Sponsored by:

DATA SCIENCE INSTITUTE

The World Data Science Institute is a Specialized Consulting Agency offering DSaaS (Data Science as a Service)

Meet Cohort Team 1

Anade Davis - Data Science Manager - LINKEDIN

Sarah Guo - Data Scientist

Christopher Rutherford - Data Scientist - LinkedIn

Tanjeel Ahmed - Data Science Researcher

Gabe Smithline - Quantitative Analyst- LinkedIn

Berkalp Altay - Data Analyst - LinkedIn

Table of Contents

What is Quantopian?

- A company founded in 2011 by Founders John Fawcett (CEO) and Jean Bredeche (CTO).
- Quantopian is a Fintech (Financial Technology) Company that provides the tools and datasets for anyone to pursue quantitative analysis.
- There are three integral pieces to the Quantopian community: learning through tutorials and documentation, testing and developing ideas and strategies with real data, and sharing findings with the Quantopian forum
- Quantopian allows users to create their own strategies for trading and, if they choose, share them with the community

What is Quantopian?

- Quantopian is a place for people to learn Quantitative Finance
- Quantopian Provides
  - Tools
  - Tutorials
  - Guides
  - Forums

What is Quantopian?

- In July 2016 Steven A. Cohen, CEO of Point72 Asset Management invested $250 million into Quantopian Fund.
- In 2018, Quantopian teamed up with FactSet, a company that provides software and data solutions, to create Quantopian Enterprise, which allows professional quants and financial analysts to quickly and easily backtest their trading strategies
- Quantopian Enterprise is more suited for professionals with industry experience

What is Quantopian?

- Quantopian is written in Python, allowing users to quickly and easily share their findings with the community with Python notebooks
- Its main library is Zipline, which is used for building, backtesting, and putting live-trading algorithms into practice
- Zipline is built and maintained by the Quantopian engineering team as opposed to being community-maintained

What is Zipline?

- Zipline is essentially the backbone of Quantopian algorithms and portfolios, as datasets are considered zipline objects and back tests are run through zipline
- A lot of these functions and data types are implicitly called as zipline objects - that is, when they are imported, zipline takes care of the hard work without the user having to specify that the object is a zipline object

What other tools are there?

- Aside from Zipline, several other libraries are also used to help create the model from setup to backtesting

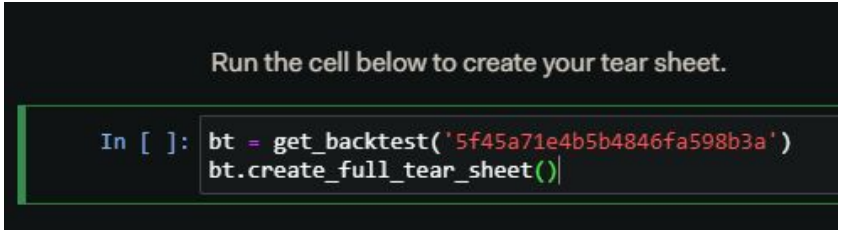
- Pipeline is a library used to create, as the name suggests, pipelines for importing and processing data. This is convenient for calculating rolling averages, returns, and other common techniques used for financial analysis and trading algorithms

What other tools are there?

- For portfolio management, Quantopian has developed a tool called pyfolio
- Pyfolio provides users with “tear sheets” of their backtested model which provide an overview of a portfolio’s performance. It measures the risk associated with the portfolio through measures such as volatility and return rate
- This tear sheet can also provide an overview of how a portfolio might perform compared to some given ETF, such as the S&P 500, through backtesting

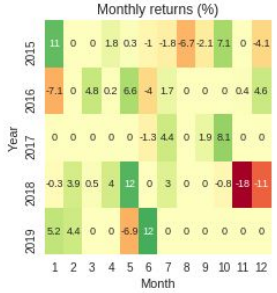
Pyfolio: An Example

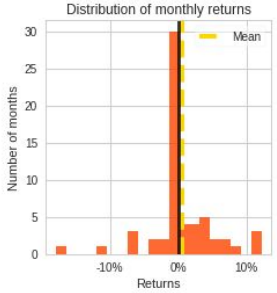
- When running backtests on Quantopian, there is direct access to pyfolio to get a more thorough assessment of a portfolio's performance
- As a quick example, we look at a simple trading strategy involving buying/selling stock based on 7-day and 50- day moving averages
- Once our backtest is finished, a block of code is provided for us to quickly create our tear sheet



Pyfolio: An Example

- Running the block of code will provide the user with a variety of statistics and graphs to assess the portfolio's performance
- We benchmarked our portfolio against SPY as a baseline, but since our trading method is quite basic and used purely for demonstrating pyfolio, its performance is not stellar
- These plots show us the return overtime (right) and how these returns are distributed throughout each month/year (below)[1]

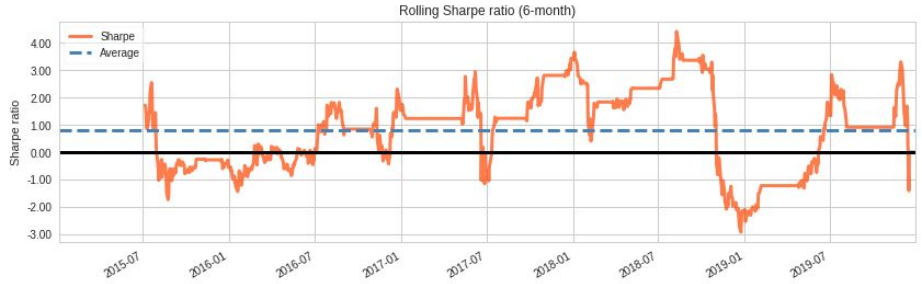




Pyfolio: An Example

Start date	2015-01-05
End date	2019-12-31
Total months	59
	Backtest
Annual return	5.212%
Cumulative returns	28.846%
Annual volatility	17.192%
Sharpe ratio	0.38
Calmar ratio	0.15
Stability	0.48
Max drawdown	-35.612%
Omega ratio	1.12
Sortino ratio	0.53
Skew	-0.56
Kurtosis	15.21
Tail ratio	0.91
Daily value at risk	-2.14%
Gross leverage	0.31
Daily turnover	5.404%
Alpha	-0.02
Beta	0.71

- Here (left) we can see more metrics about the portfolio's runtime, volatility, returns, and other summary statistics, such as alpha and beta
- The graphs to the right also show how the volatility and Sharpe ratio changed overtime



Pyfolio: An Example

- Clearly, pyfolio is an extremely useful tool for investors and analysts when assessing their portfolio - it provides many graphs and tabular outputs to help people identify where their portfolio's strengths and weaknesses are, how it performs relative to some specified benchmark
- For more complex portfolios, pyfolio is an invaluable tool as it also provides a breakdown of the returns and risk involved for each sector of the portfolio - tech, real estate, energy, etc.
- Our portfolio only consisted of Apple stock, so only information about technology is provided, but more stocks/portfolio diversity would populate the other fields

Exposures Summary	Average Risk Factor Exposure	Annualized Return	Cumulative Return
basic_materials	0.00	0.00%	0.00%
consumer_cyclical	0.00	0.00%	0.00%
financial_services	0.00	0.00%	0.00%
real_estate	0.00	0.00%	0.00%
consumer_defensive	0.00	0.00%	0.00%
health_care	0.00	0.00%	0.00%
utilities	0.00	0.00%	0.00%
communication_services	0.00	0.00%	0.00%
energy	0.00	0.00%	0.00%
industrials	0.00	0.00%	0.00%
technology	0.99	7.05%	40.51%
momentum	0.12	0.81%	4.10%
size	3.64	-3.24%	-15.15%
value	-0.60	0.02%	0.12%
short_term_reversal	0.48	0.17%	0.83%
volatility	-0.39	0.33%	1.67%

What other tools are there?

- Quantopian has also developed a library called Alphalens
- Alphalens provides users with insight into relevant statistics and plots about the performance of predictive stock factors, such as the average daily return, daily spread, and other information, such as how close the forecasts are to the actual returns

Alphalens

- Alphalens is a tool to analyse given alpha factors to predict future returns
- Alpha factors express a relationship between a variable and future returns
- Alphalens is unique to Quantopian, as, like pyfolio, it is developed and maintained by a small team working with Quantopian

Why Alphalens is useful

- It is fast
  - Quicker to analyze in alphalens than run a whole back test
- Extremely visual
  - Converts numbers to charts
  - Easy to use

Alphalens example

- This use of alphalens utilizes the use of
  - Return analysis
  - Information coefficient analysis
  - Turnover Analysis
  - Grouped Analysis

With a signal and pricing data creating a factor "tear sheet" is a two step process:

```
import alphalens

# Ingest and format data
factor_data = alphalens.utils.get_clean_factor_and_forward_returns(my_factor,
                                                                   pricing,
                                                                   quantiles=5,
                                                                   groupby=ticker_sector,
                                                                   groupby_labels=sector_names)

# Run analysis
alphalens.tears.create_full_tear_sheet(factor_data)
```

Alphalens example results

- A Tear sheet is created this sheet is a list of analysis on specific financial returns and specific statistical factors

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

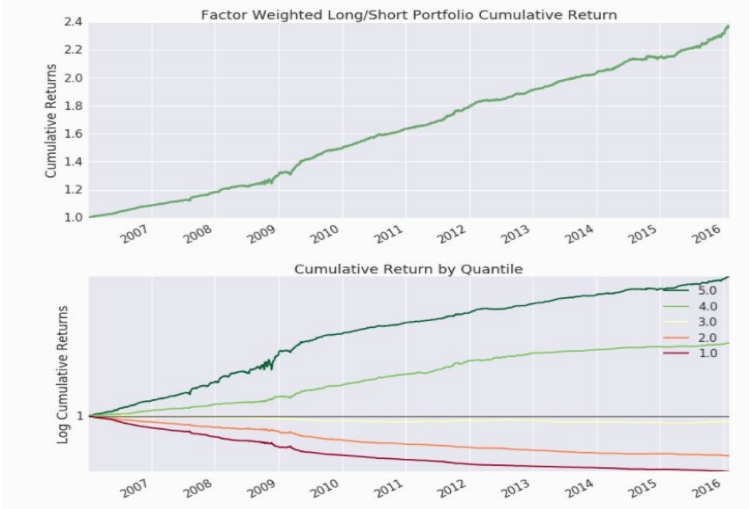
Turnover Analysis

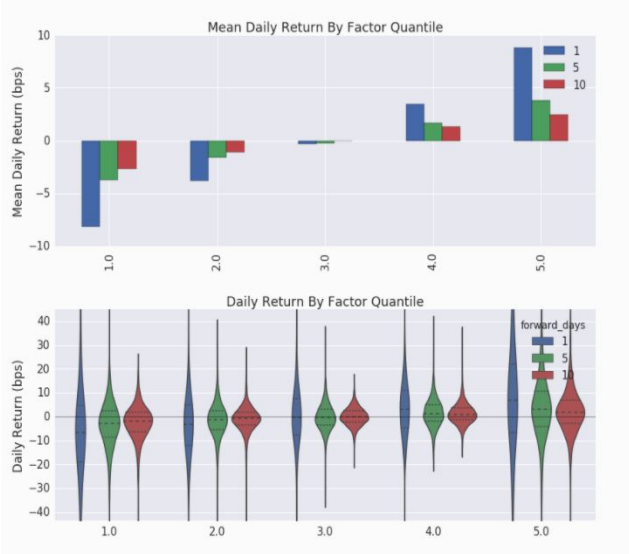
	Top Quantile	Bottom Quantile
Mean Turnover	0.44	0.452

Mean Factor Rank Autocorrelation 0.625
dtype: float64

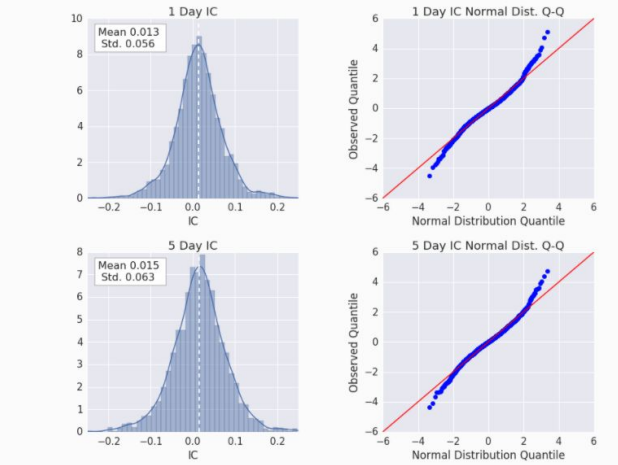
Alphalens Visualization Example

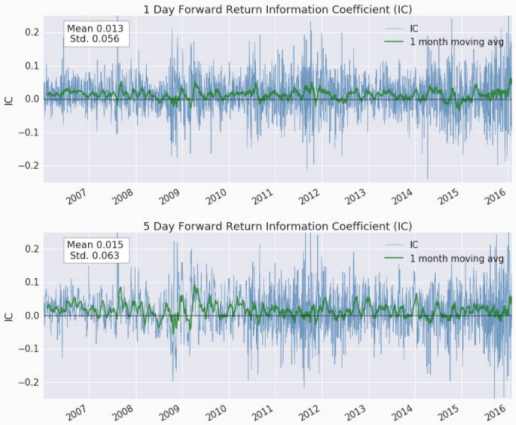
- Here are visualisations of the the tear sheet in the prior slide.

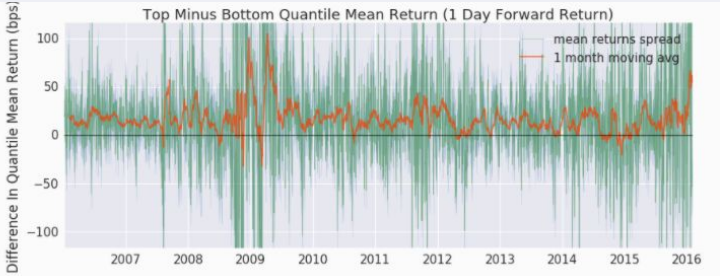


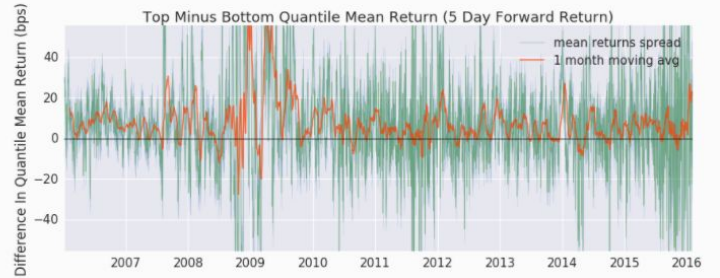


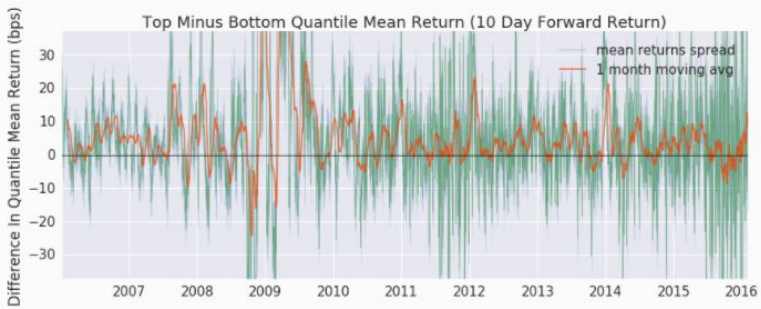
Alphalens Visualizations Continued











Pipeline

- Allows for the computation of values for an asset based on a following window of data
- Allows for easier filtering of relevant information
- Helps to calculate the weight of assets in a portfolio
- Places order to move portfolio allocation

Why use Quantopian Pipeline?

- This helps to query large sets of data
- Helps speed up computations on large sets of data
- Handles adjustments and asset delistings efficiently

Computations in a Pipeline

- Factors -> function from an asset and moment in time made into a numerical value
  - Computing target weights
  - Generating alpha signal
  - Constructing filters
- Filter -> function from an asset and moment in time made into a boolean
- Classifiers -> function from an asset and moment in time made into a categorical value

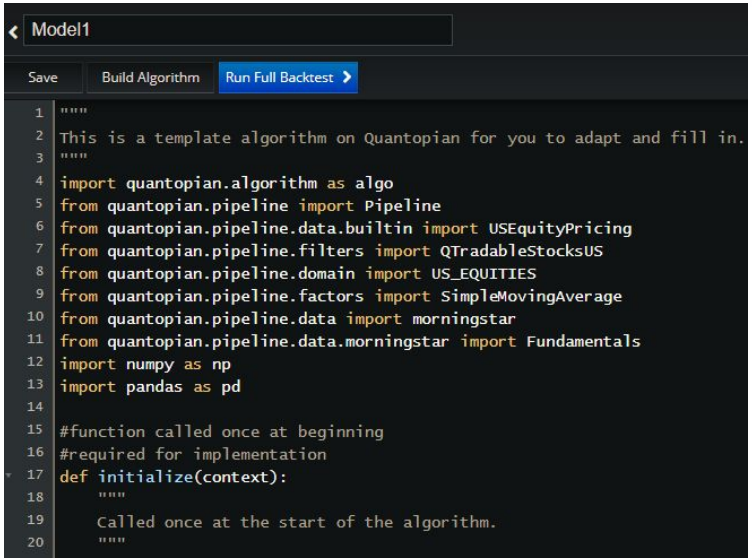
Examples of each Computation

- Factors
  - 10 day moving average
  - Just the most recent price of a security
- Filters
  - A filter indicating a security's price is below a certain point
- Classifier
  - Group assets based on the exchange they are traded on

Quantopian IDE

- IDE -> Integrated Development Environment
  - Python development environment made to help design trading strategies using the Algorithm API

- Workflow
  - Writing
  - Building
  - Backtesting



IDE: Writing

- 1st step of development workflow
- Features
  - Built in code validation
  - Debugging
  - Output logs

IDE: Building

- “Build” will check code for syntax errors and run a backtest within the IDE
- Runs a backtest that uses the same backtest engine as a full backtest
- The IDE backtest is less detailed compared to the full backtest

IDE: Output Logs

- The output in the bottom right can have up to three types of errors
  - Build errors -> essentially syntax errors
  - Log outputs -> this section just displays the output
  - Runtime Errors -> Displays any errors present in the algorithm, if it is empty then your algorithm ran

Model & Background

- Analysis of S&P 500
  - What it is
  - Why we decided to analyze it
- Model
  - Alpha
  - Beta
  - Sharpe
  - Returns
- Strategy
  - Background
  - Analysis/Results

What is the S&P 500?

- Standard and Poor's 500 is a market-capitalized index that lists 500 large US companies on US stock exchanges, it is thought to be one of the best representations of the US stock market
- The 10 largest companies in the index account for about 26% of its market capitalization

History of the S&P 500

- Introduced in 1957 as a tracking of 500 large US corporations listed on the NYSE and NASDAQ
- Intended to be a representation of the US economy
- To the right is a 90 year chart that gives the

  Historical closing price of each month

Why we picked to analyze the S&P 500

- Although we are in a pandemic the S&P is at an all time high
- Big tech is pushing it up
- Using the S&P 500 as a benchmark is useful for deciding if our model can outperform the overall performance of the S&P 500

NASDAQ 100

- One of many cap-growth indices
- It includes 100 of the world's largest non-financial companies listed on the NASDAQ stock market, such as Apple, Tesla, Microsoft, Intel, Starbucks, and more
- Of all the industries listed on the NASDAQ 100, technology accounts for 54% of the index's weight

History of the NASDAQ 100

- Began in 1985
- Always in shadow of NYSE
- Nasdaq created this index, along with the NASDAQ 100-Financial, with the hope that futures and options would be created on them and that mutual funds would adopt them as a benchmark

Why we picked to Analyze the NASDAQ 100

- The NASDAQ is a tech focused exchange, and the NASDAQ 100 is a tech focused index
- The reason the S&P continues to rise is because of the booming tech industry; it's reasonable to think that there might be some correlation between the two

The Strategy

- The strategy we picked was moving average crossover and using z-scores for daily percent change
- What are these?
  - Short term trading strategies
  - Moving average crossover is short to medium term, while z-score is very short term (daily)
  - Moving average crossover is useful for detecting a shift in trend of the asset
  - Z-scores reveal the “strength” of daily fluctuations
  - It can be risky but rewarding

How we plan to employ it

- We will calculate the 7- and 50-day moving averages of the closing price and z-scores of the daily percent change
- If the moving averages cross over each other, this usually indicates a change in trend (up to down or down to up)
- Fluctuations in these values will indicate when to buy and sell
- Z-scores above 1.5/below -1.5 indicate a sharp change in stock price from the previous day, which may indicate the beginning of an upward/downward trend

Model Performance Overview

- We ran our model's backtest from January 3, 2011 to December 31, 2019
- Our benchmark yielded a return of 200% while our strategy yielded a return of 230%
- While the difference is not enormous, our strategy was able to avoid some of the dips, such as in 2016, that are visible in the benchmark

BENCHMARK RETURNS

A benchmark is a standard against which the performance of a security, mutual fund or investment manager can be measured.

Generally, broad market and market-segment stock and bond indexes are used for this purpose.

Benchmarks are indexes created to include multiple securities representing some aspect of the total market.

Measuring Investment Risk

- The Capital Asset Pricing Model, a seminal model for asset pricing, predicts higher returns for higher risk.
- Thus, we need to manage investment risk.
- Four important measures of investment risk are:
  1. Volatility
  2. Sharpe Ratio
  3. Beta
  4. Alpha

Keep in mind that these are all **historical measures**. That is, they are calculated using data from the past, not from our forecasts.

VOLATILITY

Volatility is defined as a measure of the variation in the price of an asset over time.

Higher volatility is naturally associated with greater potential for larger losses.

There are two volatility measures commonly used in the financial industry: realized and implied.

Volatility

- Volatility is mathematically defined as the standard deviation of the portfolio's returns
- More specifically, it tells us, on average, how far from the average return rate the daily return rate is
- Larger volatility indicates that returns are more likely to be farther from the average return rate, while smaller volatility indicates that returns are less likely to stray from from the average
- In other words, higher/lower volatility indicates higher/lower risk and thus higher/lower returns

Model Volatility

- Our model yielded an annual volatility of 19.29%, exhibiting slightly higher volatility than our benchmark stock
- This means with an annual return of 14.22%, annual returns typically range from anywhere from -5% to 33% (14%±19%)
- However, this range is typically skewed more towards the positive returns due to the behavior of the market, as the overall trend for this time period was positive

TOTAL RETURNS

Total return is the amount of value an investor earns from a security over a specific period typically one year.

Total return is expressed as a percentage of the amount invested.

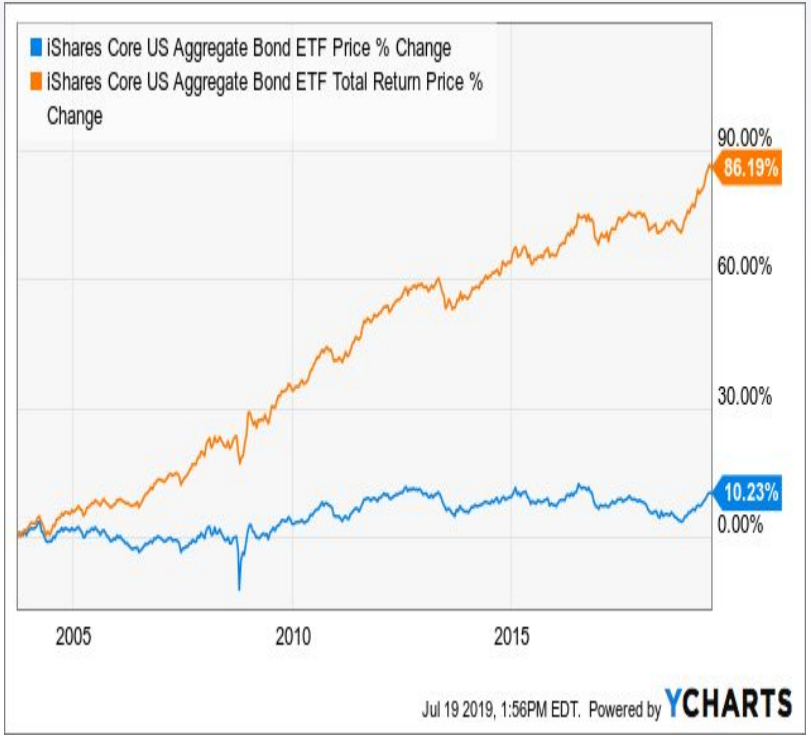
For example, a total return of 20% means the security increased by 20% of its original value due to a price increase, distribution of dividends (if a stock) or capital gains (if a fund).

Total return is a strong measure of an investment’s overall performance.

MAX DRAWDOWN

A maximum drawdown (MDD) is the maximum observed loss from a peak to a trough of a portfolio before a new peak is attained.

Maximum drawdown is an indicator of downside risk over a specified time period.



SHARPE RATIO

The Sharpe Ratio measures the expected excess return of an investment in relation to its return volatility.

The Sharpe ratio aims to determine how much additional return an investor can receive with the additional volatility on account of holding riskier assets.

A Sharpe ratio of 1 or more is considered to be a better risk-to-reward proposition for the investor.

When comparing two assets versus a common benchmark, the one with a higher Sharpe ratio provides better return for the same risk.

The Sharpe Ratio

- The Sharpe ratio is used to calculate the risk-free adjusted return of a portfolio
- It is more practical to use as a comparison with the performance of other portfolios, but in general, the higher it is, the better
- For simplicity, Quantopian assumes a risk-free rate of 0

$$Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p}$$

where:

R_p = return of portfolio

R_f = risk-free rate

σ_p = standard deviation of the portfolio's excess return

Example for Sharpe Ratio

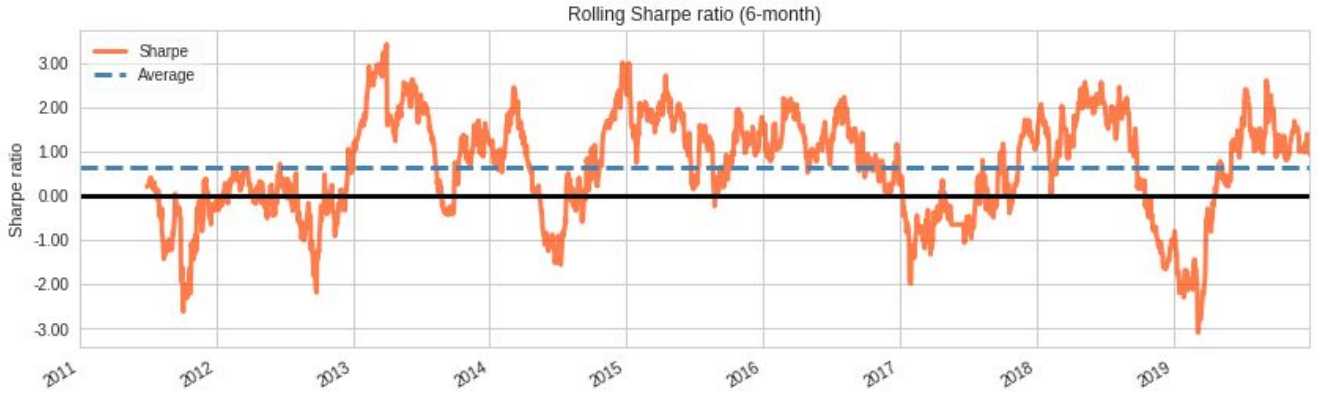
- Let's say the risk-free interest rate is 5%.
- Portfolio A has 15% return and 8% standard deviation.
- Portfolio B has 12% return and 5% standard deviation. This gives us Sharpe Ratios of 1.25 and 1.4 for Portfolios A and B respectively.
- With a higher Sharpe Ratio, Portfolio B was able to generate a higher return on a risk-adjusted basis. (*Investopedia: Sharpe Ratio*)

What is a good Sharpe Ratio?

- For some insight, a Sharpe ratio of 1 or better is good, 2 or better is very good, and 3 or better is excellent. (*Investopedia: Sharpe Ratio*)
- Standard deviation and risk both have a strong influence on the Sharpe ratio, so it can be difficult to obtain a "high" (say, above 1.5) Sharpe ratio (low/medium risk, high returns)

Model Sharpe Ratio

- Quantopian uses a 6 month rolling Sharpe ratio
- Our model produced a Sharpe Ratio of .79 which is not necessarily good but it is not necessarily bad, in general a Sharpe Ratio of 1 or higher is considered acceptable



Beta

Beta, also known as the beta coefficient, is a measure of the volatility, or systematic risk, of a security or a portfolio in comparison to the market as a whole. Beta is calculated using regression analysis.

You can think of it as the tendency of an investment's return to respond to swings in the market. (*Forbes*)

A market portfolio, which the S&P 500 is assumed to approximate, has a beta of **1** by definition.

BETA

Beta measures the relative volatility of a portfolio or mutual fund against its benchmark index.

This volatility or swing gives you the systematic risk of the security or portfolio when we compare it to the market as a whole.

A beta greater than one indicates higher-than-benchmark volatility and A beta less than one indicates lower-than-benchmark volatility.

Example for Beta

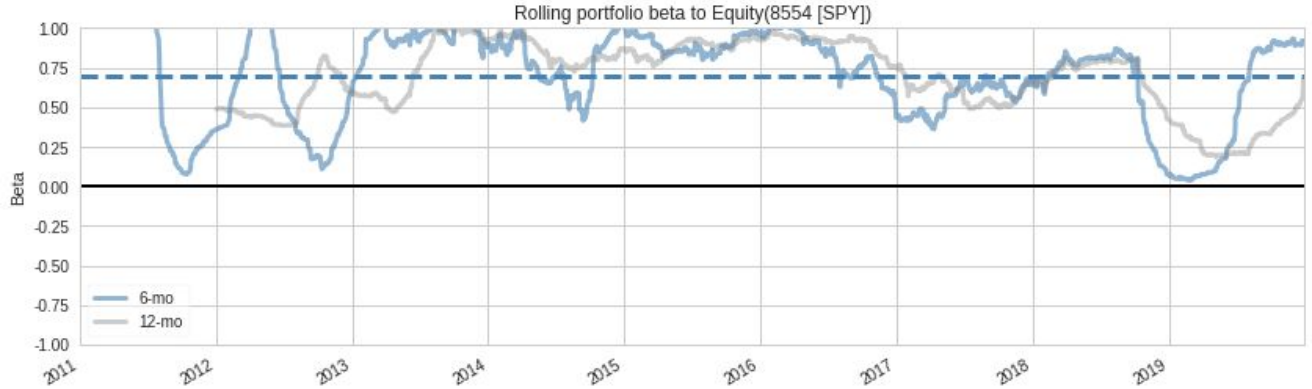
- Let's say the Apple stocks (ticker: APPL) have a beta of 1.06. That means, Apple stocks are theoretically **6% more volatile** than the market.
- For every 1% change in the price of the S&P 500 (assuming S&P 500 is the market portfolio), Apple's stock price changes by 1.06 % in the same direction as the S&P 500 (i.e. 1% increase in the S&P 500 means 1.06% increase in Apple's stock price).

Lower or Higher Beta: Which is Better?

- **Risk-averse investors** (a.k.a. conservative investors) such as retirees seeking a steady income are attracted to **lower beta**.
- **Risk-tolerant investors** who seek bigger returns are often willing to invest in **higher beta** stocks (*Investopedia: Alpha vs. Beta*).
- **CAPM** uses beta as the main component to model the relationship between returns and risk.

Model Beta Value

- Our model had a beta value of 0.64
- This indicates that our portfolio is only about 64% as volatile as our benchmark
- In general, the market has a Beta value of 1. Anything less is considered less volatile and thus less possible reward, while a beta of more than 1 is more volatile, which leads to more risk



Alpha

Alpha is a measure of an investment's performance on a **risk-adjusted basis**. It takes the volatility (price risk) of a security or fund portfolio and compares its risk-adjusted performance to a benchmark index such as the S&P 500.

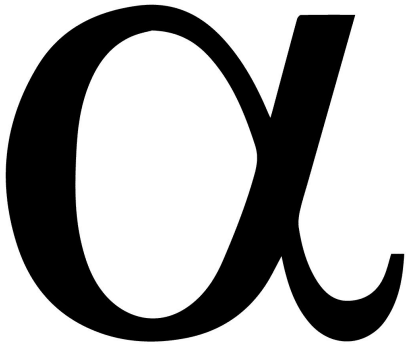
The excess return of the investment relative to the return of the benchmark index is its alpha. (*Forbes*)

ALPHA

Alpha (also called the "*holy grail of investing*") measures the performance of an investment portfolio against its benchmark index.

The alpha is the difference between the returns of your portfolio and the returns of the benchmark – which means the alpha can be positive or negative.

A positive alpha of one means the portfolio has outperformed the benchmark by 1 percent. Likewise, a negative alpha indicates the underperformance of an investment.



Example for Alpha

- The alpha is often a number like 3 and -5. It is a percentage value. That is, if we have set the benchmark as the S&P 500 and the alpha is 3, that means we have an investment portfolio that has performed **3% better** than the S&P 500. (*Investopedia: Alpha vs. Beta*)

- **Important Point: A higher alpha is always better.**

Model Alpha Value

- Our model yielded an Alpha value of 0.07, so we can interpret this as our risk-adjusted model managing to narrowly outperform our benchmark.

- An alpha of 0 is considered to be the benchmark alpha - this would indicate a portfolio that is performing identically to the overall market

Model Summary

- Overall, our portfolio's strategy managed to outperform our benchmark stock
- The use of relatively simple methods - crossing averages and percent changes - makes our portfolio less risky at the expense of less reward
- Our main measures of risk were all within a margin that indicated the model is indeed more profitable and less risky than the general market, coming at the expense of possibly missing out on returns from taking more risks
- However, even with a beta of 0.64, indicating our model is less volatile than the general market, we still managed to outperform the benchmark

Helpful Links

- https://pythonprogramming.net/finance-programming-python-zipline-quantopian-intro/
- https://www.quantopian.com/posts/machine-learning-on-quantopian-part-1-basics
- https://github.com/Gitlitio/quantopian-tools
- https://www.quantopian.com/
- https://www.quantopian.com/docs/user-guide/tools/algo-api
- https://towardsdatascience.com/hackers-guide-to-quantitative-trading-quantopian-python-part-2-f6ec90e1cb68
- Relevant Quantopian algorithms:
  - $GLD/$IAU Pairs trading template
  - https://www.quantopian.com/posts/z-score-algorithm
  - https://www.quantopian.com/posts/moving-average-crossover-1

References

- https://quantopian.github.io/alphalens/
- https://www.macrotrends.net/2324/sp-500-historical-chart-data
- https://www.nasdaq.com/nasdaq-100
- https://www.investopedia.com/articles/basics/09/simplified-measuring-interpreting-volatility.asp
- https://www.investopedia.com/terms/s/sharperatio.asp
- https://www.investopedia.com/ask/answers/010815/what-good-sharpe-ratio.asp
- https://www.forbes.com/2007/11/05/risk-alpha-beta-pf-education-in_rl_11050investopedia_inl.html#73f122bc7672
- https://www.investopedia.com/ask/answers/102714/whats-difference-between-alpha-and-beta.asp