‹ Return to "AI for Trading" in the classroom

DISCUSS ON STUDENT HUB ›

Breakout Strategy

REVIEW | CODE REVIEW | HISTORY

Meets Specifications

Dear Student,

Well done passing all the specifications for the project, you have now successfully implemented a Breakout Strategy using a Notebook. The following are a few resources for continued learning on the topic:
Here is a blog with more information on high low breakout strategy
Here is some more information on how to determine if histograms are normal or not and what causes the skewness.
Here is some more info in ks test You can learn to apply lambda functions from here.
Here is some more info on breakout strategy
Here are some good reads:
List of code, papers, and resources for AI/deep learning/machine learning/neural networks applied to algorithmic trading
Machine Trading: Deploying Computer Algorithms to Conquer The Markets
Quantitative Trading: How to Build Your Own Algorithmic Trading Business

Rate this review ☆☆☆☆☆

The project meets all specifications, well done on completing the project.

Thanks!

Generate Signal

✓ The function `get_high_lows_lookback` computes the maximum and minimum of the closing prices over a window of days.

✓ The function `get_long_short` computes long and short signals using a breakout strategy.

✓ The function `filter_signals` filters out repeated long or short signals.

✓ The function `get_lookahead_prices` gets the close price days ahead in time.

✓ The function `get_return_lookahead` generates the log price return between the closing price and the lookahead price.

✓ The function `get_signal_return` generates the signal returns.

Evaluate Signal

✓ Correctly answers the question “What do the histograms tell you about the signal returns?"

Outliers

✓ The function `calculate_kstest` calculates the ks and p values.

`calculate_kstest` has been implemented correctly 👍

```
Tests Passed
```

Congratulations!

✓ The function `find_outliers` returns the list of outlying symbols.

The function find_outliers implemented correctly to return the list of outlying symbols.

```
Tests Passed
```

Congratulations!
`find_outliers` function correctly returns 24 outliers 👍
`find_outliers` can also be implemented in one line using `intersection` as follows:
`return set(ks_values[ks_values > ks_threshold].index).intersection(p_values[p_values < pvalue_threshold].index)`

DOWNLOAD PROJECT

RETURN TO PATH