Backtesting a Trading Strategy in Python

Disclaimer

This workshop is for educational 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 the speakers. 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. The speakers are not responsible for the losses incurred due to the buying and selling of securities.

Outline

- Why Backtest ?
- Methods of Backtesting
- Types of Backtesting algorithm
- Backtest analysis
- Backtest of Time-series Momentum
- Backtest of Pairs-trading

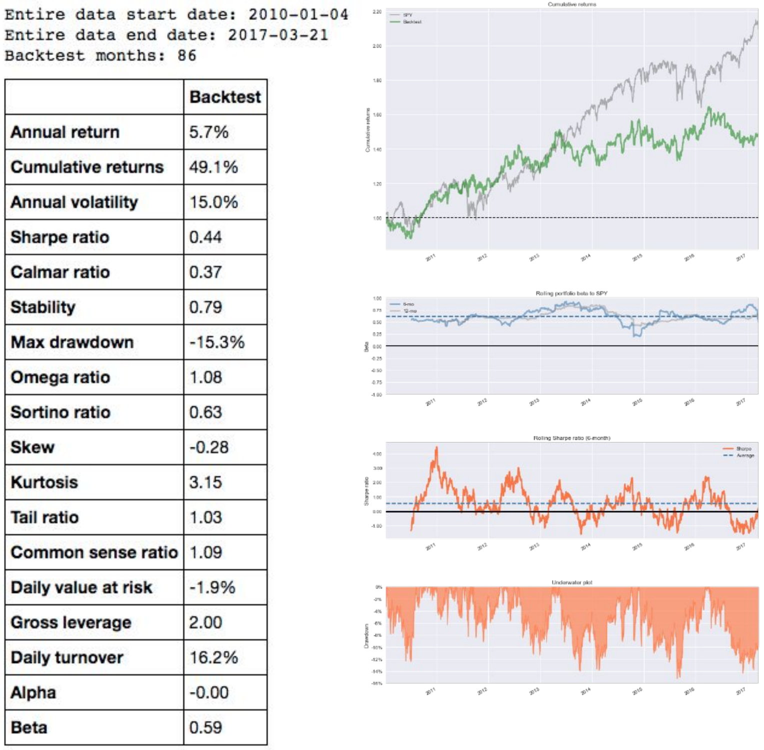
Motivation behind Backtesting

- Does the trading strategy you have hypothesised even perform ?

- Backtest is a process of testing the historical performance of a trading strategy. It is used as to check the if a strategy works on past and what performance we can expect in future.

- A backtest is a simulation which can be used for sanity check on the hypothesis behind the strategy and various factors under a given scenario of the market.

- Backtest is not an experiment and shouldn’t aimed to prove profitability of a strategy as it is only historical performance.



```
Entire data start date: 2010-01-04
Entire data end date: 2017-03-21
Backtest months: 86
```

	Backtest
Annual return	5.7%
Cumulative returns	49.1%
Annual volatility	15.0%
Sharpe ratio	0.44
Calmar ratio	0.37
Stability	0.79
Max drawdown	-15.3%
Omega ratio	1.08
Sortino ratio	0.63
Skew	-0.28
Kurtosis	3.15
Tail ratio	1.03
Common sense ratio	1.09
Daily value at risk	-1.9%
Gross leverage	2.00
Daily turnover	16.2%
Alpha	-0.00
Beta	0.59

Seven Sins of Quantitative Investing

1. *Survivorship bias*: Ignoring the stocks that have gone bust or delisted in past and only considering the existing stocks in the sample. This can lead to overestimation of historical performance. E.g.- Only considering the current composition of SP500.

2. *Look-ahead bias*: Using information that is not available at the current moment.

3. *Storytelling*: Making up a story ex-post to justify some random pattern.

4. *Data snooping*: Using the test sample for tuning the strategy and improving the backtest.

5. *Transaction cost*: Ignoring the transaction cost results inaccurate backtest results. While including transaction cost some strategies that were profitable earlier may fail.

6. *Outliers*: A backtest is severely skewed if it performance (profit or loss) is heavily dependent on few extreme outcomes that is observed in future. E.g.: Flash crash.

7. *Shorting*: Shorting involves finding lender who can lend securities, which is dependent on inventory, relative demand and markets.

Outline

- Why Backtest ?
- Methods of Backtesting
- Types of Backtesting algorithm
- Backtest analysis
- Backtest of Time-series Momentum
- Backtest of Pairs-trading

Types of Backtest algorithms

Vectorized

- Vectorized the backtesting loop using Numpy/Pandas
- Fastest
- Not a reliable backtest. Is used as a sanity check.
- Easy to implement.

For Loop

- The backtest is executed using a for loop.
- Slower than vectorized.
- Not a reliable backtest. Is used as a sanity check.
- Easy to implement.

Event Driven

- Uses market simulation and runs the backtest as events takes place.
- Slowest
- The most reliable backtest.
- Difficult to implement.

Outline

- Why Backtest ?
- Methods of Backtesting
- Types of Backtesting algorithm
- Backtest analysis
- Backtest of Time-series Momentum
- Backtest of Pairs-trading

Risk and Performance Metrics (1)

- **Cumulative Returns**

$$R_c = \prod_{t=0}^{n}(1 + r_t)$$

where

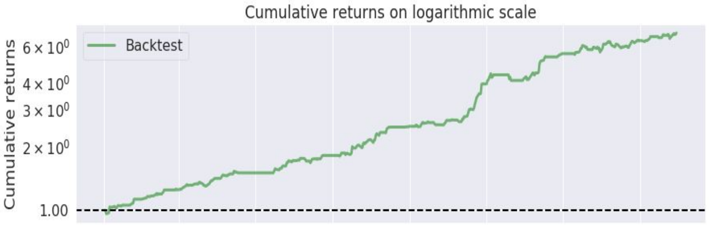
$$r_t = \text{return at } t$$
$$n = \text{number of observations}$$

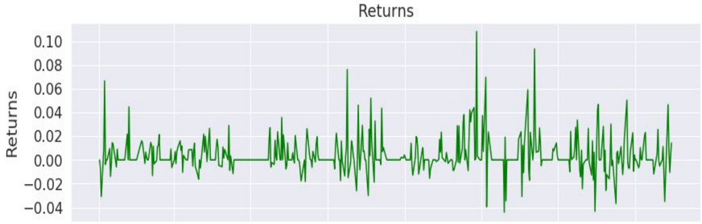
- **Annualized Return**

$$R_A = R_c^{\frac{m}{n}} - 1$$

where

$$R_c = \text{Cumulative returns}$$
$$m = \text{periods per year}$$
$$n = \text{number of observations}$$





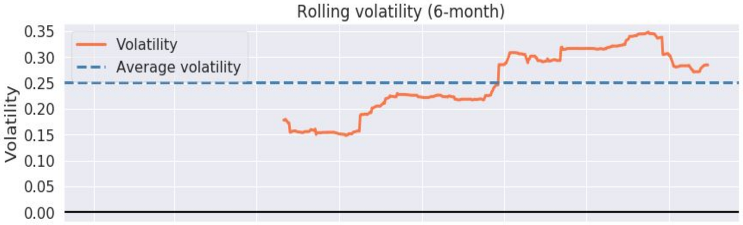
Risk and Performance Metrics (2)

- **Annualized Volatility**

$$\sigma_A = \sigma_r \sqrt{m}$$

where

$$\sigma_r = \text{std. dev. of returns}$$
$$m = \text{periods per year}$$

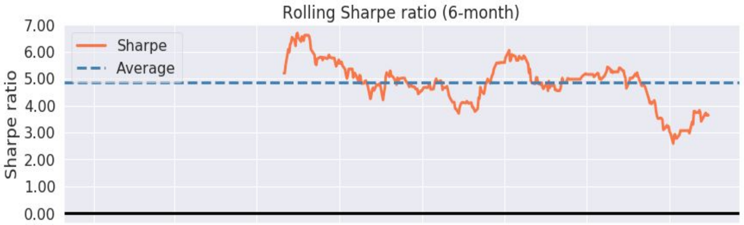


- **Annualized Sharpe Ratio**

$$SR = \frac{R_A - R_f}{\sigma_A}$$

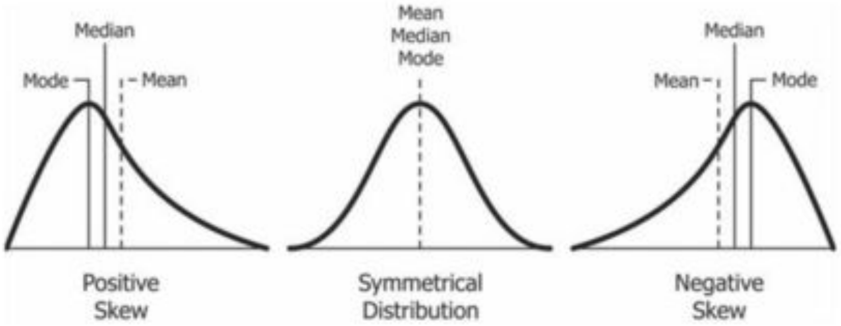
where

$$R_A = \text{Annualized return}$$
$$R_f = \text{risk-free rate}$$
$$\sigma_A = \text{Annualized volatility}$$



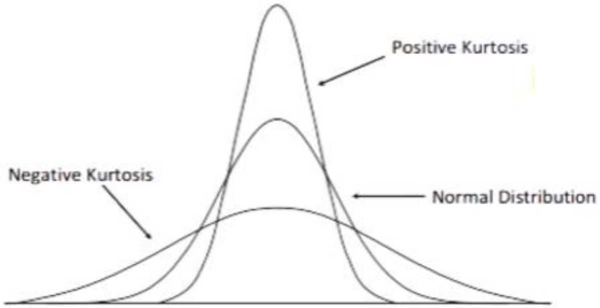
Risk and Performance Metrics (3)

- **Skewness**



Source: wikipedia

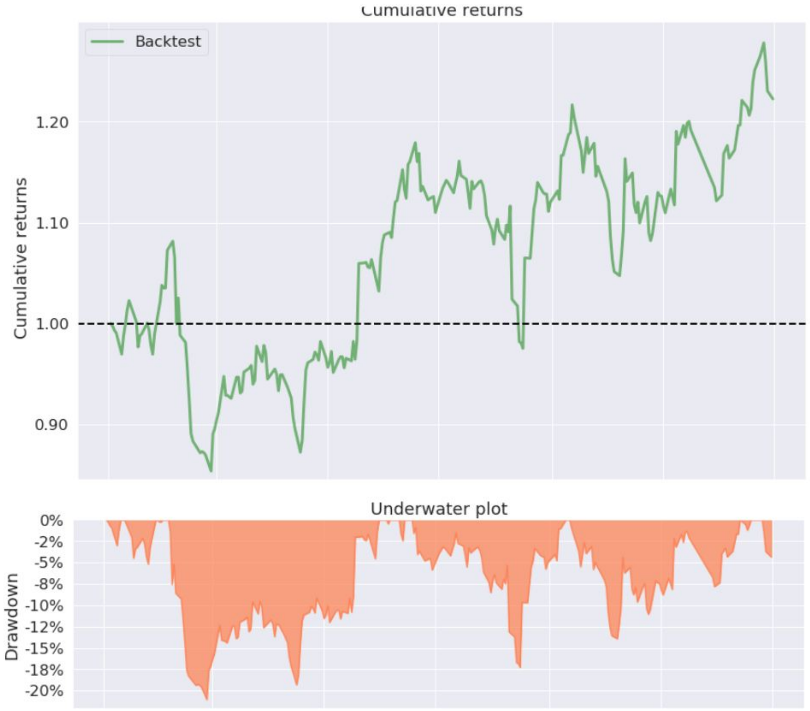
- **Kurtosis**



Risk and Performance Metrics (4)

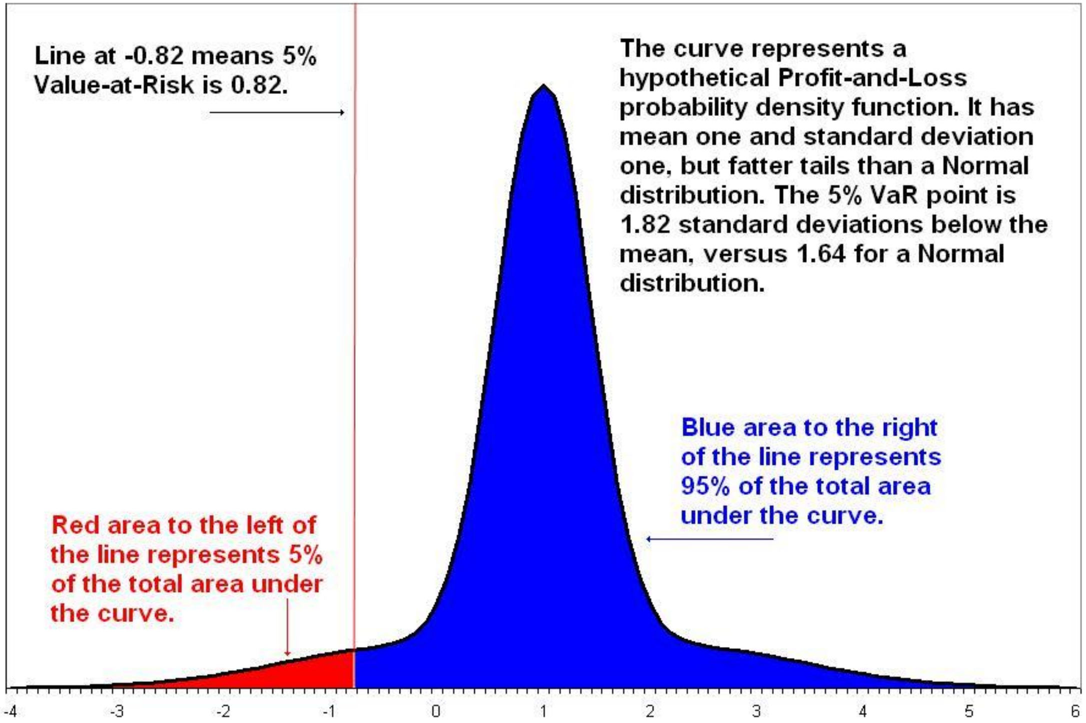
- **Maximum Drawdown**

$$MDD = \frac{\text{Trough Value} - \text{Peak Value}}{\text{Peak Value}}$$



Risk and Performance Metrics (5)

- **VaR (Value at Risk)**



Source: wikipedia

Backtest : Time-series Momentum

Notebook Link: https://bit.ly/3xOvxN8

Backtest : Pairs-Trading

Notebook Link: https://bit.ly/3v7KWWK

References

- Michael Halls-Moore. *Successful Algorithmic Trading*

- Yves Hilpisch. *Python for Finance*

- Chan, 2013. Algorithmic Trading: Winning Strategies and Their Rationale

- Lopez de Prado, 2018. Advances in Financial Machine Learning

- Luo et al. [2014]. “Seven Sins of Quantitative Investing”

- Arnott, Robert D. and Harvey, Campbell R. and Markowitz, Harry, A Backtesting Protocol in the Era of Machine Learning (November 21, 2018). Available at SSRN: https://ssrn.com/abstract=3275654 or http://dx.doi.org/10.2139/ssrn.3275654

- Bailey, David H. and Borwein, Jonathan and Borwein, Jonathan and López de Prado, Marcos and López de Prado, Marcos and Zhu, Qiji Jim, Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-of-Sample Performance (April 1, 2014). Notices of the American Mathematical Society, 61(5), May 2014, pp.458-471, Available at SSRN: https://ssrn.com/abstract=2308659 or http://dx.doi.org/10.2139/ssrn.2308659

QNA

Task for Day 2

Backtest: Time-series Momentum

- Backtest the strategy discussed in the video : https://www.youtube.com/watch?v=D_WhInJePC8

- Perform the backtest on a stock of your choice and collect the historical data using Alpaca Trading API.

- Vectorized backtest is preferable but for-loop can also be used.

- Calculate the risk and performance metrics.

- Plot cumulative returns against a benchmark, drawdown and rolling Sharpe ratio.

- Extra points: Include transaction cost.

- Use Google Colab

- Last date for submission : 05-05-2022

- Submission form link : https://forms.gle/JgrH7U7sdoSMVLUU8