Become a Quant - Data Science and Trading

Chandini A Jain
Auquan

January 29, 2016

What is Auquan?

www.auquan.com

Create the next generation of quants by making algorithmic trading skills accessible to everyone

We want to engage skilled data scientists, physicists, engineers, developers, college students – anyone – to apply the modeling techniques used in their fields to write investment algorithms.

facebook.com/tradewithauquan

twitter.com/tradewithauquan

QUANT QUEST

www.quant-quest.auquan.com

Agenda

- What is Trading/Algorithmic Trading?
- How does a trade happen? How do algorithms fit in?
- What does a quant do?
- How to build a trading strategy?
    - Simple Mean Reversion
    - Mean Reversion with Bollinger Bands
- What is Auquan?
- What's in it for me?

What is Algorithmic Trading?

Quantitative trading involves creating “end-to-end computational trading systems that use statistical rules to trade financial assets in a fully automated manner”

Fundamental assumptions

- The market is not efficient
- What happened repeatedly in the past might very well happen again

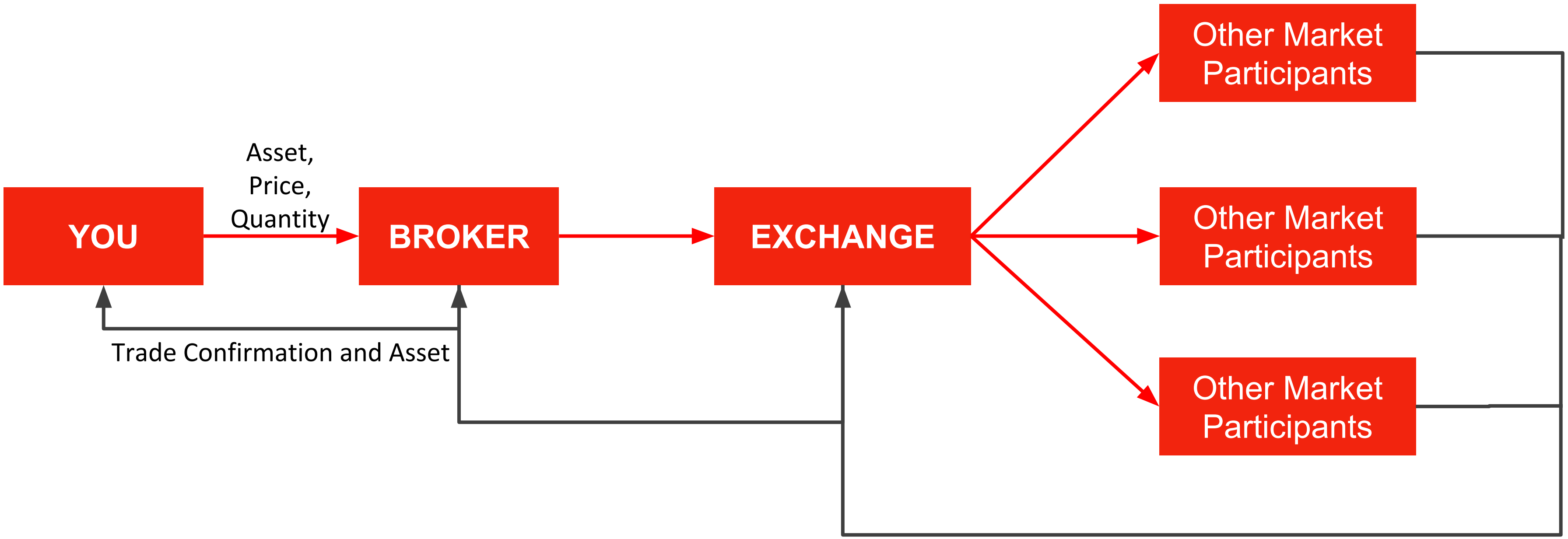
The "Turtle" trading story…

In 1983, legendary commodity trader Richard Dennis had a debate to settle with fellow trader William Eckhardt. He believed that anyone could be taught to trade successfully while Eckhardt felt he had a gift that allowed him to trade profitably.

Dennis recruited 23 people in two groups (known as "*turtles*", he said he could grow traders as quickly and efficiently as farm-grown turtles) in 1983 and 1984 and trained them for only two weeks about a simple trend-following system. After the training ended, he gave each Turtle a trading account and had them trade the systems they had been taught. Dennis believed so strongly in his experiment that he actually gave the traders his own money to trade, ranging from $250,000 to $2 million.

When his experiment ended five years later, the two classes of turtles Dennis personally trained had earned an aggregate profit of **$175 million!**

How does a trade happen?


Other Market Participants
Asset,
Price,
Quantity
YOU
BROKER
EXCHANGE
Other Market Participants
Trade Confirmation and Asset
Other Market Participants

How do algorithms fit in?

- Traditional discretionary trading relies on the experience of trader
- Algorithms are used to design a scientific approach to buying and selling financial instruments
- Make use of quantitative tools such as statistics, time series analysis and machine learning to implement a strategy to buy and sell
- Ability to provide expectations about future performance based on past performance due to availability of abundant historical data
- Less prone to error - not subject to human bias and no rule breaking

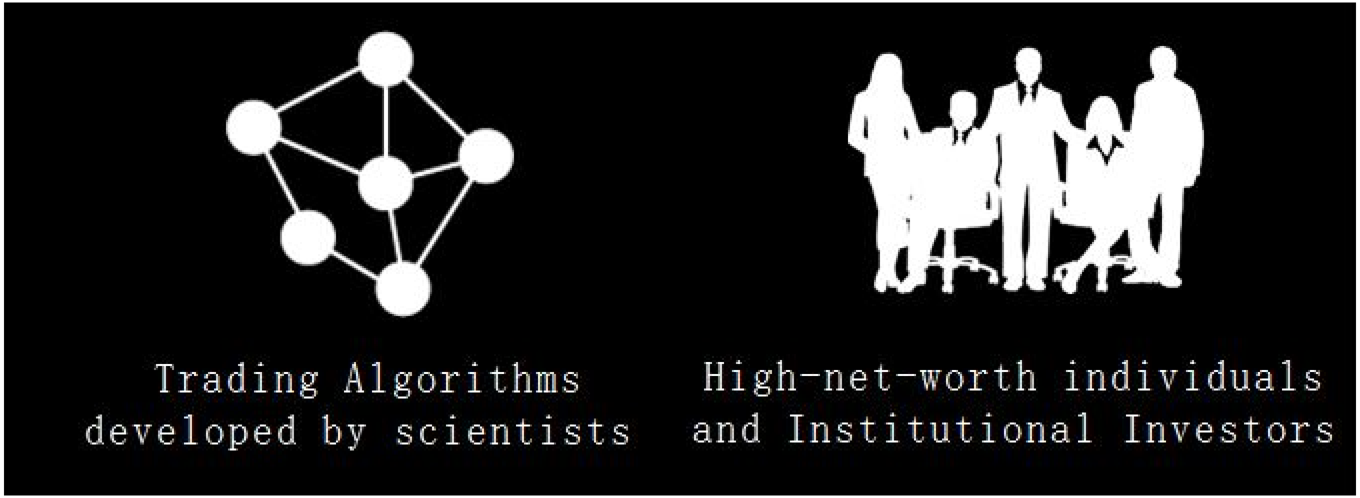
The ultimate challenge for data scientists

What does a quant do?

- **Quant:** Scan through years and years of market data to uncover patterns to forecast future prices and anticipate similar events that might reoccur in the future
- **Data scientist:** Use an algorithm to uncover patterns that are likely to reoccur in the future.

A quant is a data scientist applying his skills to financial data sets!

The process of using machine learning for a data scientist and creating a strategy for a trader are very similar - Use data science to build a strategy that will perform well on new data


Trading Algorithms developed by scientists
High-net-worth individuals and Institutional Investors

Quantitative Funds

$300 Billion Industry

Why should I be a quant?

Data Scientists Making Millions Vie With Fund Managers Over Pay

Portfolio managers at hedge funds have another thing to worry about: the $2 million data scientist.

BLOOMBERG.COM

"Experienced data scientists can earn $500k -$700k as a quant at a hedge fund, and as much as three times that for those with extensive backgrounds"

Why should I be a quant?

- Renaissance Technologies- $65 billion AUM
  - Founder: James Simons - $14 billion
- DE Shaw - $39 billion AUM
  - Founder: David Shaw - $4.7 billion
- Two Sigma - $32 billion AUM
  - Founder: John Overdeck and David Siegel- $2.8 billion each

Torch bearers of the growing tribe of quants who use data and machine learning in an attempt to beat the market consistently

How to build a Trading System?

Developing a trading algorithm system consists of several elements

- □**Identify a strategy** - Decide which markets you want to trade, develop the logic with which you want to trade them and define parameters by which the trading logic is triggered
- □**Backtest your strategy** - Analyze your strategy’s performance on historical data and remove biases
- □**Execute your strategy** - Link to a brokerage and minimize the transaction costs (cost of trading and commission)
- □**Manage Risk** - Create pre and post trade checks to avoid losses

Developing a profitable trading strategy is more than just finding a good idea

Identifying a strategy

- Which markets to trade? Stocks, debt, commodities, FX? Stocks are a good place to start for beginners
- Which strategy to use to make decisions? Finding a strategy is not hard, can use strategies published in books, blogs. Many of the strategies you will look at will fall into the categories of mean-reversion and trend-following
- Optimize the strategy to be consistently profitable on new, unseen data

Optimization methods key to a profitable strategy

Backtesting a strategy

Backtesting provides evidence that strategy is profitable with historical data → expectation of how the strategy will perform

Backtesting is NOT a guarantee of success

Key factors:

- availability and cleanliness of historical data
- identifying and avoiding any bias
- quantifying system performance

Executing a Strategy

Execution system -means by which trades are sent to broker for execution

- You want to buy AAPL shares, broker will communicate with exchange and find a seller, buy them on your behalf and charge a fee
- Interfacing with brokerage - calling a broker to fully-automated API
- Discount brokers such as Interactive Brokers or Zerodha offer an online trading platform where clients can submit trades for execution

We need a good execution system to minimize transaction costs

Risk Management

Once you have your trading and execution system in place, the final part is to setup a risk management system

- Risk is anything that can go wrong - faulty algorithms, defective implementation, system malfunction, personal biases, technology risk, brokerage risk
- Risk management includes pre and post trade checks in algorithm, assigning reasonable stop losses, optimal capital allocation etc
- Effective risk management encompasses creating broad protection against catastrophic losses without compromising execution speed and profitability

Where to start?

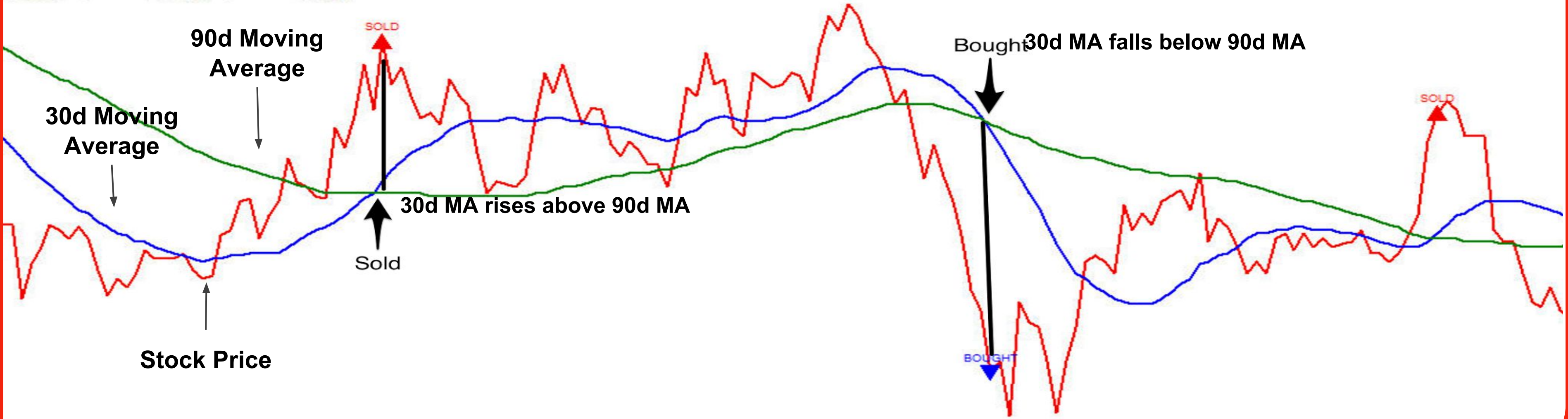
- Start Simple, don‘t be intimidated by the problem
- Apply methods you already know on financial data
- Visualize and optimize
- Be wary of overfitting

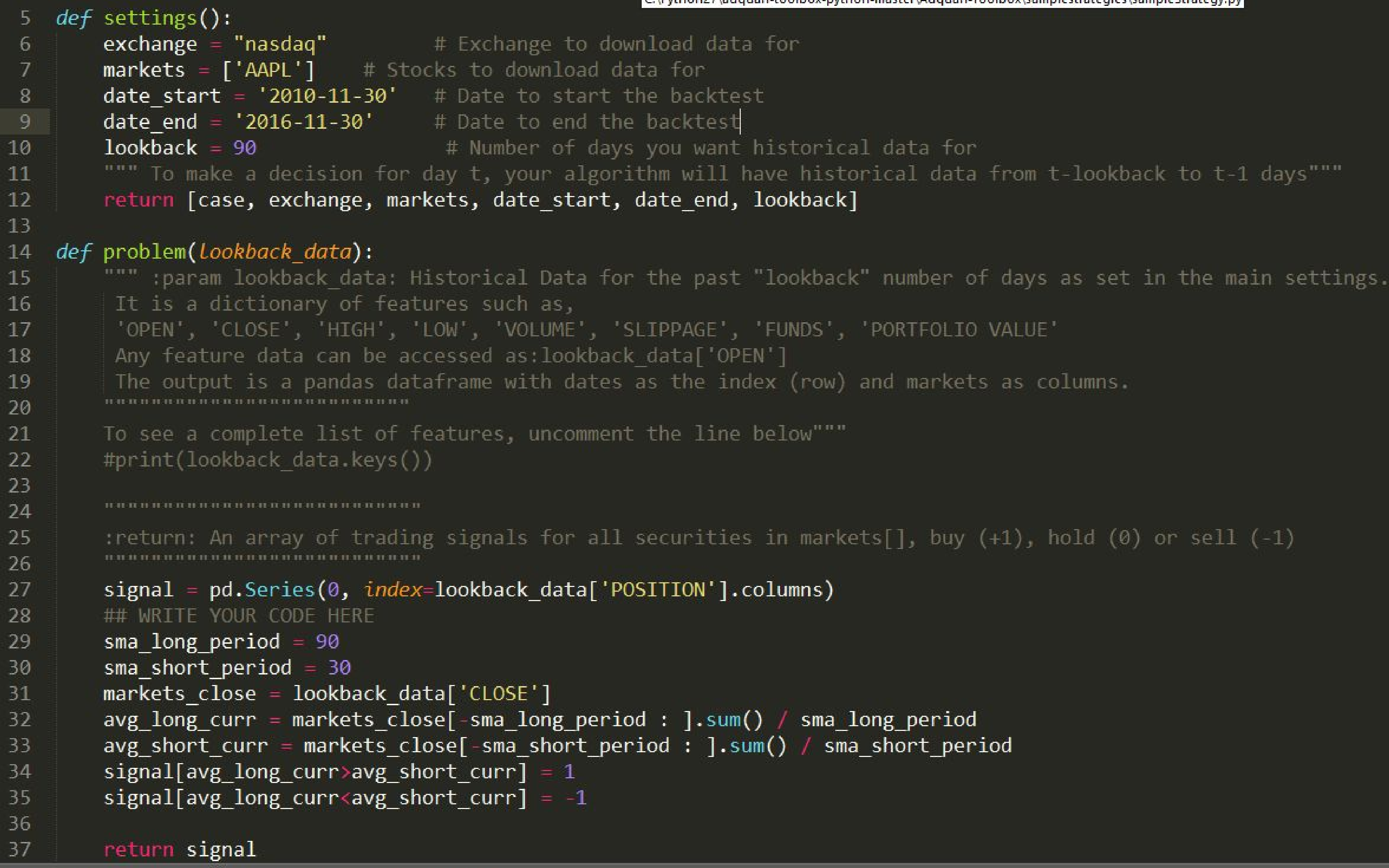
Finding a good strategy is both science and an art!

Sample Strategy: Mean Reversion

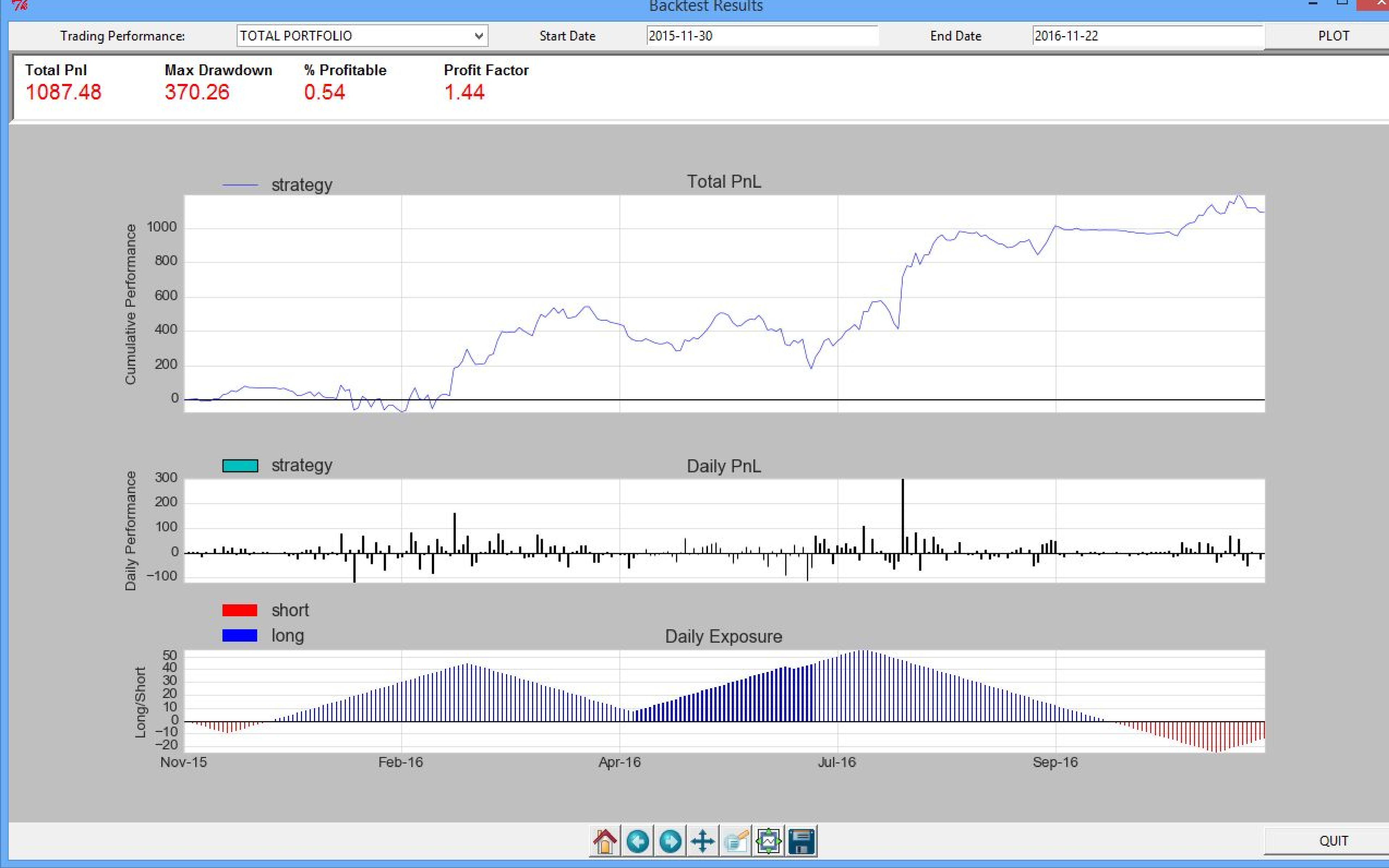
Mean Reversion Idea:

- If the value of 30d MA falls below 90d MA, the current price is too low and likely to increase. Hence this is a signal to buy
- If the value of 30d MA rises above 90d MA, the current price is too high and is a signal to sell.





```
def settings():
    exchange = "nasdaq"         # Exchange to download data for
    markets = ['AAPL']    # Stocks to download data for
    date_start = '2010-11-30'   # Date to start the backtest
    date_end = '2016-11-30'     # Date to end the backtest
    lookback = 90                # Number of days you want historical data for
    """ To make a decision for day t, your algorithm will have historical data from t-lookback to t-1 days"""
    return [case, exchange, markets, date_start, date_end, lookback]

def problem(lookback_data):
    """ :param lookback_data: Historical Data for the past "lookback" number of days as set in the main settings.
     It is a dictionary of features such as,
     'OPEN', 'CLOSE', 'HIGH', 'LOW', 'VOLUME', 'SLIPPAGE', 'FUNDS', 'PORTFOLIO VALUE'
     Any feature data can be accessed as:lookback_data['OPEN']
     The output is a pandas dataframe with dates as the index (row) and markets as columns.
    """"""""""""""""""""""""""

    To see a complete list of features, uncomment the line below"""
    #print(lookback_data.keys())

    """""""""""""""""""""""""""
    :return: An array of trading signals for all securities in markets[], buy (+1), hold (0) or sell (-1)
    """""""""""""""""""""""""""
    signal = pd.Series(0, index=lookback_data['POSITION'].columns)
    ## WRITE YOUR CODE HERE
    sma_long_period = 90
    sma_short_period = 30
    markets_close = lookback_data['CLOSE']
    avg_long_curr = markets_close[-sma_long_period : ].sum() / sma_long_period
    avg_short_curr = markets_close[-sma_short_period : ].sum() / sma_short_period
    signal[avg_long_curr>avg_short_curr] = 1
    signal[avg_long_curr<avg_short_curr] = -1

    return signal
```



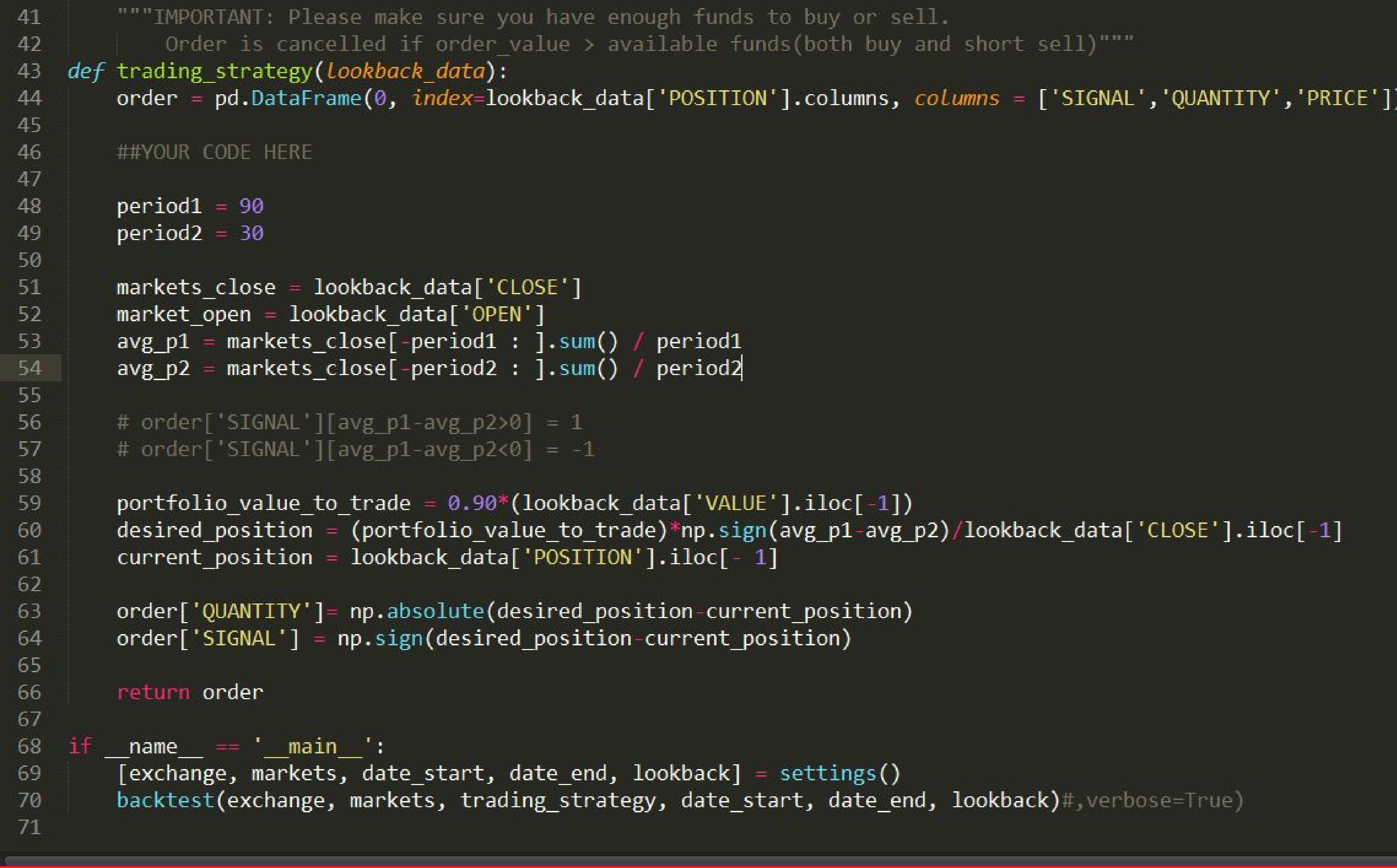
Backtest Results

Trading Performance: TOTAL PORTFOLIO

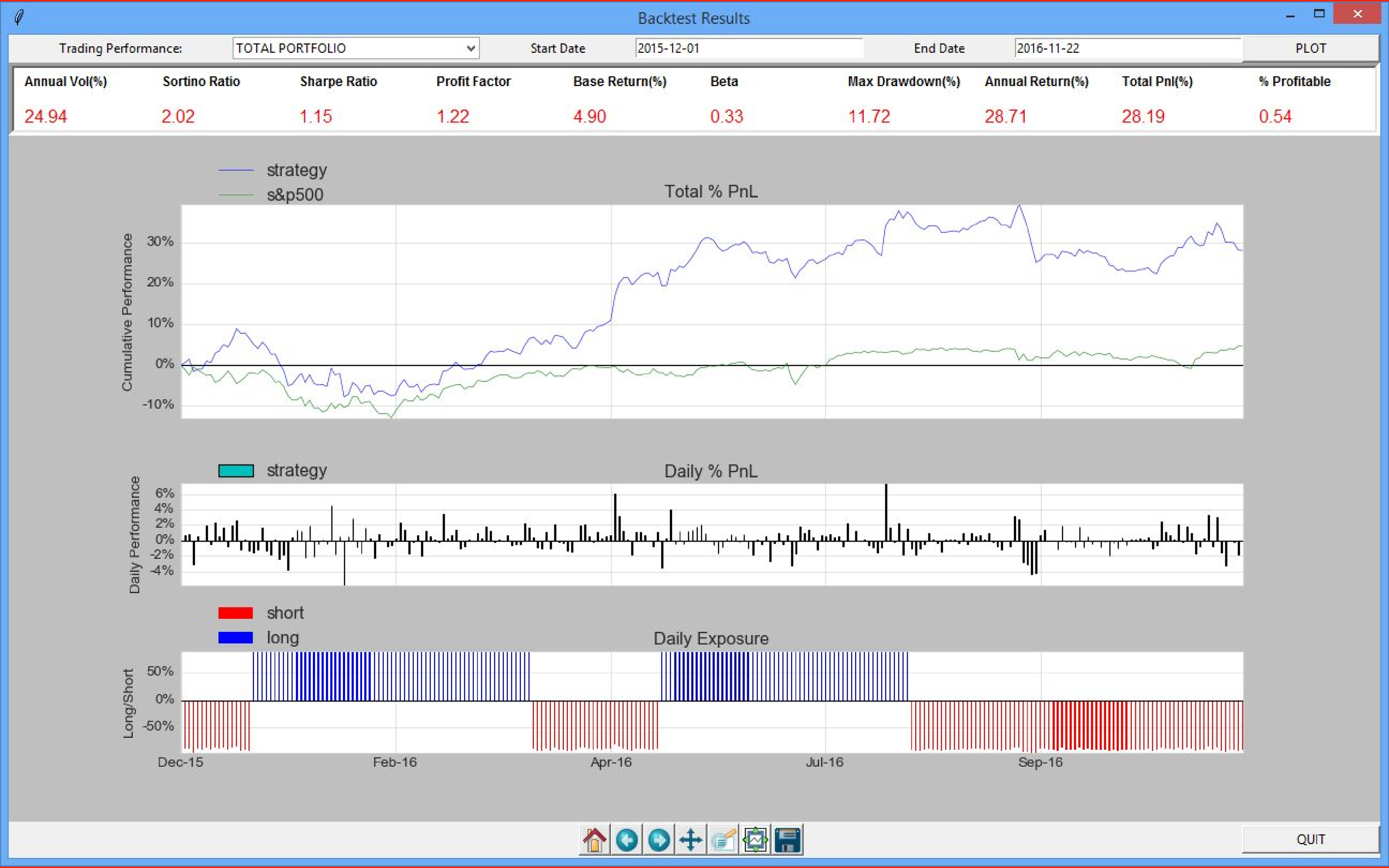
Start Date 2015-11-30

End Date 2016-11-22

Total Pnl	Max Drawdown	% Profitable	Profit Factor
1087.48	370.26	0.54	1.44



```
    """IMPORTANT: Please make sure you have enough funds to buy or sell.
        Order is cancelled if order_value > available funds(both buy and short sell)"""
def trading_strategy(lookback_data):
    order = pd.DataFrame(0, index=lookback_data['POSITION'].columns, columns = ['SIGNAL','QUANTITY','PRICE']

    ##YOUR CODE HERE

    period1 = 90
    period2 = 30

    markets_close = lookback_data['CLOSE']
    market_open = lookback_data['OPEN']
    avg_p1 = markets_close[-period1 : ].sum() / period1
    avg_p2 = markets_close[-period2 : ].sum() / period2

    # order['SIGNAL'][avg_p1-avg_p2>0] = 1
    # order['SIGNAL'][avg_p1-avg_p2<0] = -1

    portfolio_value_to_trade = 0.90*(lookback_data['VALUE'].iloc[-1])
    desired_position = (portfolio_value_to_trade)*np.sign(avg_p1-avg_p2)/lookback_data['CLOSE'].iloc[-1]
    current_position = lookback_data['POSITION'].iloc[- 1]

    order['QUANTITY']= np.absolute(desired_position-current_position)
    order['SIGNAL'] = np.sign(desired_position-current_position)

    return order

if __name__ == '__main__':
    [exchange, markets, date_start, date_end, lookback] = settings()
    backtest(exchange, markets, trading_strategy, date_start, date_end, lookback)#,verbose=True)

```



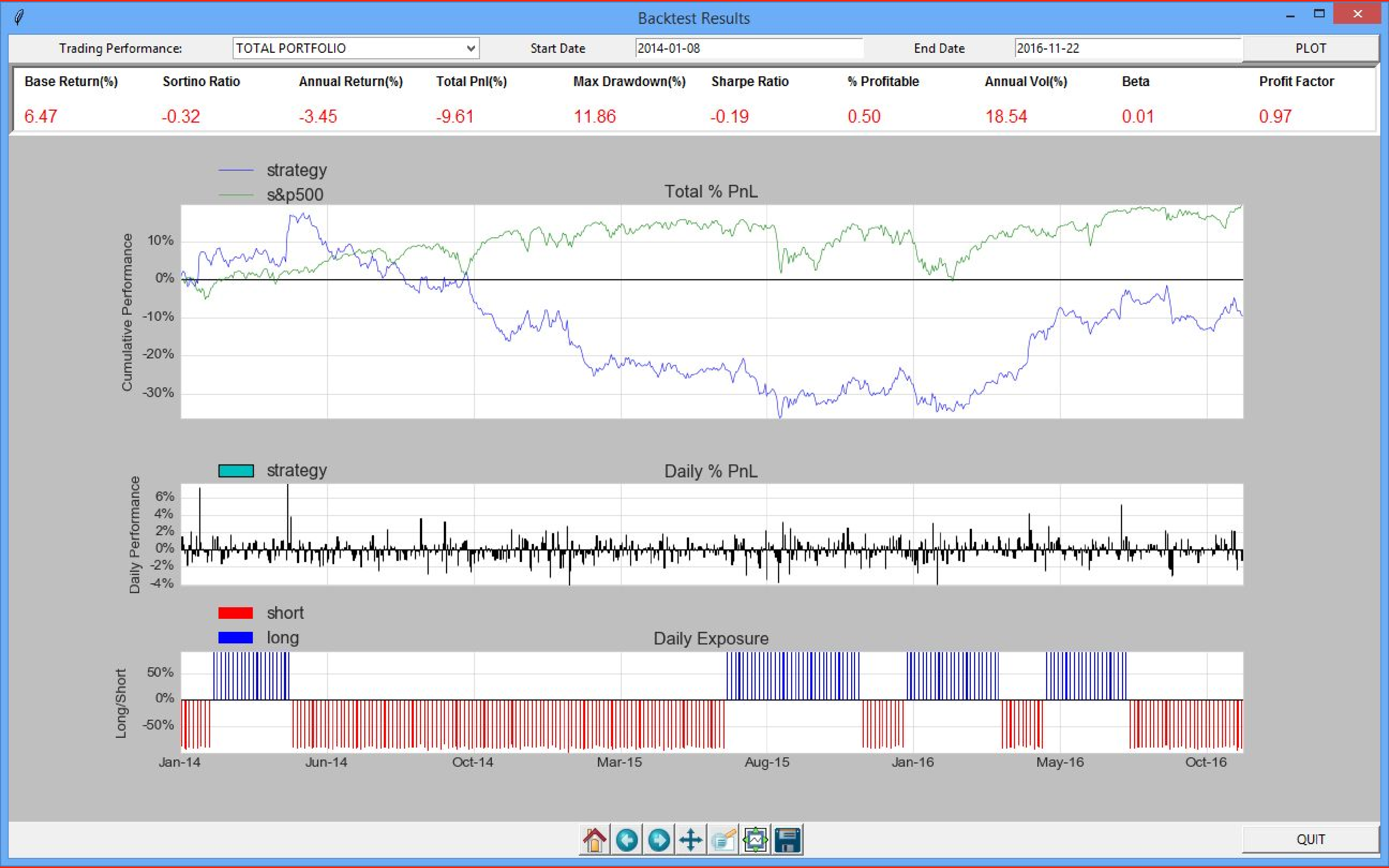
Backtest Results

Trading Performance: TOTAL PORTFOLIO

Start Date: 2015-12-01

End Date: 2016-11-22

Annual Vol(%)	Sortino Ratio	Sharpe Ratio	Profit Factor	Base Return(%)	Beta	Max Drawdown(%)	Annual Return(%)	Total Pnl(%)	% Profitable
24.94	2.02	1.15	1.22	4.90	0.33	11.72	28.71	28.19	0.54



Backtest Results

Trading Performance: TOTAL PORTFOLIO

Start Date 2014-01-08

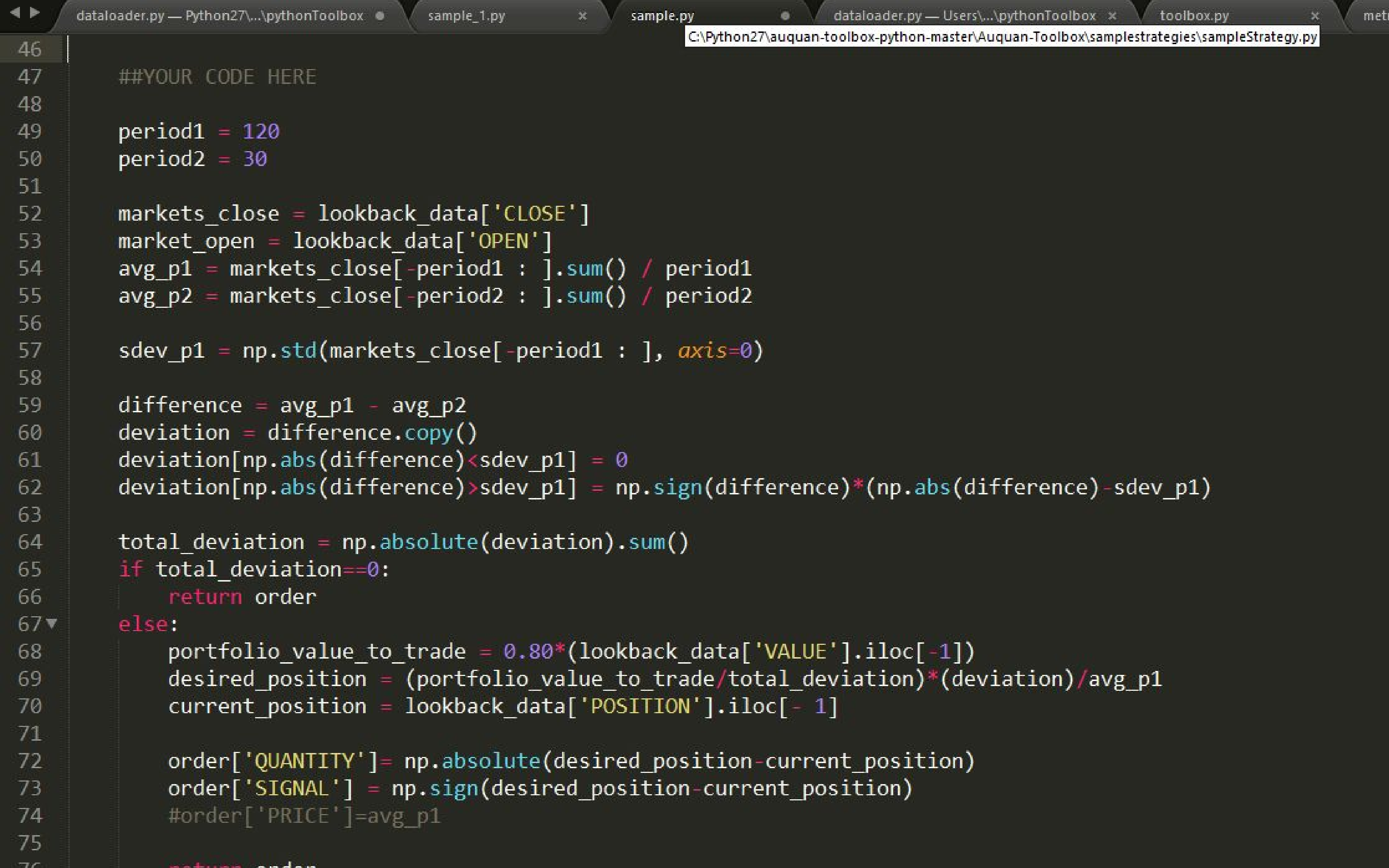
End Date 2016-11-22

Base Return(%)	Sortino Ratio	Annual Return(%)	Total Pnl(%)	Max Drawdown(%)	Sharpe Ratio	% Profitable	Annual Vol(%)	Beta	Profit Factor
6.47	-0.32	-3.45	-9.61	11.86	-0.19	0.50	18.54	0.01	0.97

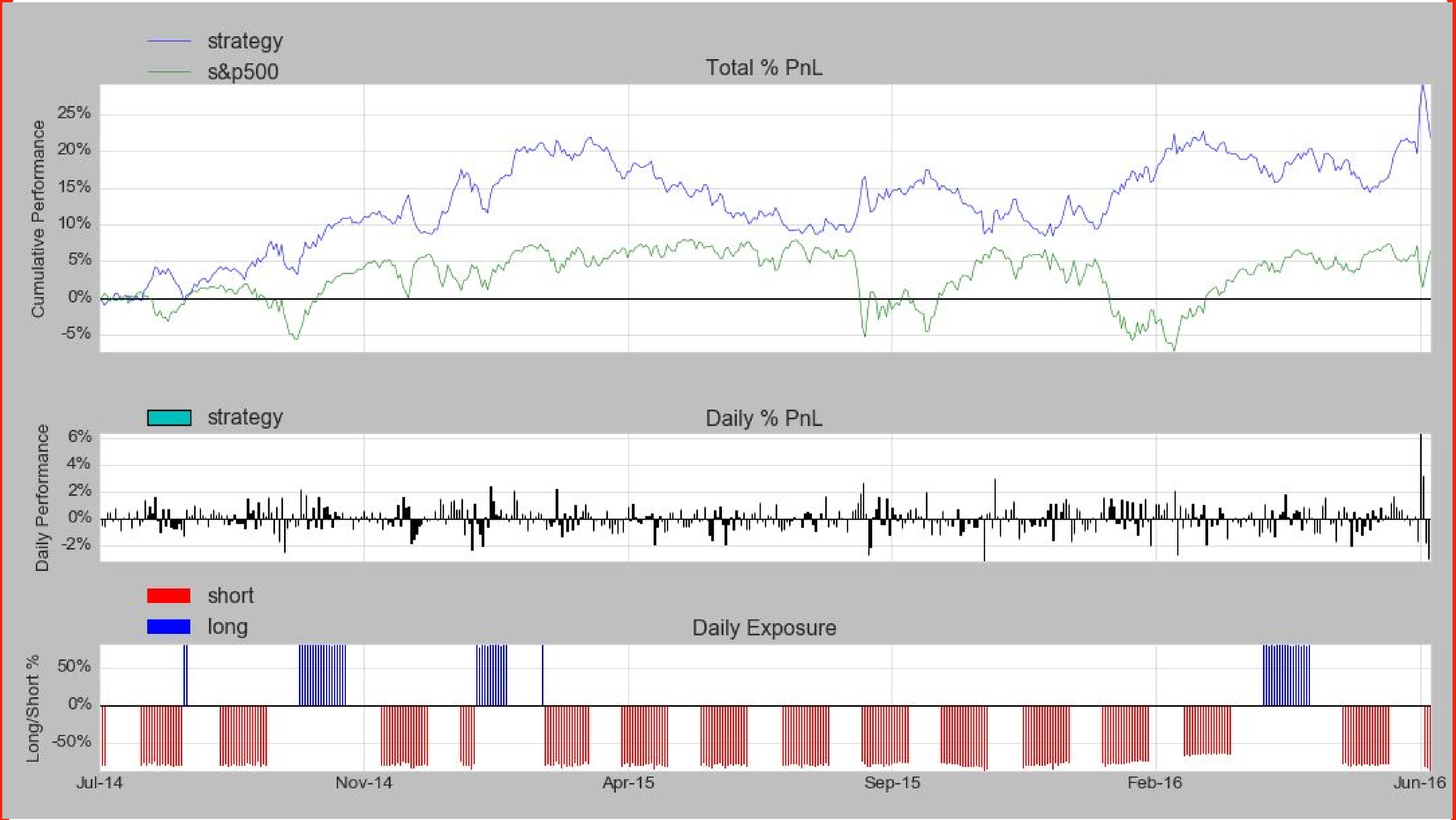
Mean Reversion: Bollinger Band

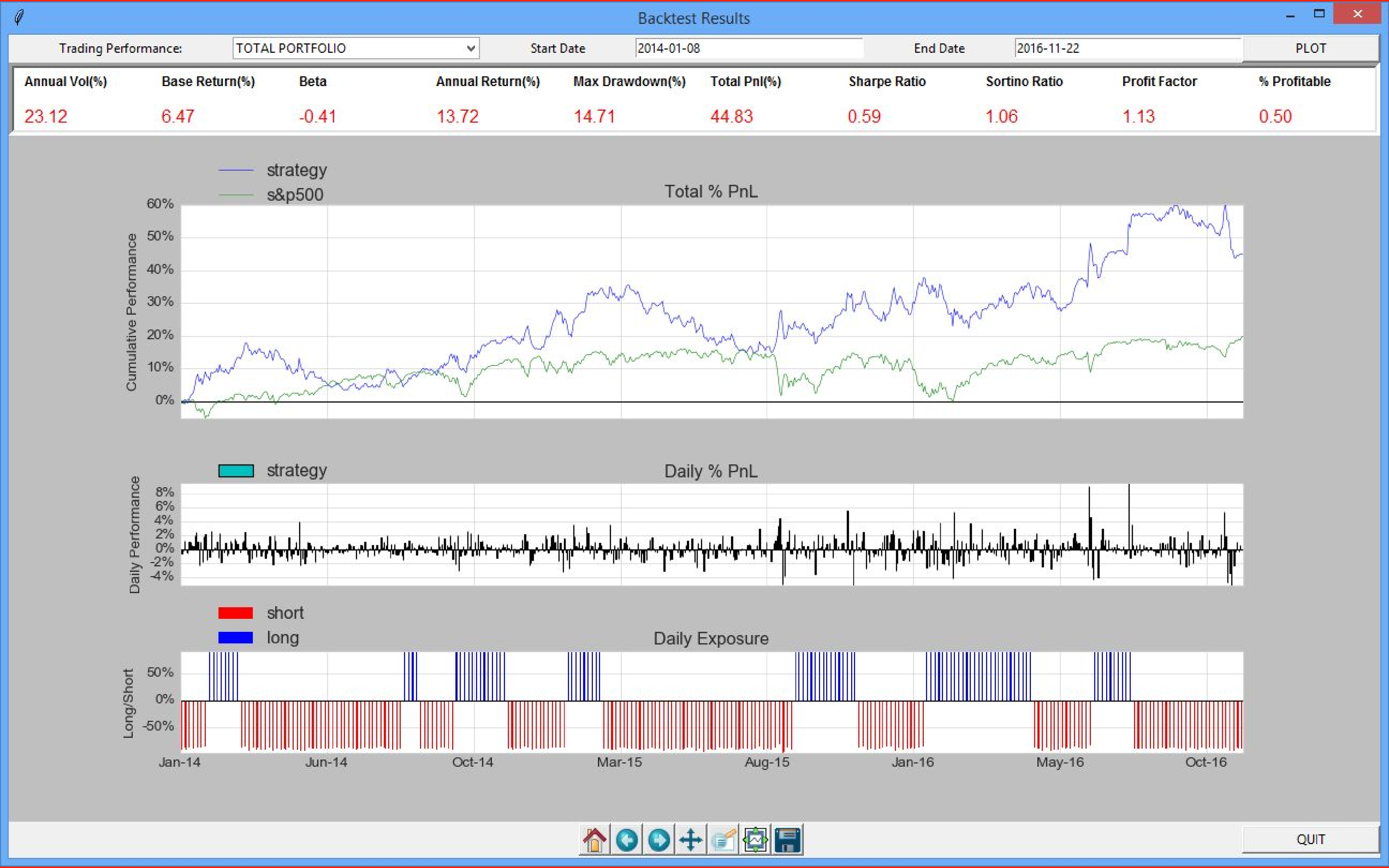
Bollinger Bands idea:

- If the value of 30d MA falls below 90d MA by more than **one standard deviation**, the current price is too low and likely to increase. Hence this is a signal to buy
- If the value of 30d MA rises above 90d MA by more than **one standard deviation**, the current price is too high and is a signal to sell.


dataloader.py — Python27\...\pythonToolbox
sample_1.py
sample.py
dataloader.py — Users\...\pythonToolbox
toolbox.py
metr
C:\Python27\auquan-toolbox-python-master\Auquan-Toolbox\samplestrategies\sampleStrategy.py

```python

    ##YOUR CODE HERE

    period1 = 120
    period2 = 30

    markets_close = lookback_data['CLOSE']
    market_open = lookback_data['OPEN']
    avg_p1 = markets_close[-period1 : ].sum() / period1
    avg_p2 = markets_close[-period2 : ].sum() / period2

    sdev_p1 = np.std(markets_close[-period1 : ], axis=0)

    difference = avg_p1 - avg_p2
    deviation = difference.copy()
    deviation[np.abs(difference)<sdev_p1] = 0
    deviation[np.abs(difference)>sdev_p1] = np.sign(difference)*(np.abs(difference)-sdev_p1)

    total_deviation = np.absolute(deviation).sum()
    if total_deviation==0:
        return order
    else:
        portfolio_value_to_trade = 0.80*(lookback_data['VALUE'].iloc[-1])
        desired_position = (portfolio_value_to_trade/total_deviation)*(deviation)/avg_p1
        current_position = lookback_data['POSITION'].iloc[- 1]

        order['QUANTITY']= np.absolute(desired_position-current_position)
        order['SIGNAL'] = np.sign(desired_position-current_position)
        #order['PRICE']=avg_p1

        return order
```


strategy
s&p500
Total % PnL
Cumulative Performance
25%
20%
15%
10%
5%
0%
-5%
strategy
Daily % PnL
Daily Performance
6%
4%
2%
0%
-2%
short
long
Daily Exposure
Long/Short %
50%
0%
-50%
Jul-14
Nov-14
Apr-15
Sep-15
Feb-16
Jun-16



Backtest Results

Trading Performance: TOTAL PORTFOLIO

Start Date 2014-01-08

End Date 2016-11-22

Annual Vol(%)	Base Return(%)	Beta	Annual Return(%)	Max Drawdown(%)	Total Pnl(%)	Sharpe Ratio	Sortino Ratio	Profit Factor	% Profitable
23.12	6.47	-0.41	13.72	14.71	44.83	0.59	1.06	1.13	0.50

RSI Indicator

$$closeMom(t) = CLOSE(t) - CLOSE(t-1)$$

$$up(t) = \begin{cases} 1 & \dots \quad if\ closeMom(t) \geq 0 \\ 0 & \dots \quad otherwise \end{cases}$$

$$down(t) = \begin{cases} 1 & \dots \quad if\ closeMom(t) < 0 \\ 0 & \dots \quad otherwise \end{cases}$$

$$meanUp(t, period) = \frac{1}{period} \sum_{t}^{i=t-period+1} up(i)$$

$$meanDown(t, period) = \frac{1}{period} \sum_{t}^{i=t-period+1} down(i)$$

$$RSI(t, period) = 100 - \frac{100}{1 + \frac{meanUp(t, period)}{meanDown(t, period)}}$$

$t \dots index\ of\ the\ trading\ day \qquad period \dots number\ of\ days\ to\ compute\ the\ RSI$

Average True Range

$$TR(t) = \max(HIGH(t) - LOW(t), |HIGH(t) - CLOSE(t-1)|, |LOW(t) - CLOSE(t-1)|)$$

$$ATR(t, period) = \frac{1}{period} \sum_{t}^{i=t-period+1} TR(i)$$

$t \ldots index\ of\ the\ trading\ day \qquad period \ldots number\ of\ days\ to\ compute\ the\ ATR$

$$VolaRatio(t, period) = \frac{ATR(t, period)}{CLOSE(t)}$$

How to evaluate performance?

Multiple metrics to consider:

- Performance (Net Profit/Loss)
- Volatility (Standard Deviation of Returns)
- Sharpe Ratio - Returns/Risk
- Drawdowns
- Correlations
- Profit Factor (Total Profit/Total Loss)
- % Profitability

What is Auquan?

uquan

www.auquan.com

Create the next generation of quants by making algorithmic trading skills accessible to everyone

We want to engage skilled data scientists, physicists, engineers, developers, college students – anyone – to apply the modeling techniques used in their fields to write investment algorithms.

facebook.com/tradewithauquan

twitter.com/tradewithauquan

Auquan

QUANT QUEST

PARTNERED BY

optiver

30 JANUARY - 5 FEBRUARY 2017

ARE YOU AN EXPERT CODER,
PASSIONATE ABOUT MATHS,
AND STRONG ON DATA?
PARTICIPATE IN OUR QUANTS CHALLENGE AND
STAND A CHANCE TO GET HIRED BY
OPTIVER

www.quant-quest.auquan.com

Questions/Feedback?

Reach us at info@auquan.com