UDACITY

Return to Classroom

Breakout Strategy

REVIEW

CODE REVIEW

HISTORY

Requires Changes

1 specification requires changes

Great job, you are almost there! 💯Clearly, you have acquired all the important concepts from this project. You only need to make some modifications and then you are ready to go. Wish you all the best for the upcoming projects!👌
Tip: For a better understanding of how to apply this strategy in trading, such as **false breakouts**, I strongly suggest that you can read this article and this article

Generate Signal

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

Well done! You successfully compute the maximum and minimum of the closing prices over a window of days.👏 `shift` and `rolling` functions are correctly applied

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

You correctly get long and short signals. 💯

- `long_short[lookback_high < close] = 1` => long signals
- `long_short[lookback_low > close] = -1` => short signals

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

Fantastic, you correctly filter out repeated long or short signals with the `clear_signals` function. 💪
Tip: Here is another way to do the job.

```
def filter_signals(signal, lookahead_days):
    pos_signal = signal[signal == 1].fillna(0)
    neg_signal = signal[signal == -1].fillna(0) * -1

    pos_signal = pos_signal.apply(lambda signals: clear_signals(signals, lookahea
d_days))
    neg_signal = neg_signal.apply(lambda signals: clear_signals(signals, lookahea
d_days))

    return pos_signal + neg_signal*-1
```

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

Excellent, you correctly get the close price days ahead in time👍 `shift` function is correctly applied

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

Good job!🎉You successfully generate the log price return. `log` function is correctly applied

The function `get_signal_return` generates the signal returns.

Good, you successfully generate the signal returns by multiplying `signal` and `lookahead_returns`.🎉

Evaluate Signal

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

Your observations are very good👍But in order to pass this rubric, you should specifically mention where the outliers could be in the histograms, such as are they close to the center or tail, are they more likely on the right, left or both sides

Outliers

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

Well done, you correctly calculate the ks and p values with `kstest` function.🎉
Tip: The reason why we are comparing the mean/std of a single stock with the ones of our portfolio is that when we look at the return-histogram, we are looking at the histogram of all the stocks in the portfolio instead of a single stock. The hypothesis we have is that we will get a normal distribution from our portfolio instead of one single stock.

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

Well done, you correctly find the list of outlying symbols with `intersection` function.👏

RESUBMIT

DOWNLOAD PROJECT

Learn the best practices for revising and resubmitting your project.

RETURN TO PATH