UDACITY

‹ Return to Classroom

DISCUSS ON STUDENT HUB

Breakout Strategy

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Hello Udacity Student,
You have a good submission👏! I saw your determination in completing this project which is such an admirable trait. Congratulations👏 on successful completion of this project and. Thanks for putting in this much effort, you showed great understanding of the course which is good. Keep it up and continue working hard. Have a nice day.

> I could not wrap my head around the idea of calculating look-ahead prices/log returns then use them to find signal returns. Could you share with me something I could read up?

Look-Ahead Bias And How To Detect It

Suggestions

For more insights on the project, you can visit the links below:
The Anatomy of Trading Breakouts
Momentum Day Trading Strategies for Beginners: A Step by Step Guide
Day Trading Breakouts – 4 Simple Trading Strategies

Generate Signal

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

Well done! The implementation `get_high_lows_lookback` is perfectly implemented.
Please check the following reference to know more about `shift` and `rolling` functions:

- What is meant by shift in dataframe
- Shifting and lagging time-series data
- Pandas DataFrame Rolling
- Using shift and rolling in pandas with groupBy

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

Excellent job in generating the long and short signals. It's nice to see that you've specified that data type using the astype function to implement `get_long_short()` correctly . Well done!👏

Suggestions

The following links may help you understand the necessary methods used in get_long_short() function:

- Pandas DataFrame astype
- Change data type of columns in Pandas
- Numpy Array types and conversions between types
- Difference between np.int, np.int_, int, and np.int_t in cython

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

Splendid work! The `filter_signals` function is nicely implemented to filter out repeated long or short signals as expected. You've used iterrows over each column as recommended.

Suggestions

You may refer on the following links below to deepen your understanding on how to access a group of rows and columns by label(s) or a boolean array using Pandas Dataframe functions:

- Pandas DataFrame .loc
- Using iloc, loc, & ix to select rows and columns in Pandas DataFrames
- Selection with .loc in python
- The difference between iloc and loc in Pandas
- Looping with iterrows

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

Good job! Function `get_lookahead_prices` to get the close price days ahead in time was correctly implemented. 👍

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

Lookahead closing prices for 5, 10, and 20 days were correctly generated using `get_return_lookahead`. It's great that Natural logarithm or np.log was used to implement this function. Well done!👍

Suggestions

To know more about natural logarithm or log, please check the links below:

- NumPy: Logarithm with base n
- numpy.log() in Python
- log(x) vs ln(x):The curse of scientific computing

The function `get_signal_return` generates the signal returns.

The function `get_signal_return` generates the signal returns. Nice work multiplying **signal** to **lookahead_returns** `signal * lookahead_returns` as seen in the implementation. 👍

Evaluate Signal

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

Nice effort with the observation. Ideally, the distributions do not look normal and that there appear to be outliers in the right tails.✔

Suggestions

Please check the resources below to gain more knowledge about frequently observed shapes of histograms and what they mean:

- Typical Histogram Shapes and What They Mean

- Common shapes of distributions
- Data Visualization in Python—Histogram in Matplotlib
- How to Analyze a Histogram
- Interpreting a Histogram
- 3 Things a Histogram Can Tell You
- How to interpret the shape of statistical data in a Histogram

Outliers

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

Very good calculation of the ks-values and p-values using the Kolmogorov-Smirnov Test in your `calculate_kstest` function. You calculated correct values for all tickers. 👍

Suggestions

- What is a Kolmogorov-Smirnov normality test?
- How to use a proper normalization to have the right p_values and ks_values from Kolmogorov-Smirnov test (KS test)?
- numpy.mean can be use to compute the arithmetic mean along the specified axis.
- numpy.std can be use to compute the standard deviation along the specified axis.
- What are the differences between np.mean and np.average?
- Summarising, Aggregating, and Grouping data in Python Pandas

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

The `find_outliers` function was performed correctly to find the outlying symbols that pass the null hypothesis with a p-value less than `pvalue_threshold` and a KS value above `ks_threshold` with the correct 24 outlying symbols obtained 👏

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review