the standard deviation, and ξ is known as the "tail index", a measure of the fatness of the tail. As ξ increases, the tails get fatter, and the risk usually increases, especially if it is in the left tail. When $\xi > 0$, the distribution converges to the Frechet distribution.

Hill Estimator

(d) Estimating the parameter ξ therefore gives a characterization of extreme value risk through the parameter for tail fatness. A common estimator is the Hill estimator, based on a tail cut off at the kth observation, and a comparison of this value with the average of values in the tail counted from that cut off. Hence, if we are computing the 5% VaR, then k is the fifth percentile observation. The Hill estimator is

$$\xi_k = \frac{1}{k}\left(\sum_{i=1}^{k} \ln(x_i)\right) - \ln(x_k)$$

Note that the tail index changes if we change k, hence the choice of k is important. A simple heuristic is to choose k for the VaR cut off level that is being used in the risk analysis of the portfolio.

VaR and Shortfall

(9) *VaR and Downside Risk.* (a) Value-at-risk (VaR) is the amount of portfolio loss that occurs within a specified probability level. If the change in the portfolio value is normally distributed with mean μ and standard deviation σ, then the 5% VaR is given by $(\mu - 1.645\sigma)$, i.e. the extent of loss below which only 5% of the probability mass of the distribution is left in the tail. Similar calculations may be done for other probability distributions. Downside risk examines the moments of losses (i.e. the average loss or variance) in the tail region defined by the VaR cut off. Using Gaussian distributions of loss does not depict complex fund returns well, and hence, a conservative approach is to use extreme-value distributions.

(b) However, these measures are complicated to calibrate and use, and hence may be approximated with the fat-tailed T-distribution. As we reduce the degrees of freedom, the T-distribution becomes increasingly fat-tailed. Choosing the degrees of freedom allows one to tune the extent of tail fatness as required.

(c) Another measure of downside risk is computed from *worst case scenario* analysis. In repeated simulations, we can compute the worst outcome from each simulation exercise. The distribution of these outcomes offers insights into how adverse outcomes might be on average when extreme negative outcomes occur.

Liquidity Risk

(10) *Liquidity risk and default risk.* (a) Many portfolios of complex funds comprise deals in (i) illiquid securities, or (ii) deal sizes that are substantive enough that their unwinding is enough to move prices adversely in a significant manner. Recent research documents that much of the returns to complex funds are attributable to their being compensated as providers of liquidity. Hence, ranking funds on the average liquidity of their assets might be a useful way to compare across funds. A simple measure of liquidity of individual assets is the bid-ask spread at which they trade. Other measures of liquidity may be found in work by Pastor and Stambaugh (2003), Chacko (2004), Chacko and Stafford (2004), and Sadka (2006).

Default Risk

(b) Other trades that are popular with complex funds are highly levered bets on default risk. These also need to be carefully assessed. One approach to doing this is to compute the credit deltas for these assets, and then apply suitable hedges as required.

(c) Complex funds also face correlation risk amongst their counterparties. Because they tend to undertake margin trades against a set of counterparties who may all retain similar exposures, complex funds are susceptible to the joint default of many of their counterparties, especially when the fund's positions are profitable. These risks may be assessed using simple default correlations embodied in standard heat maps. Hence, careful assessment of "concentration" risk is required. Another way to do this is to examine the equity correlations of the counterparties of the fund.

Sortino and Price Ratio

(11) *Sortino and Price (1994) ratio.* This ratio is a measure of downside risk thresholds. The measure is akin to the distance to default metric used in credit models. we may think of this as a "distance to failure" measure.

$$\text{SP Ratio} = \frac{r - MAR}{DSD}$$

where r is return and MAR is the minimum acceptable return (zero, risk free rate, etc) which is an acceptable lower barrier on return. DSD is downside standard deviation.

Lavinio *d*-ratio

(12) *Lavinio (1999) d-ratio.* This ratio is a skewness metric. It is defined as $d = l/w$, where l is the average value of losses, and w is that of wins. We propose an extension of this metric to the second moment by computing a new measure which we call d^2:

$$d^2 = \frac{\sigma_l^2}{\sigma_w^2}$$

which is the ratio of the downside variance to upside variance. The larger this ratio, the greater risk on the downside relative to the upside.

Using Quantopian

- Getting started: https://www.quantopian.com/tutorials/getting-started#lesson1
- Portfolio analysis using **pyfolio**: https://www.quantopian.com/research/notebooks/Tutorials%20and%20Documentation/6.%20Tutorial%20-%20Analyzing%20Backtest%20Results.ipynb
- Full documentation: https://www.quantopian.com/help#api-doco

Core Functions

https://www.quantopian.com/tutorials/getting-started#lesson2

initialize()

`initialize()` is called exactly once when our algorithm starts and requires `context` as input.

`context` is an augmented Python dictionary used for maintaining state during our backtest or live trading, and can be referenced in different parts of our algorithm. `context` should be used instead of global variables in the algorithm. Properties can be accessed using dot notation (`context.some_property`).

handle_data()

`handle_data()` is called once at the end of each minute and requires `context` and `data` as input. `context` is a reference to the same dictionary in `initialize()` and `data` is an object that stores several API functions which we will discuss in a later lesson.

The following example stores a string variable with the value 'hello' in `context`, and prints it out each minute.

```
def initialize(context):
    context.message = 'hello'

def handle_data(context, data):
    print context.message
```

Clone

before_trading_start()

`before_trading_start()` is called once per day before the market opens and requires `context` and `data` as input. It is frequently used when selecting securities to order. This will be explored in a later tutorial.

Referencing Securities

The best algorithms select securities to trade dynamically using pipeline. Pipeline is an advanced tool and will be covered in a later tutorial. To keep things simple, we'll start by manually referencing hand-picked securities. There are two ways to manually select securities on Quantopian: the `sid()` function, and the `symbol()` function.

sid()

The sid() function returns a security given a unique integer security ID (SID). `sid()` is a robust way to reference a security. The SID of a security remains constant despite ticker changes and as such, is a reliable way to reference a stock. *A security's SID never changes.*

For example, if we want to reference AAPL, we can use `sid()`, which brings up a text prompt allowing us to fill in a ticker symbol and gives us the corresponding security ID.

We can store the reference to AAPL in a context variable:

```
context.aapl = sid(24)
```

To reference this later in our code, we can simply refer to `context.aapl`. For example, to print out the security reference each minute, we can add `print context.aapl` to `handle_data()`:

https://www.quantopian.com/tutorials/getting-started#lesson3

Using a SID

To reference this later in our code, we can simply refer to `context.aapl`. For example, to print out the security reference each minute, we can add `print context.aapl` to `handle_data()`:

```
def initialize(context):
    context.aapl = sid(24)

def handle_data(context, data):
    print context.aapl
```

Clone

symbol()

Alternatively, the symbol() function returns a security given a ticker symbol. `symbol()` is not robust to ticker changes unless you specify a reference date with `set_symbol_lookup_date('YYYY-MM-DD')`. It is strongly discouraged to us `symbol()` to reference a security without setting a reference date.

Ordering Securities

https://www.quantopian.com/tutorials/getting-started#lesson4

There are several functions that can be used to order securities in the Quantopian API. In this tutorial, we are going to focus on the order_target_percent() function. `order_target_percent()` allows us to order securities to a target percent of our portfolio value (sum of open positions value + cash balance).

`order_target_percent()` requires two arguments: the asset being ordered, and the target percent of the portfolio. For example, the following line would order 50% of our portfolio value worth of AAPL:

```
order_target_percent(sid(24), 0.50)
```

To open a short position (a position that profits when the security will drop in price), we can simply enter a negative target percent:

```
order_target_percent(sid(24), -0.50)
```

The following example takes a long position in Apple stock worth 60% of our portfolio value, and takes a short position in the SPY ETF worth 40% of our portfolio value:

Clone

```
def initialize(context):
    context.aapl = sid(24)
    context.spy = sid(8554)

def handle_data(context, data):
    # Note: data.can_trade() is explained in the next lesson
    if data.can_trade(context.aapl):
        order_target_percent(context.aapl, 0.60)
    if data.can_trade(context.spy):
        order_target_percent(context.spy, -0.40)
```

“data” object

https://www.quantopian.com/tutorials/getting-started#lesson5

The `data` object contains functions that allow us to look up current or historical pricing and volume data for any security. `data` is available in `handle_data()` and `before_trading_start()`, as well as any scheduled functions. Let's have a look at some of `data`'s functions:

`data.current()`

data.current() can be used to retrieve the most recent value of a given field(s) for a given asset(s). `data.current()` requires two arguments: the asset or list of assets, and the field or list of fields being queried. Possible fields include `'price'`, `'open'`, `'high'`, `'low'`, `'close'`, and `'volume'`. The output type will depend on the input types. To get the most recent price of AAPL, we can use:

```
data.current(sid(24), 'price')
```

The following line of code gets the most recent price of AAPL and SPY and returns a pandas Series indexed by asset:

```
data.current([sid(24), sid(8554)], 'price')
```

The following line will get the last known low and high prices of AAPL and SPY from and returns a pandas DataFrame (indexed by assets, fields as columns):

```
data.current([sid(24), sid(8554)], ['low', 'high'])
```

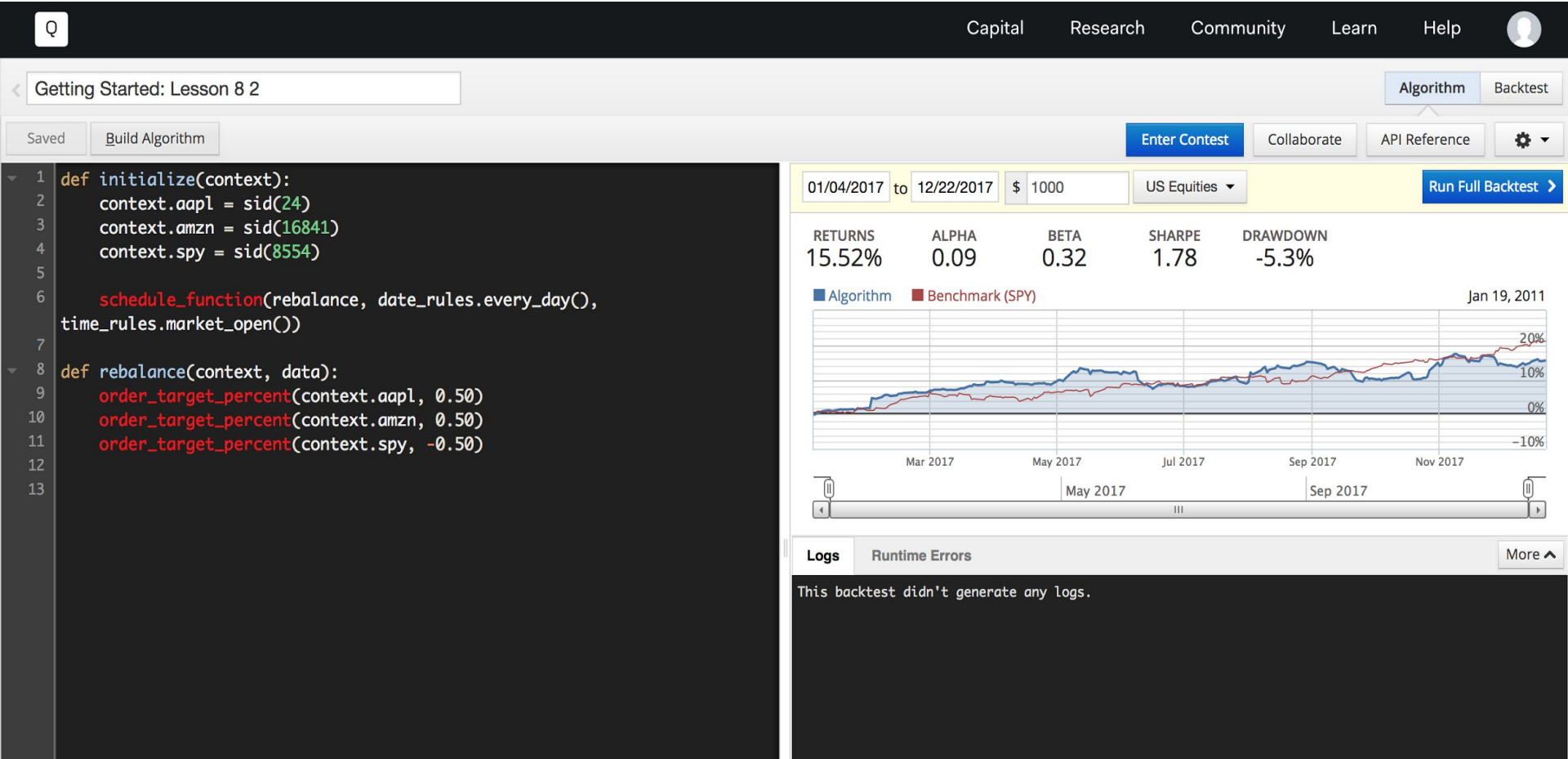
`data.can_trade()`

data.can_trade() is used to determine if an asset(s) is currently listed on a supported exchange and can be ordered. If `data.can_trade()` returns `True` for a particular asset in a given minute bar, we are able to place an order for that asset in that minute. This is an important guard to have in our algorithm if we hand-pick the securities that we want to trade. It requires a single argument: an asset or a list of assets. The following example checks if AAPL is currently listed on a major exchange:

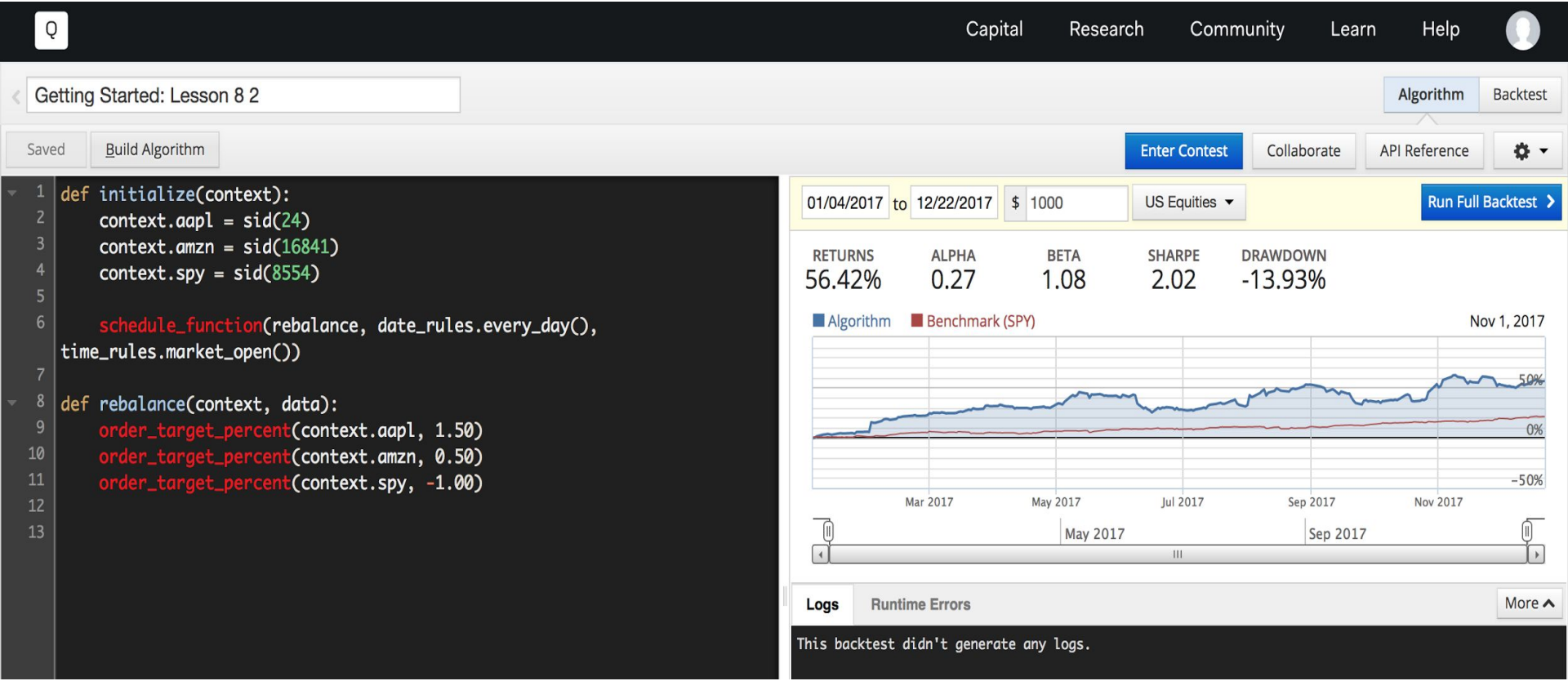
```
data.can_trade(sid(24))
```

Example

Clone from Lesson 8:
https://www.quantopian.com/tutorials/getting-started#lesson8

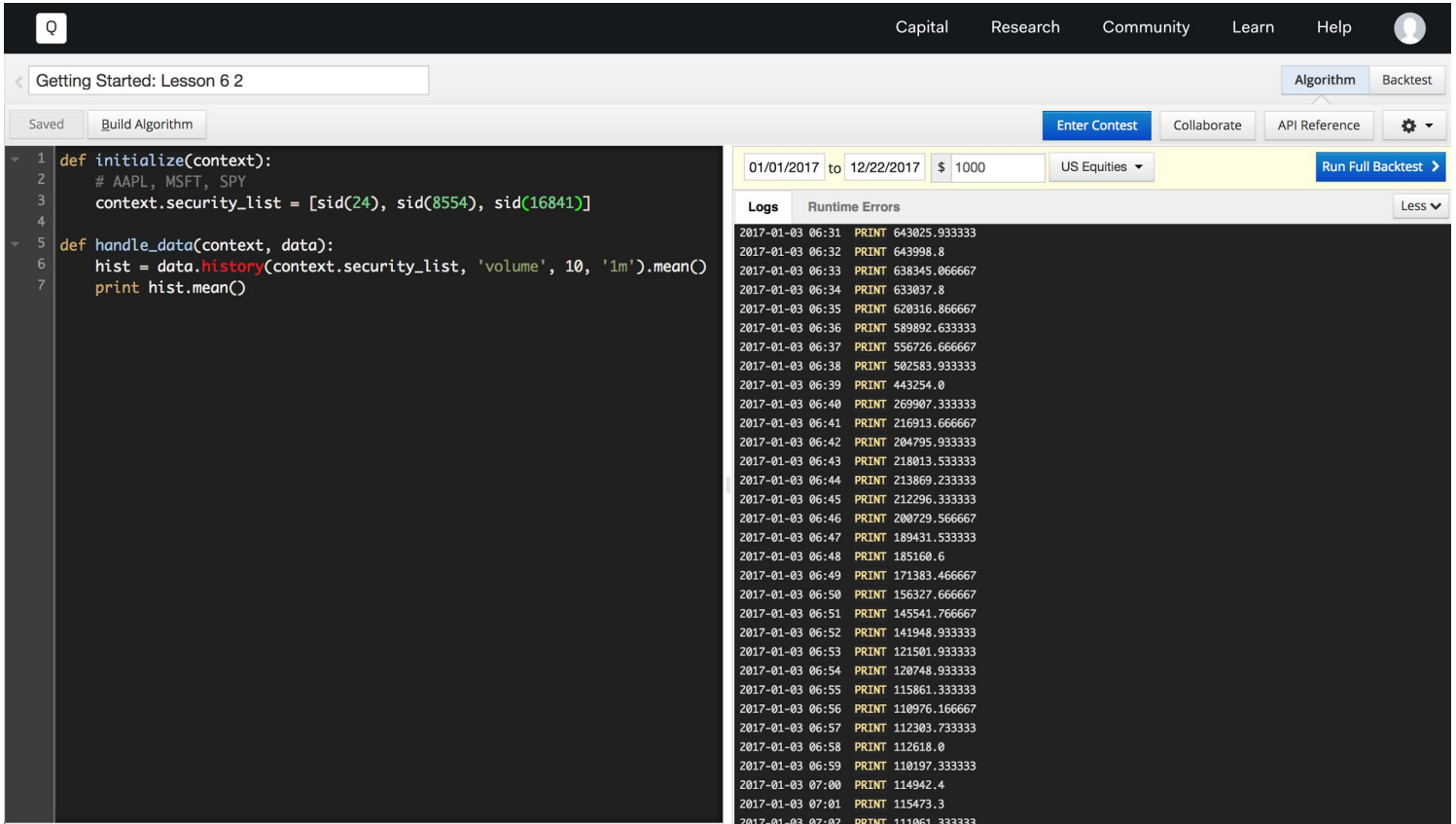


Same securities, new weights



history() function

The following example prints the mean trading volume of a list of securities over the last 10 minutes.



Portfolio Statistics

Q

Capital Research Community Learn Help

Getting Started: Lesson 8 3

Algorithm Backtest

Save Build Algorithm

Enter Contest Collaborate API Reference

```
def initialize(context):
    context.aapl = sid(24)
    context.spy = sid(8554)

    schedule_function(rebalance, date_rules.every_day(),
time_rules.market_open())
    schedule_function(record_vars, date_rules.every_day(),
time_rules.market_close())

def rebalance(context, data):
    order_target_percent(context.aapl, 0.50)
    order_target_percent(context.spy, -0.50)

def record_vars(context, data):

    long_count = 0
    short_count = 0

    for position in context.portfolio.positions.itervalues():
        if position.amount > 0:
            long_count += 1
        if position.amount < 0:
            short_count += 1

    # Plot the counts
    record(num_long=long_count, num_short=short_count)
```

01/01/2017 to 12/22/2017 $ 1000 US Equities Run Full Backtest

RETURNS	ALPHA	BETA	SHARPE	DRAWDOWN
15.27%	0.08	0.31	1.75	-5.31%

Logs Runtime Errors More

This backtest didn't generate any logs.

Numerai

- https://numer.ai/
- https://hackernoon.com/numerai-walkthrough-quantitative-analysis-machine-learning-for-fun-and-profit-3dcdccabd920