w8_pairs_trading

November 26, 2020

[28]:
```
#install pandas-datareader package with pip
#!pip install pandas-datareader
```

```
Requirement already satisfied: pandas-datareader in /usr/local/lib/python3.6
/dist-packages (0.9.0)
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.6
/dist-packages (from pandas-datareader) (2.23.0)
Requirement already satisfied: lxml in /usr/local/lib/python3.6/dist-packages
(from pandas-datareader) (4.2.6)
Requirement already satisfied: pandas>=0.23 in /usr/local/lib/python3.6/dist-
packages (from pandas-datareader) (1.1.4)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6
/dist-packages (from requests>=2.19.0->pandas-datareader) (2020.11.8)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.6
/dist-packages (from requests>=2.19.0->pandas-datareader) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.6/dist-packages (from requests>=2.19.0->pandas-
datareader) (1.24.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.6/dist-
packages (from requests>=2.19.0->pandas-datareader) (2.10)
Requirement already satisfied: numpy>=1.15.4 in /usr/local/lib/python3.6/dist-
packages (from pandas>=0.23->pandas-datareader) (1.18.5)
Requirement already satisfied: python-dateutil>=2.7.3 in
/usr/local/lib/python3.6/dist-packages (from pandas>=0.23->pandas-datareader)
(2.8.1)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-
packages (from pandas>=0.23->pandas-datareader) (2018.9)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.6/dist-
packages (from python-dateutil>=2.7.3->pandas>=0.23->pandas-datareader) (1.15.0)
```

[29]:
```
#import libraries
import datetime as dt
import math

import matplotlib.pyplot as plt
from matplotlib import style
```

```
import seaborn as sn

import pandas as pd
import pandas_datareader as web

import numpy as np
import statsmodels
from statsmodels.tsa.stattools import coint

np.random.seed(107)
```

[30]:
```
start = dt.date(2019,1,1)
end = dt.date.today()
```

[31]:
```
#symbol_list = ['IG', 'NVS', 'PBH', 'GNC', 'JNJ', 'PFE', 'MRTX','SPX', 'CPHD']

symbol_list = ['NVS', 'PBH']

df_stock_prices = pd.DataFrame()
```

[32]:
```
for symbol in symbol_list:
  try:
    df = web.DataReader(symbol, 'yahoo', start, end)
    df_stock_prices[symbol] = df['Adj Close']
  except:
    pass
```

[33]:
```
df_stock_prices.head()
```

[33]:

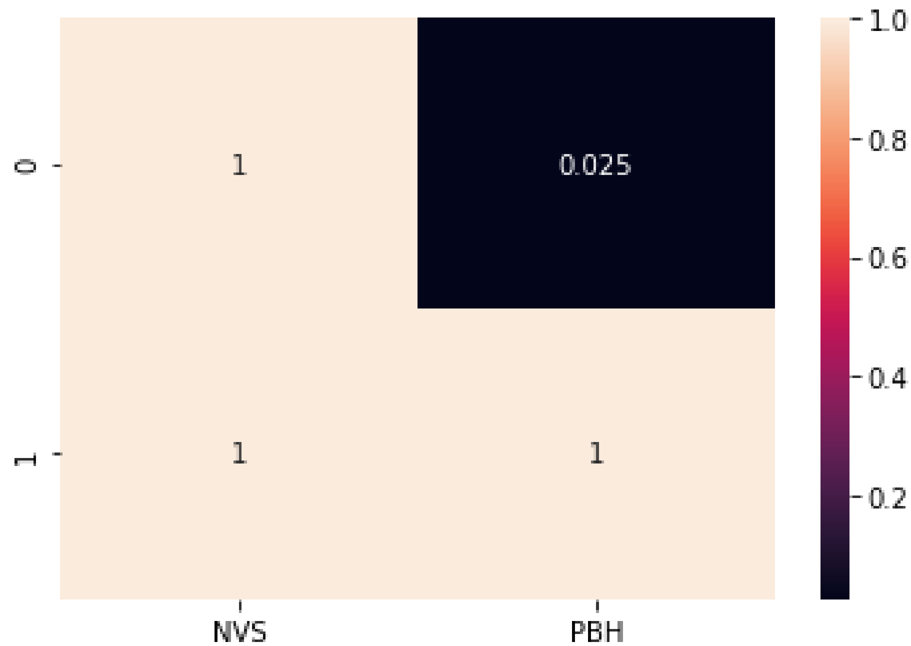
	NVS	PBH
Date		
2019-01-02	70.406586	30.709999
2019-01-03	70.976135	30.209999
2019-01-04	71.939354	31.379999
2019-01-07	71.093399	31.440001
2019-01-08	72.634537	31.719999

[34]:
```
def find_cointegrated_pairs(stockprice_data):
  symb = list(df_stock_prices.columns)
  n = len(symb)
  score_matrix = np.zeros((n,n))
  pvalue_matrix = np.ones((n,n))
  pairs = []

  for i in range(n):
    for j in range(i+1, n):
      S1 = stockprice_data[symb[i]]
      S2 = stockprice_data[symb[j]]
      S1 = S1.fillna(S1.mean())
      S2 = S2.fillna(S2.mean())
```

```
        result = coint(S1,S2)
        score = result[0]
        pvalue = result[1]
        score_matrix[i,j] = score
        pvalue_matrix[i,j] = pvalue

        if(pvalue < 0.01):
          pairs.append((S1,S2,symb[i], symb[j]))
    return score_matrix, pvalue_matrix, pairs
```

[35]:
```
score, pvalue, _ = find_cointegrated_pairs(df_stock_prices)
```

[36]:
```
df_pvalue = pd.DataFrame(pvalue, columns=list(df_stock_prices.columns))
sn.heatmap(df_pvalue, annot=True)
plt.show()
```



[36]:
```
```

1 Week 8

1.1 Visualize the Time Series in Graph

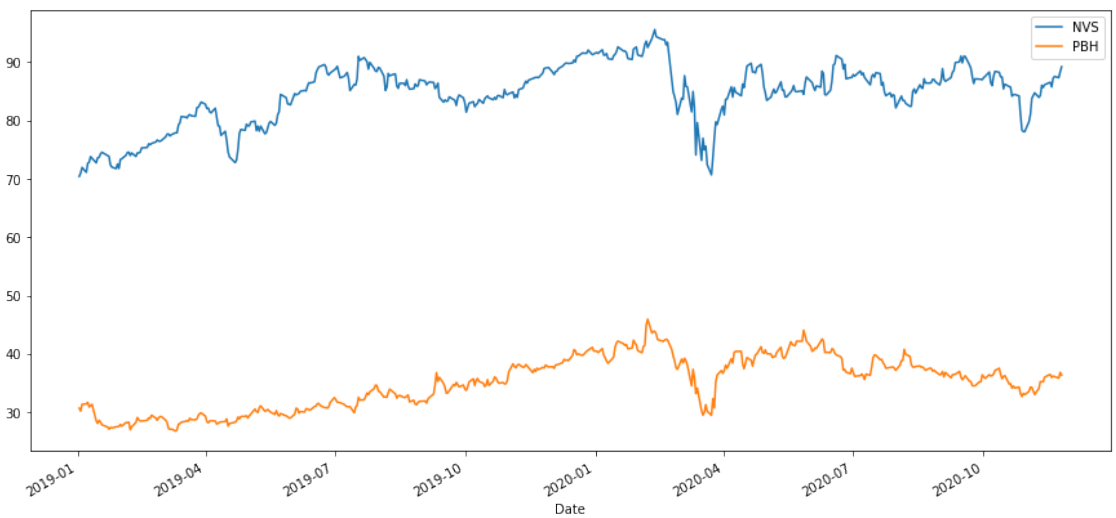
[37]:
```
X = df_stock_prices['NVS']
Y = df_stock_prices['PBH']
```

[38]:
```
print("Correlation : " , X.corr(Y))
print("Cointegration test p-value : ", pvalue[0][1])
```

```
Correlation :  0.708861839412714
Cointegration test p-value :  0.025479193494852313
```

We see that these two stock prices are both cointegrated and correlated.

[39]:
```
s1 = X
s2 = Y

pd.concat([s1, s2], axis=1 ).plot(figsize=(15,7))
plt.show()
```



1.2 How to make a pairs trade ?

Because two cointegrated time series (such as X and Y above) drift towards and apart from each other, there will be times when the spread is high and times when the spread is low.

We make a pairs trade by buying one security and selling another. This way, if both securities go down together or go up together, we neither make nor lose money — we are market neutral.

1.3 Getting Ratio of time series

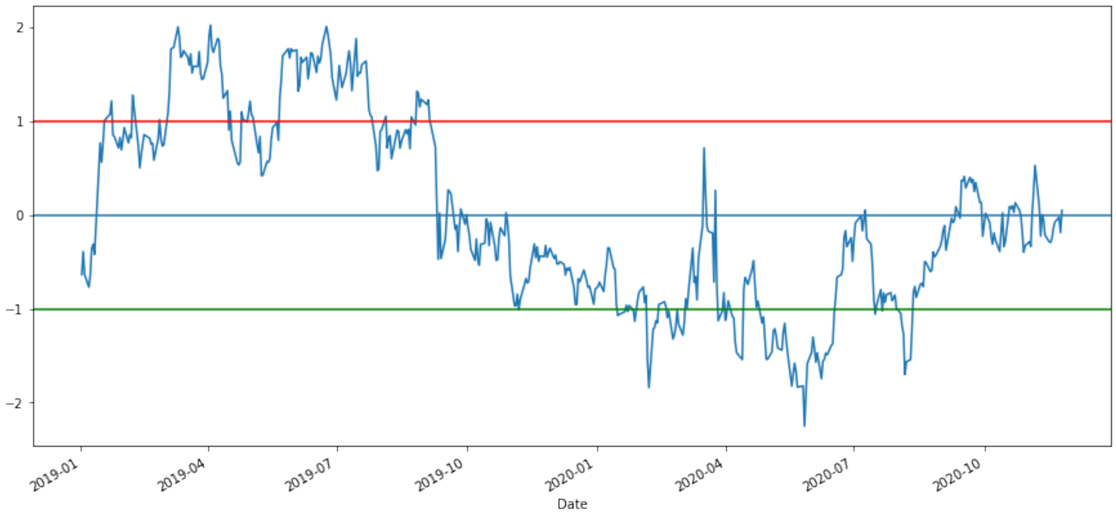
[40]:
```
ratios = s1 / s2
```

1.4 Z-Score

$ZScore(Value) = (Value - Mean)/StandardDeviation$

[41]:
```python
def zscore(series):
    return (series - series.mean()) / np.std(series)
```

[42]:
```python
zscore(ratios).plot(figsize=(15,7))
plt.axhline(zscore(ratios).mean())
plt.axhline(1.0, color='red')
plt.axhline(-1.0, color='green')
plt.show()
```



It's easier to now observe the ratio now moves around the mean, but sometimes is prone to large divergences from the mean, which we can take advantages of.

By setting two other lines placed at the z-score of 1 and -1, we can clearly see that for the most part, any big divergences from the mean eventually converge back. This is precisely what we want for a pairs trading strategy.

1.5 Trading Signals

When conducting any type of trading strategy, it's always important to clearly define and delineate at what point you will actually make a trade. As in, what is the best indicator that I need to buy or sell a particular stock?

1.5.1 Setup rules

We're going to use the **ratio** time series that we've created to see if it tells us whether to buy or sell a particular moment in time.

We'll start off by creating a prediction variable Y

. If the ratio is positive, it will signal a "buy," otherwise, it will signal a sell. The prediction model is as follows:

$Y_t = sign(Ratio_{t+1} - Ratio_t)$

What's great about pair trading signals is that we don't need to know absolutes about where the prices will go, all we need to know is where it's heading: up or down.

[60]:
```
#signal of sell
ratios[0] - ratios[1]
```

[60]: -0.05679785892271472

[59]:
```
#signal of buy
ratios[1] - ratios[2]
```

[59]: 0.056902775776478176

1.6 Train Test Split

[61]:
```
print(len(ratios) * .70 )
```

```
336.7
```

[62]:
```
train = ratios[:336]
test = ratios[336:]
```

[63]:
```
train.head()
```

[63]:
```
Date
2019-01-02    2.292627
2019-01-03    2.349425
2019-01-04    2.292522
2019-01-07    2.261240
2019-01-08    2.289866
dtype: float64
```

[64]:
```
test.head()
```

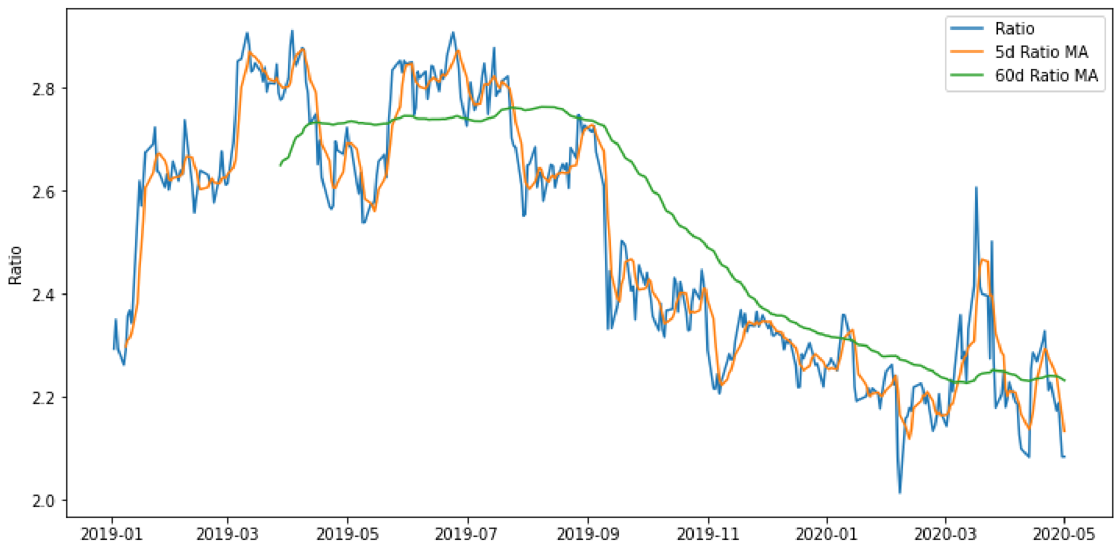
[64]:
```
Date
2020-05-04    2.101051
2020-05-05    2.152654
2020-05-06    2.158028
2020-05-07    2.140834
2020-05-08    2.110586
dtype: float64
```

1.7 Feature Engineering

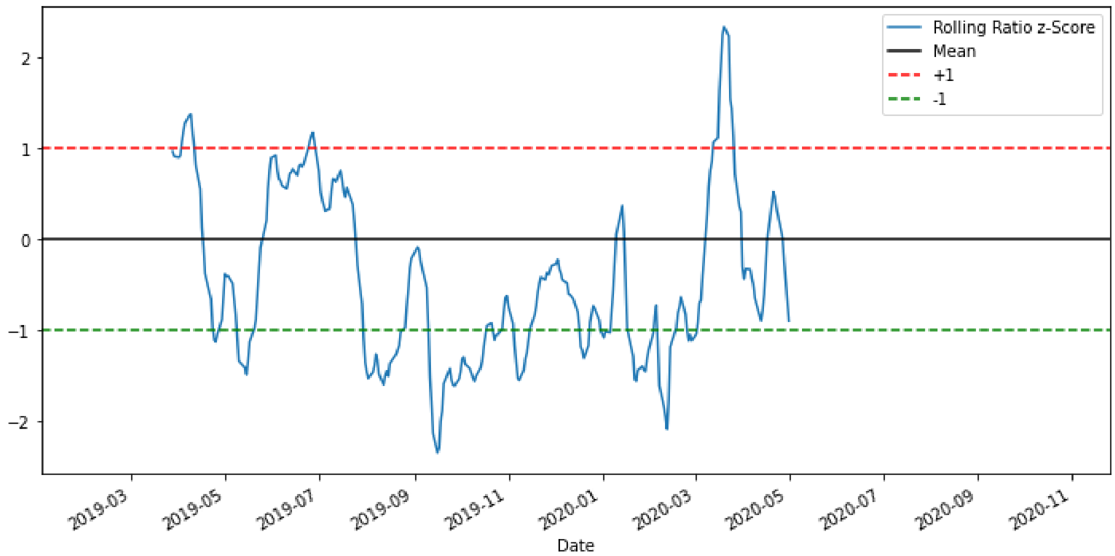
We need to find out what features are actually important in determining the direction of the ratio moves. Knowing that the ratios always eventually revert back to the mean, maybe the moving averages and metrics related to the mean will be important.

- 60 day Moving Average of Ratio
- 5 day Moving Average of Ratio
- 60 day Standard Deviation
- z score

[65]:
```
ratios_mavg60 = train.rolling(window=60, center=False).mean()
ratios_mavg5 = train.rolling(window=5, center=False).mean()

std_60 = train.rolling(window=60, center=False).std()

zscore_60_5 = (ratios_mavg5 - ratios_mavg60)/std_60

plt.figure(figsize=(12, 6))
plt.plot(train.index, train.values)
plt.plot(ratios_mavg5.index, ratios_mavg5.values)
plt.plot(ratios_mavg60.index, ratios_mavg60.values)
plt.legend(['Ratio', '5d Ratio MA', '60d Ratio MA'])

plt.ylabel('Ratio')
plt.show()
```



[70]:
```
plt.figure(figsize=(12,6))
zscore_60_5.plot()
plt.xlim('2019-01-02', '2020-11-25')
plt.axhline(0, color='black')
plt.axhline(1.0, color='red', linestyle='--')
plt.axhline(-1.0, color='green', linestyle='--')
plt.legend(['Rolling Ratio z-Score', 'Mean', '+1', '-1'])
plt.show()
```



1.8 Creating a Model

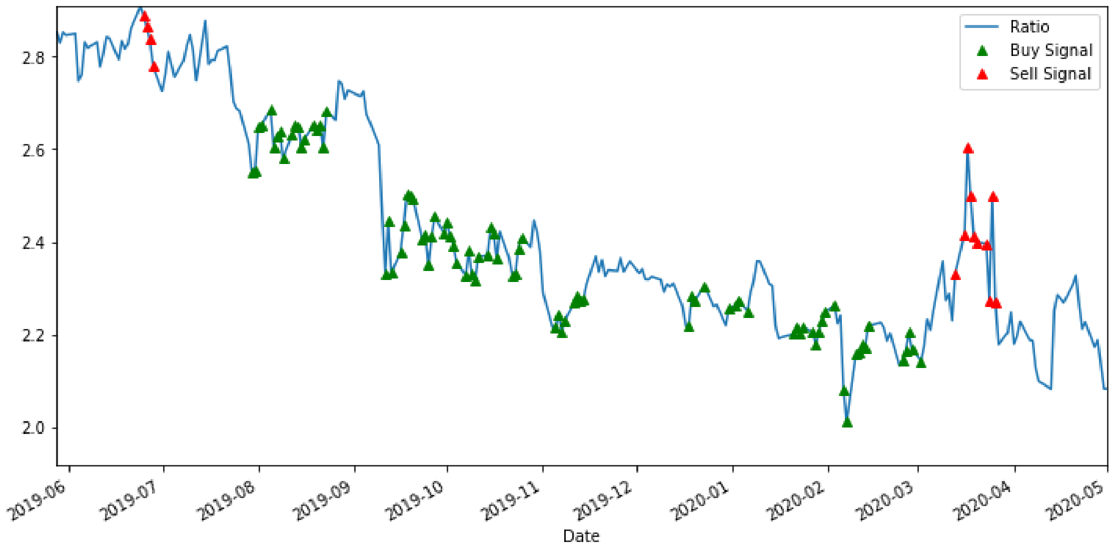
A standard normal distribution has a mean of 0 and a standard deviation 1. Looking at the plot, it's pretty clear that if the time series moves 1 standard deviation beyond the mean, it tends to revert back towards the mean. Using these models, we can create the following trading signals:

```
Buy(1) whenever the z-score is below -1, meaning we expect the ratio to increase.
Sell(-1) whenever the z-score is above 1, meaning we expect the ratio to decrease.
```

We can use our model on actual data

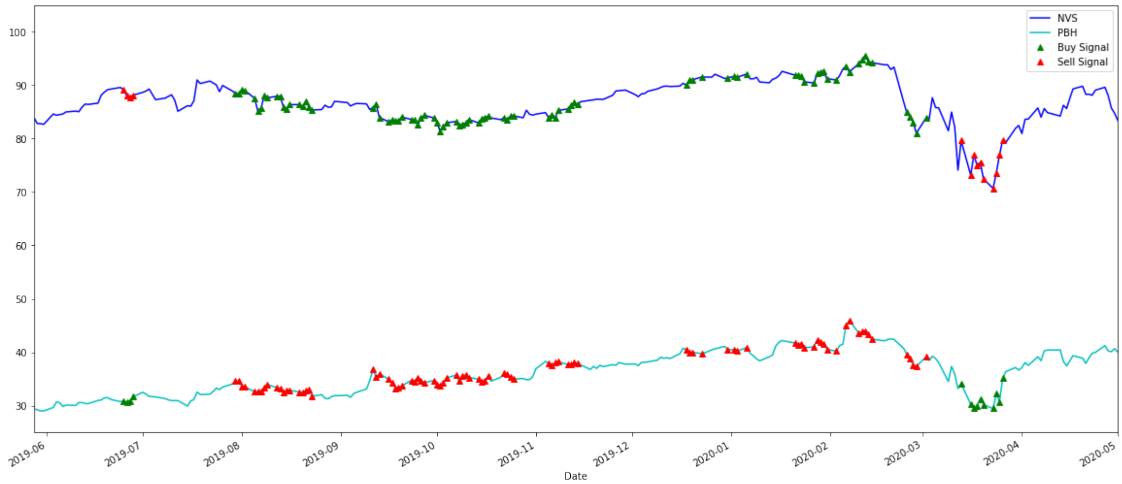
[77]:
```
plt.figure(figsize=(12,6))

train[100:].plot()

buy = train.copy()
sell = train.copy()

#Buy signal (1) whenever the z-score is below -1, meaning we expect the ratio␣
↪to increase.
buy[zscore_60_5>-1] = 0

#Sell signal (-1) whenever the z-score is above 1, meaning we expect the ratio␣
↪to decrease.
sell[zscore_60_5<1] = 0

buy[100:].plot(color='g', linestyle='None', marker='^')
sell[100:].plot(color='r', linestyle='None', marker='^')
```

```
x1, x2, y1, y2 = plt.axis()
plt.axis((x1, x2, ratios.min(), ratios.max()))
plt.xlim('2019-05-28','2020-05-01')
plt.legend(['Ratio', 'Buy Signal', 'Sell Signal'])
plt.show()
```



[96]:
```
plt.figure(figsize=(20,9))

S1 = df_stock_prices['NVS'].iloc[:336]
S2 = df_stock_prices['PBH'].iloc[:336]

S1[60:].plot(color='b')
S2[60:].plot(color='c')
buyR = 0*S1.copy()
sellR = 0*S1.copy()

# When you buy the ratio, you buy stock S1 and sell S2
buyR[buy!=0] = S1[buy!=0]
sellR[buy!=0] = S2[buy!=0]

# When you sell the ratio, you sell stock S1 and buy S2
buyR[sell!=0] = S2[sell!=0]
sellR[sell!=0] = S1[sell!=0]

buyR[60:].plot(color='g', linestyle='None', marker='^')
sellR[60:].plot(color='r', linestyle='None', marker='^')

x1, x2, y1, y2 = plt.axis()
plt.axis((x1, x2, min(S1.min(), S2.min()), max(S1.max(), S2.max())))
```

```
plt.ylim(25, 105)
plt.xlim('2019-05-28', '2020-05-01')

plt.legend(['NVS', 'PBH', 'Buy Signal', 'Sell Signal'])
plt.show()
```



Now we can clearly see when we should buy or sell on the respective stocks.

1.9 how much can we expect to make of this strategy?

[131]:
```
# Trade using a simple strategy
def trade(S1, S2, window1, window2):

    # If window length is 0, algorithm doesn't make sense, so exit
    if (window1 == 0) or (window2 == 0):
        return 0

    # Compute rolling mean and rolling standard deviation
    ratios = S1/S2

    #get moving average of first window
    ma1 = ratios.rolling(window=window1, center=False).mean()

    #get moving average of second window
    ma2 = ratios.rolling(window=window2, center=False).mean()

    std = ratios.rolling(window=window2, center=False).std()
    zscore = (ma1 - ma2)/std

    # Simulate trading
    # Start with no money and no positions
```

```
    money = 0
    countS1 = 0
    countS2 = 0
    for i in range(len(ratios)):
        # Sell short if the z-score is > 1
        if zscore[i] < -1:
            money += S1[i] - S2[i] * ratios[i]
            countS1 -= 1
            countS2 += ratios[i]
            print('Selling Ratio %s %s %s %s'%(money, ratios[i],␣
↪countS1,countS2))
        # Buy long if the z-score is < -1
        elif zscore[i] > 1:
            money -= S1[i] - S2[i] * ratios[i]
            countS1 += 1
            countS2 -= ratios[i]
            print('Buying Ratio %s %s %s %s'%(money,ratios[i], countS1,countS2))
        # Clear positions if the z-score between -.5 and .5
        elif abs(zscore[i]) < 0.75:
            money += S1[i] * countS1 + S2[i] * countS2
            countS1 = 0
            countS2 = 0
            print('Exit pos %s %s %s %s'%(money,ratios[i], countS1,countS2))


    return money
```

[132]:
```
S1 = df_stock_prices['NVS'].iloc[:336]
S2 = df_stock_prices['PBH'].iloc[:336]

trade(S1,S2,60,5)
```

```
Selling Ratio 0.0 2.776671704893592 -1 2.776671704893592
Selling Ratio 0.0 2.779347119214838 -2 5.556018824108429
Selling Ratio 0.0 2.8197090263339533 -3 8.375727850442383
Selling Ratio 0.0 2.883682844066577 -4 11.25941069450896
Selling Ratio 0.0 2.9117004472529513 -5 14.17111114176191
Selling Ratio 0.0 2.8565948049227785 -6 17.02770594668469
Selling Ratio 0.0 2.8436392559044554 -7 19.871345202589147
Selling Ratio 0.0 2.8781208255850084 -8 22.749466028174155
Selling Ratio 0.0 2.8746130807059154 -9 25.62407910888007
Selling Ratio 0.0 2.8115571486603894 -10 28.43563625754046
Selling Ratio 0.0 2.790642578422786 -11 31.226278835963246
Selling Ratio 0.0 2.7302597491338916 -12 33.956538585097135
Selling Ratio 0.0 2.748942787259153 -13 36.70548137235629
Exit pos 64.75238371697674 2.6510279044188474 0 0
Exit pos 64.75238371697674 2.6977622212934196 0 0
Buying Ratio 64.75238371697674 2.5701045632660007 1 -2.5701045632660007
```

```
Buying Ratio 64.75238371697674 2.5645010757219424 2 -5.134605638987943
Buying Ratio 64.75238371697674 2.5717434145850766 3 -7.706349053573019
Buying Ratio 64.75238371697674 2.695929162725529 4 -10.402278216298548
Buying Ratio 64.75238371697674 2.678163027314482 5 -13.08044124361303
Buying Ratio 64.75238371697674 2.6710679935635606 6 -15.751509237176592
Buying Ratio 64.75238371697674 2.697161337511693 7 -18.448670574688286
Buying Ratio 64.75238371697674 2.7227589656060305 8 -21.171429540294316
Buying Ratio 64.75238371697674 2.6882349469593647 9 -23.85966448725368
Buying Ratio 64.75238371697674 2.684080009997846 10 -26.543744497251527
Buying Ratio 64.75238371697674 2.61359843482569 11 -29.157342932077217
Buying Ratio 64.75238371697674 2.594729783733232 12 -31.75207271581045
Buying Ratio 64.75238371697674 2.6351931658132375 13 -34.387265881623684
Buying Ratio 64.75238371697674 2.537975130503775 14 -36.92524101212746
Buying Ratio 64.75238371697674 2.538969515703234 15 -39.464210527830694
Buying Ratio 64.75238371697674 2.5733922027412373 16 -42.03760273057193
Buying Ratio 64.75238371697674 2.5710566699102086 17 -44.60865940048214
Buying Ratio 64.75238371697674 2.5809871558303845 18 -47.189646556312525
Buying Ratio 64.75238371697674 2.630174368866219 19 -49.819820925178746
Buying Ratio 64.75238371697674 2.657476587343704 20 -52.47729751252245
Buying Ratio 64.75238371697674 2.6699795159504287 21 -55.14727702847288
Buying Ratio 64.75238371697674 2.6256563203504077 22 -57.77293334882329
Buying Ratio 64.75238371697674 2.730915947422535 23 -60.503849296245825
Exit pos 161.86542412891708 2.774364689594118 0 0
Exit pos 161.86542412891708 2.8350196625292763 0 0
Exit pos 161.86542412891708 2.852933492503958 0 0
Selling Ratio 161.86542412891708 2.8299765583856296 -1 2.8299765583856296
Selling Ratio 161.86542412891708 2.853048583231711 -2 5.683025141617341
Selling Ratio 161.86542412891708 2.8467470048286865 -3 8.529772146446028
Selling Ratio 161.86542412891708 2.8500788002314525 -4 11.37985094667748
Selling Ratio 161.86542412891708 2.747841682860491 -5 14.127692629537972
Selling Ratio 161.86542412891708 2.7615083455075675 -6 16.88920097504554
Selling Ratio 161.86542412891708 2.831949691462262 -7 19.721150666507803
Selling Ratio 161.86542412891708 2.819387117106399 -8 22.5405377836142
Selling Ratio 161.86542412891708 2.8320269856207103 -9 25.372564769234913
Selling Ratio 161.86542412891708 2.7782654260091166 -10 28.15083019524403
Selling Ratio 161.86542412891708 2.806016724954035 -11 30.956846920198064
Selling Ratio 161.86542412891708 2.8429053467103222 -12 33.799752266908385
Selling Ratio 161.86542412891708 2.840684016242758 -13 36.640436283151146
Selling Ratio 161.86542412891708 2.7940902709960938 -14 39.43452655414724
Selling Ratio 161.86542412891708 2.8347128369411454 -15 42.269239391088384
Selling Ratio 161.86542412891708 2.81739245080125 -16 45.086631841889634
Selling Ratio 161.86542412891708 2.8278999111557988 -17 47.91453175304543
Selling Ratio 161.86542412891708 2.862450190551345 -18 50.77698194359677
Selling Ratio 161.86542412891708 2.9084229851176016 -19 53.68540492871438
Selling Ratio 161.86542412891708 2.889316220160582 -20 56.57472114887496
Selling Ratio 161.86542412891708 2.864790771859678 -21 59.43951192073464
Selling Ratio 161.86542412891708 2.8381461089501965 -22 62.27765802968484
Selling Ratio 161.86542412891708 2.781329465333096 -23 65.05898749501793
```

```
Selling Ratio 161.86542412891708 2.7258076114001266 -24 67.78479510641806
Selling Ratio 161.86542412891708 2.811365322803888 -25 70.59616042922195
Selling Ratio 161.86542412891708 2.7565694183936125 -26 73.35272984761556
Selling Ratio 161.86542412891708 2.7930614368851043 -27 76.14579128450067
Selling Ratio 161.86542412891708 2.825442705731619 -28 78.97123399023229
Selling Ratio 161.86542412891708 2.847585473331946 -29 81.81881946356424
Selling Ratio 161.86542412891708 2.8148687467586004 -30 84.63368821032284
Selling Ratio 161.86542412891708 2.7487654599359437 -31 87.38245367025878
Selling Ratio 161.86542412891708 2.8781807730592903 -32 90.26063444331807
Selling Ratio 161.86542412891708 2.784395221406452 -33 93.04502966472452
Selling Ratio 161.86542412891708 2.793402079065077 -34 95.8384317437896
Selling Ratio 161.86542412891708 2.8134536966009254 -35 98.65188544039053
Selling Ratio 161.86542412891708 2.822610296953107 -36 101.47449573734363
Selling Ratio 161.86542412891708 2.7678973951551007 -37 104.24239313249873
Exit pos 303.29306033673606 2.702645032341352 0 0
Exit pos 303.29306033673606 2.688331497418841 0 0
Exit pos 303.29306033673606 2.6835778469783804 0 0
Buying Ratio 303.29306033673606 2.610978060392765 1 -2.610978060392765
Buying Ratio 303.29306033673606 2.5506480034399397 2 -5.1616260638327045
Buying Ratio 303.29306033673606 2.5541350768926137 3 -7.715761140725318
Buying Ratio 303.29306033673606 2.6487510128003904 4 -10.364512153525709
Buying Ratio 303.29306033673606 2.652118499006518 5 -13.016630652532227
Buying Ratio 303.29306033673606 2.6853738181166302 6 -15.702004470648857
Buying Ratio 303.29306033673606 2.6065747113602176 7 -18.308579182009076
Buying Ratio 303.29306033673606 2.6293468557074084 8 -20.937926037716483
Buying Ratio 303.29306033673606 2.637367324136233 9 -23.575293361852715
Buying Ratio 303.29306033673606 2.5799139183329154 10 -26.155207280185632
Buying Ratio 303.29306033673606 2.633609081767656 11 -28.788816361953288
Buying Ratio 303.29306033673606 2.650864672428236 12 -31.439681034381522
Buying Ratio 303.29306033673606 2.647751850874964 13 -34.08743288525648
Buying Ratio 303.29306033673606 2.606040270876077 14 -36.69347315613256
Buying Ratio 303.29306033673606 2.6234260896315926 15 -39.316899245764155
Buying Ratio 303.29306033673606 2.6519225967964744 16 -41.96882184256063
Buying Ratio 303.29306033673606 2.641690443069924 17 -44.610512285630556
Buying Ratio 303.29306033673606 2.6529651766304982 18 -47.26347746226105
Buying Ratio 303.29306033673606 2.6048957246698095 19 -49.86837318693086
Buying Ratio 303.29306033673606 2.6837429954419165 20 -52.55211618237278
Buying Ratio 303.29306033673606 2.6635553549809576 21 -55.215671537353735
Buying Ratio 303.29306033673606 2.7473863429217253 22 -57.96305788027546
Exit pos 368.4307932405104 2.7275579158231027 0 0
Exit pos 368.4307932405104 2.714634981238734 0 0
Buying Ratio 368.4307932405104 2.674763273227639 1 -2.674763273227639
Buying Ratio 368.4307932405104 2.6095867925370984 2 -5.2843500657647375
Buying Ratio 368.4307932405104 2.3306415875194713 3 -7.614991653284209
Buying Ratio 368.4307932405104 2.4449673752657706 4 -10.05995902854998
Buying Ratio 368.4307932405104 2.3324198481689082 5 -12.392378876718887
Buying Ratio 368.4307932405104 2.378356619722061 6 -14.770735496440947
Buying Ratio 368.4307932405104 2.436530484462094 7 -17.207265980903042
```

```
Buying Ratio 368.4307932405104 2.5025145241127964 8 -19.70978050501584
Buying Ratio 368.4307932405104 2.4985371464599035 9 -22.208317651475742
Buying Ratio 368.4307932405104 2.493204575417978 10 -24.70152222689372
Buying Ratio 368.4307932405104 2.4050964794570593 11 -27.10661870635078
Buying Ratio 368.4307932405104 2.4145501721030245 12 -29.521168878453803
Buying Ratio 368.4307932405104 2.349688824739913 13 -31.870857703193717
Buying Ratio 368.4307932405104 2.4125941434352516 14 -34.28345184662897
Buying Ratio 368.4307932405104 2.4552967434810875 15 -36.738748590110056
Buying Ratio 368.4307932405104 2.4173234987670136 16 -39.15607208887707
Buying Ratio 368.4307932405104 2.4412545004400754 17 -41.59732658931714
Buying Ratio 368.4307932405104 2.411740451388889 18 -44.00906704070603
Buying Ratio 368.4307932405104 2.391123628032188 19 -46.40019066873822
Buying Ratio 368.4307932405104 2.3559819057641915 20 -48.75617257450241
Buying Ratio 368.4307932405104 2.327945047015021 21 -51.08411762151743
Buying Ratio 368.4307932405104 2.3809216608162878 22 -53.46503928233372
Buying Ratio 368.4307932405104 2.3292767162054355 23 -55.794315998539155
Buying Ratio 368.4307932405104 2.315579740751863 24 -58.10989573929102
Buying Ratio 368.4307932405104 2.3673499513653105 25 -60.47724569065633
Buying Ratio 368.4307932405104 2.3705235786571937 26 -62.847769269313524
Buying Ratio 368.4307932405104 2.4306784575766187 27 -65.27844772689014
Buying Ratio 368.4307932405104 2.4201051818414245 28 -67.69855290873156
Buying Ratio 368.4307932405104 2.364477623263482 29 -70.06303053199504
Buying Ratio 368.4307932405104 2.422353221782373 30 -72.48538375377741
Buying Ratio 368.4307932405104 2.364916622079062 31 -74.85030037585648
Buying Ratio 368.4307932405104 2.3280028096489302 32 -77.17830318550541
Buying Ratio 368.4307932405104 2.329239309600368 33 -79.50754249510578
Buying Ratio 368.4307932405104 2.3841699252563524 34 -81.89171242036214
Buying Ratio 368.4307932405104 2.408593123708191 35 -84.30030554407033
Buying Ratio 368.4307932405104 2.388955913791103 36 -86.68926145786143
Buying Ratio 368.4307932405104 2.4460789188357768 37 -89.13534037669721
Buying Ratio 368.4307932405104 2.4224091703132182 38 -91.55774954701043
Buying Ratio 368.4307932405104 2.379527532626635 39 -93.93727707963707
Buying Ratio 368.4307932405104 2.289774855391676 40 -96.22705193502874
Buying Ratio 368.4307932405104 2.2154278113348305 41 -98.44247974636357
Buying Ratio 368.4307932405104 2.2146670976421423 42 -100.65714684400572
Buying Ratio 368.4307932405104 2.244013227799801 43 -102.90116007180552
Buying Ratio 368.4307932405104 2.2055646668840287 44 -105.10672473868955
Buying Ratio 368.4307932405104 2.2305905197337994 45 -107.33731525842335
Buying Ratio 368.4307932405104 2.268002367115306 46 -109.60531762553865
Buying Ratio 368.4307932405104 2.282470298918667 47 -111.88778792445731
Buying Ratio 368.4307932405104 2.2717657621859444 48 -114.15955368664326
Buying Ratio 368.4307932405104 2.2754362231039216 49 -116.43498990974719
Buying Ratio 368.4307932405104 2.3083686189457895 50 -118.74335852869298
Buying Ratio 368.4307932405104 2.3686641686618675 51 -121.11202269735485
Buying Ratio 368.4307932405104 2.3349675812001593 52 -123.446990278555
Buying Ratio 368.4307932405104 2.361076561180321 53 -125.80806683973533
Buying Ratio 368.4307932405104 2.3252550831323773 54 -128.1333219228677
Buying Ratio 368.4307932405104 2.338912076597237 55 -130.47223399946495
```

```
Buying Ratio 368.4307932405104 2.3369462327032067 56 -132.80918023216816
Buying Ratio 368.4307932405104 2.364835250035189 57 -135.17401548220334
Buying Ratio 368.4307932405104 2.335565967772869 58 -137.50958144997622
Buying Ratio 368.4307932405104 2.3581640125888037 59 -139.86774546256504
Buying Ratio 368.4307932405104 2.3322313769574143 60 -142.19997683952246
Buying Ratio 368.4307932405104 2.3409560513735603 61 -144.54093289089602
Buying Ratio 368.4307932405104 2.318996412181961 62 -146.859929303078
Buying Ratio 368.4307932405104 2.31858843796646 63 -149.17851774104446
Buying Ratio 368.4307932405104 2.3245920010731456 64 -151.5031097421176
Buying Ratio 368.4307932405104 2.31801126325996 65 -153.82112100537756
Buying Ratio 368.4307932405104 2.291462184716457 66 -156.11258319009403
Buying Ratio 368.4307932405104 2.30861244676197 67 -158.421195636856
Buying Ratio 368.4307932405104 2.3031863336201615 68 -160.72438197047617
Buying Ratio 368.4307932405104 2.3103421215747155 69 -163.03472409205088
Buying Ratio 368.4307932405104 2.26053086414656 70 -165.29525495619743
Buying Ratio 368.4307932405104 2.2178241801983214 71 -167.51307913639576
Buying Ratio 368.4307932405104 2.2185548712111784 72 -169.73163400760694
Buying Ratio 368.4307932405104 2.2822093087799287 73 -172.01384331638687
Buying Ratio 368.4307932405104 2.2741357317206536 74 -174.28797904810753
Buying Ratio 368.4307932405104 2.303670239689495 75 -176.59164928779703
Buying Ratio 368.4307932405104 2.2909267412866305 76 -178.88257602908365
Buying Ratio 368.4307932405104 2.2606856764955836 77 -181.14326170557922
Buying Ratio 368.4307932405104 2.263969158103383 78 -183.4072308636826
Buying Ratio 368.4307932405104 2.2188489651659316 79 -185.62607982884853
Buying Ratio 368.4307932405104 2.256152117693866 80 -187.8822319465424
Buying Ratio 368.4307932405104 2.2629056460249375 81 -190.14513759256732
Buying Ratio 368.4307932405104 2.2736927265584574 82 -192.4188303191258
Buying Ratio 368.4307932405104 2.250446073490538 83 -194.66927639261633
Buying Ratio 368.4307932405104 2.2901661209091304 84 -196.95944251352546
Buying Ratio 368.4307932405104 2.316938245332255 85 -199.27638075885773
Exit pos 415.85059333795243 2.3588174851866857 0 0
Exit pos 415.85059333795243 2.3580637148795343 0 0
Exit pos 415.85059333795243 2.3092051782080256 0 0
Exit pos 415.85059333795243 2.2184035080010376 0 0
Exit pos 415.85059333795243 2.190732862638477 0 0
Buying Ratio 415.85059333795243 2.2006863347478056 1 -2.2006863347478056
Buying Ratio 415.85059333795243 2.2165487178245935 2 -4.4172350525724
Buying Ratio 415.85059333795243 2.200771500086202 3 -6.618006552658601
Buying Ratio 415.85059333795243 2.215823770695642 4 -8.833830323354242
Buying Ratio 415.85059333795243 2.204799486536704 5 -11.038629809890946
Buying Ratio 415.85059333795243 2.17644937781532 6 -13.215079187706266
Buying Ratio 415.85059333795243 2.2044235378927963 7 -15.419502725599063
Buying Ratio 415.85059333795243 2.2287765779601796 8 -17.648279303559242
Buying Ratio 415.85059333795243 2.2485320328752403 9 -19.89681133643448
Buying Ratio 415.85059333795243 2.2618645056689193 10 -22.1586758421034
Buying Ratio 415.85059333795243 2.223105644199627 11 -24.381781486303026
Buying Ratio 415.85059333795243 2.2408513448324547 12 -26.62263283113548
Buying Ratio 415.85059333795243 2.157149064497617 13 -28.779781895633096
```

```
Buying Ratio 415.85059333795243 2.161199223910028 14 -30.940981119543125
Buying Ratio 415.85059333795243 2.1778638512165425 15 -33.11884497075967
Buying Ratio 415.85059333795243 2.1723761813220546 16 -35.291221152081725
Buying Ratio 415.85059333795243 2.2183000193294755 17 -37.5095211714112
Buying Ratio 415.85059333795243 2.2252968268779125 18 -39.73481799828912
Buying Ratio 415.85059333795243 2.213153299537701 19 -41.94797129782682
Buying Ratio 415.85059333795243 2.1857946985666663 20 -44.133765996393485
Buying Ratio 415.85059333795243 2.2021620335790515 21 -46.335928029972536
Buying Ratio 415.85059333795243 2.1326200710173873 22 -48.46854810098992
Buying Ratio 415.85059333795243 2.142645213705867 23 -50.61119331469579
Buying Ratio 415.85059333795243 2.1636961776633705 24 -52.77488949235916
Buying Ratio 415.85059333795243 2.2047212677317556 25 -54.979610760090914
Buying Ratio 415.85059333795243 2.1686267702706843 26 -57.1482375303616
Buying Ratio 415.85059333795243 2.142294930781233 27 -59.29053246114283
Buying Ratio 415.85059333795243 2.173641801227208 28 -61.46417426237004
Buying Ratio 415.85059333795243 2.23255235030932 29 -63.696726612679356
Buying Ratio 415.85059333795243 2.2093202168248776 30 -65.90604682950423
Exit pos 478.4074472727493 2.2516417627488745 0 0
Exit pos 478.4074472727493 2.3584249254565766 0 0
Exit pos 478.4074472727493 2.2732137911773207 0 0
Selling Ratio 478.4074472727493 2.229680823992952 -1 2.229680823992952
Selling Ratio 478.4074472727493 2.3311858580745017 -2 4.560866682067454
Selling Ratio 478.4074472727493 2.415455671436151 -3 6.976322353503605
Selling Ratio 478.4074472727493 2.4986666361490886 -4 9.474988989652694
Selling Ratio 478.4074472727493 2.413286453127683 -5 11.888275442780376
Selling Ratio 478.4074472727493 2.399602300750325 -6 14.2878777435307
Selling Ratio 478.4074472727493 2.3955931582693326 -7 16.683470901800032
Selling Ratio 478.4074472727493 2.273879475371205 -8 18.957350377171238
Selling Ratio 478.4074472727493 2.501301234943099 -9 21.458651612114338
Selling Ratio 478.4074472727493 2.268563140888105 -10 23.727214753002443
Exit pos 549.278112693926 2.178021853184046 0 0
Exit pos 549.278112693926 2.2039828272922497 0 0
Exit pos 549.278112693926 2.247818873016966 0 0
Buying Ratio 549.278112693926 2.195378005626678 1 -2.195378005626678
Buying Ratio 549.278112693926 2.227430122211525 2 -4.422808127838203
Buying Ratio 549.278112693926 2.1886649824993327 3 -6.611473110337536
Buying Ratio 549.278112693926 2.185532063408745 4 -8.797005173746282
Buying Ratio 549.278112693926 2.126148541676396 5 -10.923153715422679
Buying Ratio 549.278112693926 2.0986890386755386 6 -13.021842754098216
Buying Ratio 549.278112693926 2.0815820855406946 7 -15.103424839638912
Buying Ratio 549.278112693926 2.2523512008922486 8 -17.35577604053116
Exit pos 584.0107629881287 2.2853711346148247 0 0
Exit pos 584.0107629881287 2.277907890199454 0 0
Exit pos 584.0107629881287 2.2679706207259667 0 0
Selling Ratio 584.0107629881287 2.3073759735639814 -1 2.3073759735639814
Selling Ratio 584.0107629881287 2.3271815431813065 -2 4.6345575167452875
Selling Ratio 584.0107629881287 2.2652643429634054 -3 6.899821859708693
Exit pos 592.7846415837278 2.2268066845245493 0 0
```