

Systematic Edge

Beating The Market Systematically

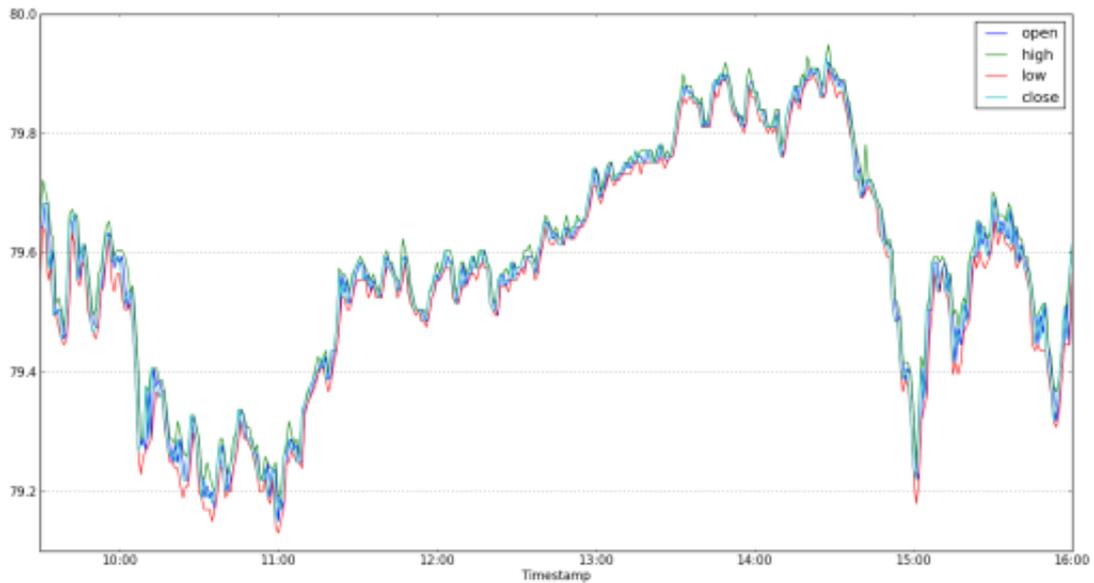
Energy Stat Arb Part 2

In my previous post, I rushed through a lot of technical details on how I implemented the strategy. For that I apologize! I am here to make up by providing more on how I approached it and hopefully make my analysis more understandable.

In this post, I want to re-visit energy pairs (XLE vs OIL) trading but with the traditional spread construction approach through regression analysis. My data comes from QuantQuote, all adjusted for dividends and splits. To read in the data, I used the following code:

```
1 from matplotlib.pyplot import *
2 import pandas as pd
3 import numpy as np
4 import datetime as dt
5 xle = pd.read_csv('/Users/mg326/xle.csv', header=None, parse_dates = [[0
6 xle.columns = ['Timestamp', 'open', 'high', 'low', 'close', 'volume', '
7 xle['Timestamp'] = xle['Timestamp'].apply(lambda x: dt.datetime.strptime
8 xle = xle.set_index('Timestamp')
9 xle = xle[["open", "high", "low", "close"]]
```

For minute data, there are approximately 391 rows of data per day. Taking in to account OHLC, there are a total of $391 * 4 = 1564$ data observations per day. Heres a image displaying May 9th, 2013:



https://systematicedge.files.wordpress.com/2014/03/figure_1.png

If you look in to the data, you may see a price for 9:45AM but the next data point comes in at 9:50AM. This means that there was a 5 minute gap where no shares were traded. To fix and align this, the following function will align the two data sets.

```

1 | def align_data(data_leg1,data_leg2,symbols):
2 |     combined_df = pd.concat([data_leg1,data_leg2],axis=1)
3 |     combined_df = combined_df.fillna(method='pad')
4 |     data_panel = pd.Panel({symbols[0]: combined_df.ix[:,0:4], symbols[1]
5 |     return(data_panel)

```

To construct the spread, we will run a rolling regression on the prices to extract the hedge ratio. This is then piped in to the following equation:

$$spread_{a,b} = price_a - \beta price_b$$

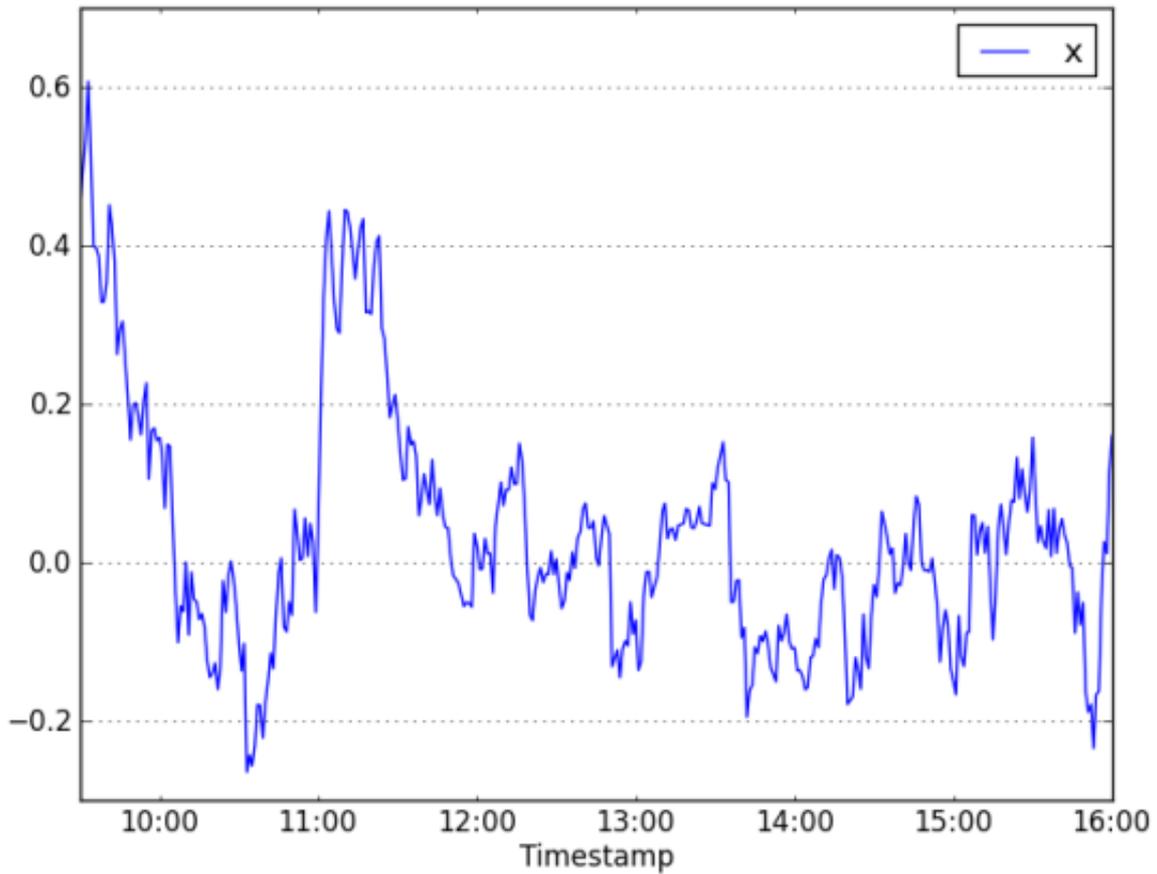
<https://systematicedge.files.wordpress.com/2014/03/codecogseqn.gif>

Given two series of prices, the following helper function will return a dictionary of the model and the spread. Following is the spread displayed.

```

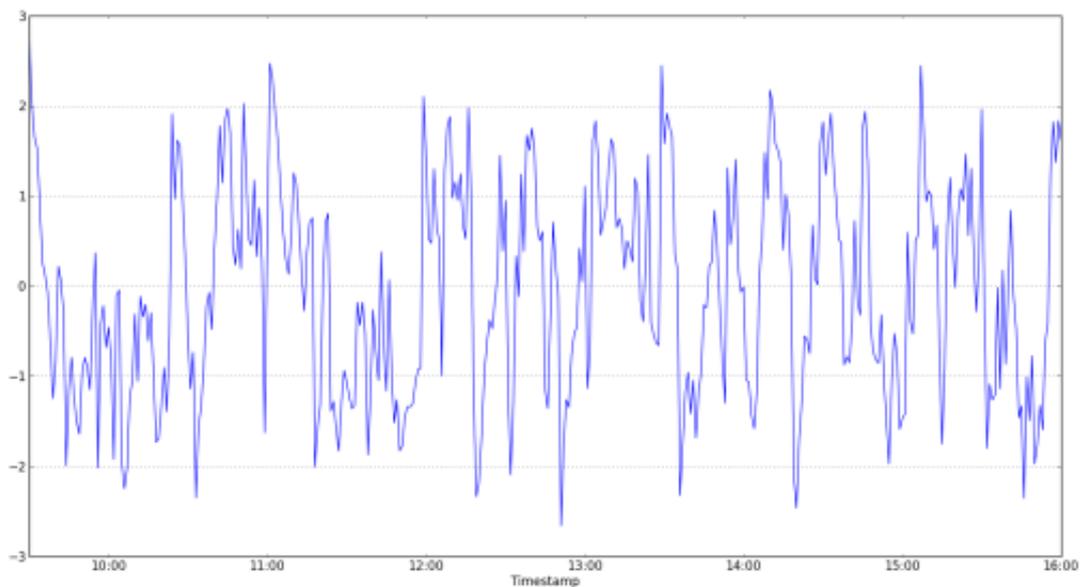
1 | def construct_spread(priceY,priceX,ols_lookback):
2 |     data = pd.DataFrame({'x':priceX,'y':priceY})
3 |     model = {}
4 |     model['model_ols'] = pd.ols(y=data.y, x=data.x, window=ols_lookback)
5 |     model['spread'] = data.y - (model['model_ols'].beta*data.x)
6 |     return model

```



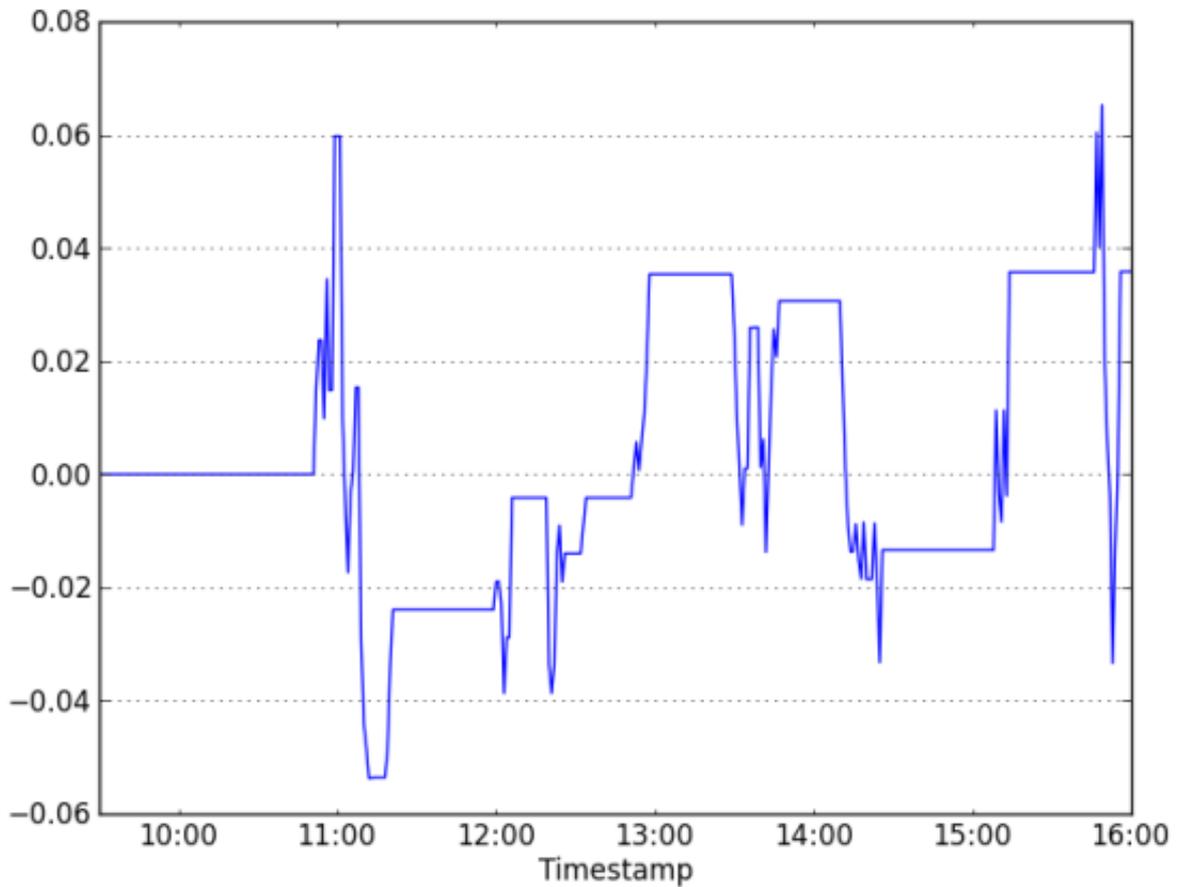
To normalize it, simply subtract a rolling mean and divide that by the rolling standard deviation. Image of normalized spread follows.

```
1 | zscore = lambda x: (x[-1] - x.mean()) / x.std(ddof=1)
2 | sprd['zs'] = pd.rolling_apply(sprd, zs_window, zscore) # zscore
```



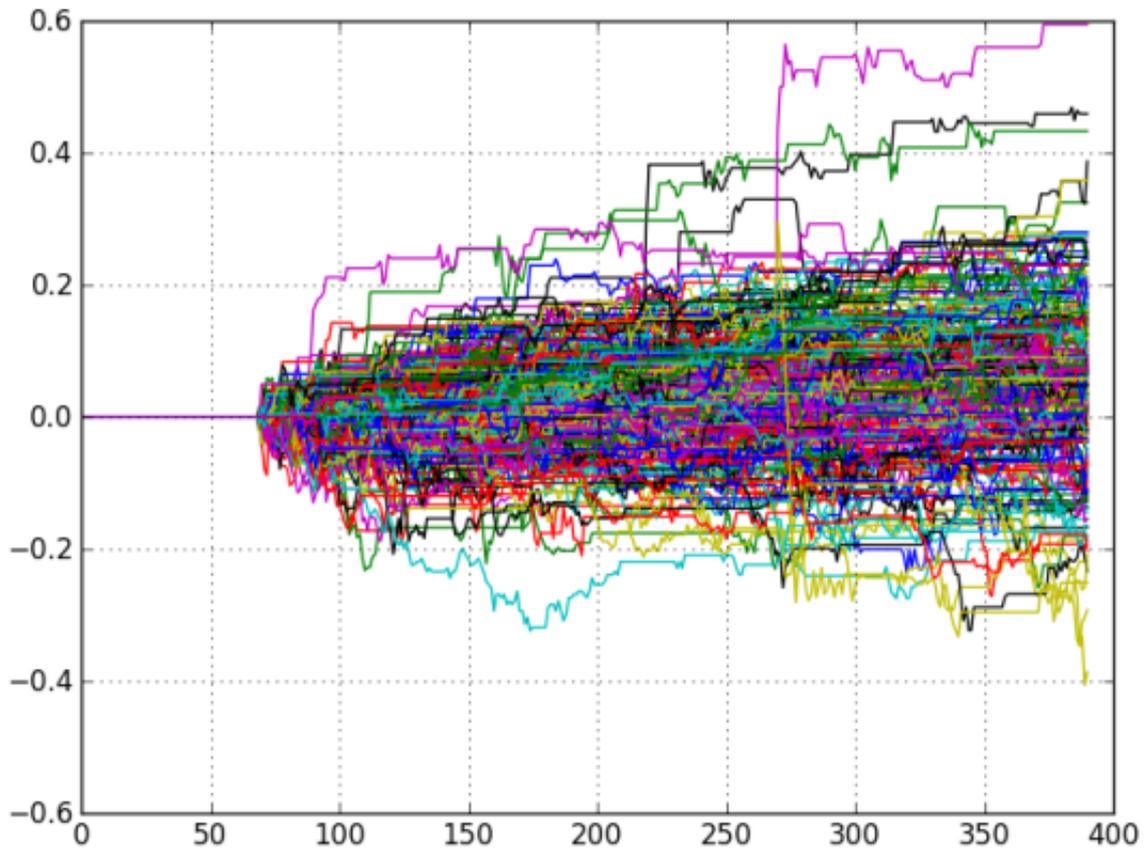
https://systematicedge.files.wordpress.com/2014/03/figure_12.png

Without changing our parameters, a $+2$ std will be our trigger point. At this threshold, there is a total of 16 trades. Here is the performance if we took all the trades for the day, frictionless:



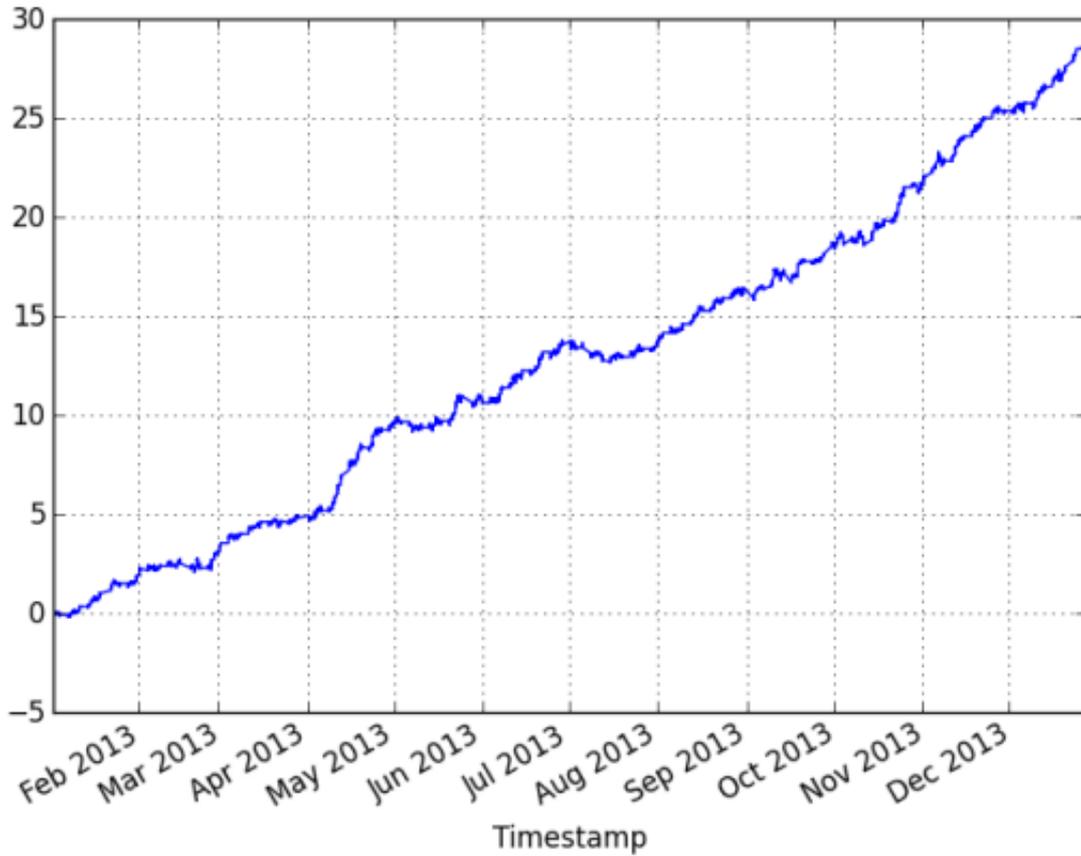
https://systematicedge.files.wordpress.com/2014/03/figure_13.png

Pretty ugly in my opinion but its only a day. Lets display all the daily equity performance distributed for the whole year of 2013.



https://systematicedge.files.wordpress.com/2014/03/figure_3.png

The flat line initially is for the 60 bar lookback window for each day, unrealistic but it does give a rough picture on the returns. The average final portfolio gain is 0.06 with std of 0.13. The performance is pretty stellar when you look at 2013 as a whole. Comparing this to the other spread construction in my last post, it seems to reduce the variance of returns when incorporating a longer lookback period.



(<https://systematicedge.files.wordpress.com/2014/03/2013.png>) Coming up in the next instalment I want to investigate whether incorporating Garch models for volatility forecasting will help improve the performance of spread trading.

Thanks for reading,

Mike

About these ads (<https://wordpress.com/about-these-ads/>)

<p>посмотреть</p>	<p>Японские свечи. Метод анализа акций...</p> <p>754 руб</p> <p>посмотреть</p>
-------------------	---

Posted in [mean reversion](#) on [March 1, 2014](#) by [Author](#). [3 Comments](#)

3 comments

1. **Alex** says:

March 13, 2014 at 11:07 am

Thanks for sharing your thoughts!

It looks like the strategy can't even withstand 1 cent round trip commission per contract – upward sloping equity curve rapidly turns into a downward...

Do you have any ideas of how to struggle with it?

REPLY

2. **Wonderwall** says:

April 30, 2014 at 6:26 am

Thank you for your post!

Can you share the trading strategy code? If can't, I am wondering how you will close the position since hedge ratio(beta) is time-varying. the Beta when you open the position is totally different from the Beta when you want to close your previous position.

And How did you calculate zscore in the first rolling window?

REPLY

3. **Hero** says:

May 14, 2014 at 11:37 pm

Be careful with the QuantQuote adjusted data. Their back adjuster is unreliable and known to have problems (missing dividends is common). If you are not careful, the price history will drift over time. Compare with CSI for an example. Or use raw, un-adjusted price data.

REPLY

[CREATE A FREE WEBSITE OR BLOG AT WORDPRESS.COM. THE SUITS THEME.](https://systematicedge.wordpress.com/2014/03/01/energy-stat-arb-part-2/)