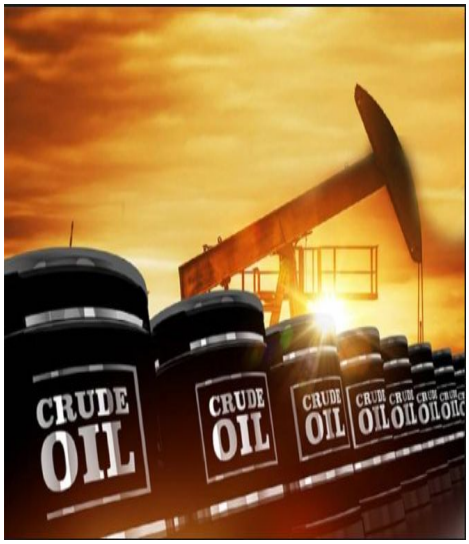
Are Canadian Dollar and Crude Oil Correlated and How?

VS



Issam Ozeir

Mentor : Hassan Shaban

Springboard Data Science career track 2019

How Crude Oil and Candaian Dollar are Correlated

Abstract

Throughout analysis we reached out for the following results:

- The Crude Oil and Candaian dollar are correlated
- The rate of correlation is strong enough to prove the relation
- Both settlement volume spike in June each year

In our analysis we were able to quantify the fundamental relationship between Crude oil and Canadian dollar.

Introduction

The exchange rate between Canada and the U.S. has been strongly correlated to the price of oil in recent years. Over the long run, when the price of oil rises, the value of the Canadian dollar (also called the loonie) also usually rises relative to that of the U.S. dollar.
That correlation can be directly attributed to the way Canada earns most of its U.S. dollars – from the sale of crude oil – and the percentage of Canada's revenue that this constitutes.
The main initiative in this project is try to prove this relationship and quantify the correlation between Crude oil and Canadian Dollar
Data Wrangling Section:
The Data we chose for this project is a Time series Data from **Quandl.com.**
We imported the Data sets from **Quandl.com** using the API provided for crude oil and CAD.
The crude oil API is **Crude_Oil** and that of CAD is **CAD**

Step 1
Importing Libraries

```
#First step is to import the Libraries we will need to use
import pandas as pd
from datetime import datetime
import numpy as np
from sklearn import preprocessing
import matplotlib.pyplot as plt
import seaborn as sns
% matplotlib inline
sns.set()
np.random.seed(42)
plt.rcParams['figure.figsize']=[15,5]
```

After importing the Data sets the first step was to create a time series data with the following steps:

```
#Now open the crude oil as csv file from the provided API and save it to df as pandas data frame
df = pd.read_csv("https://www.quandl.com/api/v3/datasets/CHRIS/CME_CL17.csv[redacted]",parse_dates=['Date'])
```

```
# Importing the Cad dollar data set
df_cad = pd.read_csv("https://www.quandl.com/api/v3/datasets/CHRIS/CME_CD4.csv?[redacted]",parse_dates=['Date'])
```

The second step is to print the top 5 rows of the Data and the last five rows to check the Data df.head() and df.tail()

The first findings was that the data has missing values as the following figures show

```
# first step is to print the top 5 rows of the Data and the last five rows to check the Da
df.head()
```

Date	Open	High	Low	Last	Change	Settle	Volume	Previous Day Open Interest
1983-03-30	28.90	28.95	28.70	28.95	NaN	28.95	18.0	14.0
1983-03-31	28.75	28.75	28.75	28.75	NaN	28.75	3.0	13.0
1983-04-04	28.95	28.95	28.95	28.95	NaN	28.95	0.0	13.0
1983-04-05	29.00	29.10	29.00	29.10	NaN	29.10	5.0	18.0
1983-04-06	29.10	29.20	29.10	29.20	NaN	29.20	5.0	19.0

```
df_cad.head()
```

Date	Open	High	Low	Last	Change	Settle	Volume	Previous Day Open Interest
1977-03-11	0.93	0.931	0.93	0.93	NaN	0.93	1.0	1.0
1977-03-14	0.93	0.930	0.93	0.93	NaN	0.93	0.0	1.0
1977-03-15	0.93	0.930	0.93	0.93	NaN	0.93	0.0	1.0
1977-03-16	0.93	0.930	0.93	0.93	NaN	0.93	0.0	1.0
1977-03-17	0.93	0.930	0.93	0.93	NaN	0.93	0.0	1.0

Then printing the shape of the Data using df.shape the crude oil data has 9084 rows and 8 columns and the CAD data has 10112 rows and 8 columns

Calling the .info() method provides important information about a DataFrame, such as the number of rows, number of columns,number of non-missing values in each column, and the data type stored in each column

df.info()

```
df.info()
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 9106 entries, 1983-03-30 to 2019-06-27
Data columns (total 8 columns):
Open                            8635 non-null float64
High                            8800 non-null float64
Low                             8836 non-null float64
Last                            8956 non-null float64
Change                          1338 non-null float64
Settle                          9106 non-null float64
Volume                          9106 non-null float64
Previous Day Open Interest      9106 non-null float64
dtypes: float64(8)
memory usage: 960.3 KB
```

As the above figure shows that we have a lot of missing data especially in the change column we have 7768 rows of missing data. Since the change is the difference between the settle price of two consecutive trading days.since the settle data has no missing values finding the changes is easy and would be accurate values.

Since the Data is a numeric data we would run a statistical analysis to check for outliers in case of any **df.describe()** as the following Table shows that Volume raises a question mark about the outliers and also Settle ,Last ,low,High and Open but visualizing the data shows that the outliers are fact based and not bad data or errors in our Data frame but actual Data the price level of crude oil and the trading volume levels are actual values so the outliers are accurate historical Data

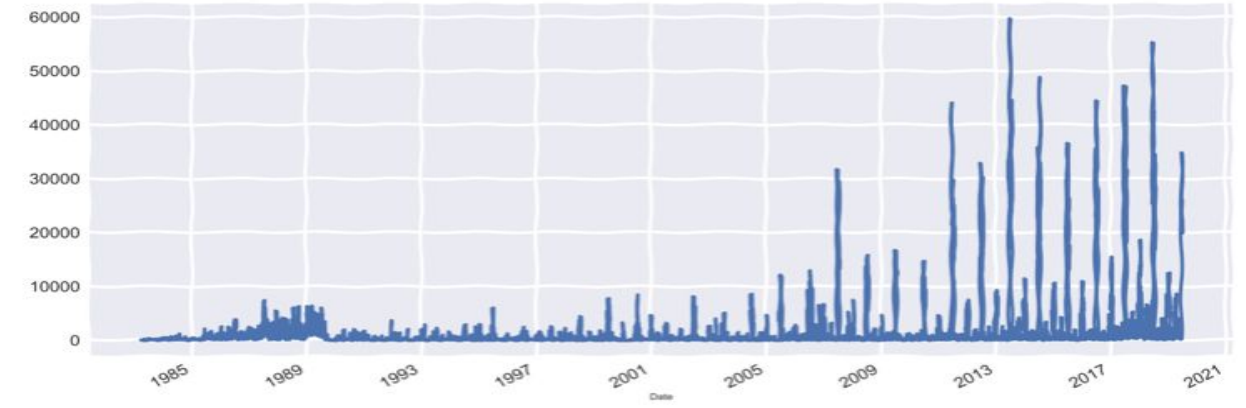
```
df.describe()
```

	Open	High	Low	Last	Change	Settle	Volume	Previous Day Open Interest
count	8835.000000	8800.000000	8836.000000	8956.000000	1338.000000	9106.00000	9106.000000	9106.000000
mean	41.836340	42.209819	42.040237	42.349295	0.630097	42.54078	1149.548320	12120.488250
std	29.439638	29.281092	29.129827	29.049085	0.602967	28.85391	3999.824787	25341.188845
min	10.750000	10.850000	10.400000	10.840000	0.010000	10.84000	0.000000	0.000000
25%	19.050000	19.167500	19.117500	19.200000	0.210000	19.26000	5.000000	1989.000000
50%	25.180000	25.535000	25.450000	25.800000	0.460000	26.24000	151.000000	5099.000000
75%	65.405000	64.970000	64.330000	64.157500	0.850000	63.79000	610.000000	11806.000000
max	145.880000	146.860000	143.550000	145.450000	5.930000	145.45000	59716.000000	237239.000000

Showing the Volume distribution

This shows that the Volume plot is left skewed with the Data centralized to the right where we have spikes of trading volume in 2008 during the Financial Crisis and Years 2013,2015,2016,2017 and this is due to many reasons

```
df['Volume'].plot(figsize=(20,10), linewidth=4, fontsize=20)
<matplotlib.axes._subplots.AxesSubplot at 0x14ab2664a20>
```



Dealing with missing values

Now we have to clean the data our data has a lot of NAN values.there is more than an approach to deal with NAN, one of them is to drop NAN values but since the number of the remaining solid rows of Data is few compared to the original Data so dropping the NaN does not make sense as taking

this avenue would affect the analysis. So we should deal with the missing values in a different manner like filling the missing Data.The type of our Data suggest that we should follow a certain approach in dealing with filling Data.First we have to fill the NaN values in the following columns High,Open,Low and Last applying the forward Method Since the Data has outliers and applying the mean approach would affect the analysis. for example The mean is $42 per barrel as statistic table shows and some of the NaN values is at the prices level of 80 to 100 dollars per barrel so it is of more statistical sense to fill the data with the forward fill method rather than the mean approach As for the Change column missing Data is just the change between the settlement Prices between two consecutive trading days . so we will deal with the Change column later after we finish with the main columns High,Open,Low and Last

The Change is the difference between the settlement prices of two consecutive dates and its value in the data is absolute value so it has no directional sense and has a lot of missing values so dropping this column and creating a new changes column is the most convenient way. and this is done by creating a __shifted__ column to capture the settlement changes from day to day with a one day forwarded to future using shift method.Then creating a new column and appending this column to the data frame __changes__ to capture the changes in settlement from day to day

df_new=df.fillna(method='ffill')

```
df_new=df.fillna(method='ffill')
df_new=df_new.drop(['Change'],axis=1)
df_new['shifted']=df_new.Settle.shift()
df_new['Changes']=df_new['Settle']-df_new['shifted']
df_new.tail()
```

Date	Open	High	Low	Last	Settle	Volume	Previous Day Open Interest	shifted	Changes
2019-06-21	54.66	55.22	53.95	54.60	54.53	26108.0	145794.0	54.57	-0.04
2019-06-24	54.80	54.93	53.96	54.67	54.78	21987.0	144955.0	54.53	0.25
2019-06-25	54.67	55.75	54.30	55.66	55.07	21840.0	145683.0	54.78	0.29
2019-06-26	55.63	56.26	55.46	55.67	55.89	34807.0	145373.0	55.07	0.82
2019-06-27	55.64	56.04	55.27	55.54	55.82	19853.0	143318.0	55.89	-0.07

The same steps in cleaning the crude oil data done to clean the Candian Dollar Data

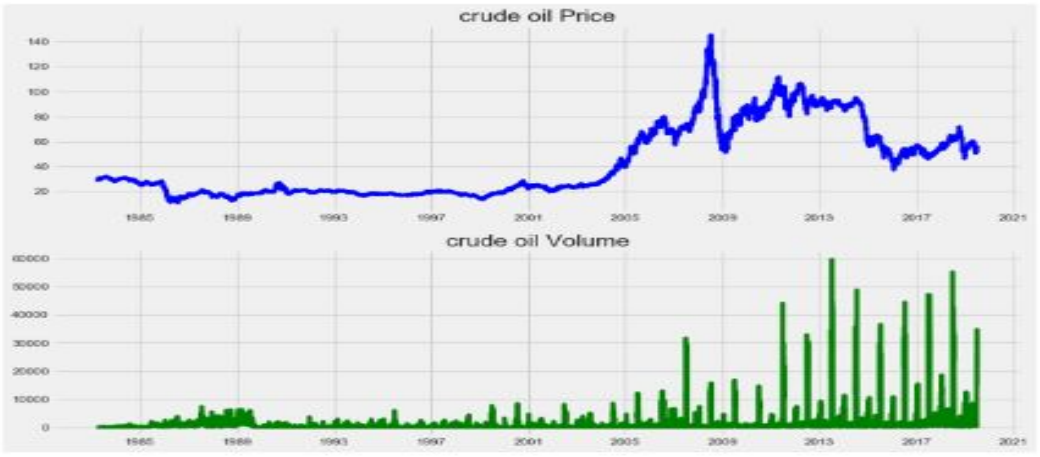
```
df_cadnew=df_cad.fillna(method='ffill')
df_cadnew=df_cadnew.drop(['Change'],axis=1)
df_cadnew['shifted']=df_cadnew.Settle.shift()
df_cadnew['Changes']=df_cadnew['Settle']-df_cadnew['shifted']
df_cadnew.tail()
```

Date	Open	High	Low	Last	Settle	Volume	Previous Day Open Interest	shifted	Changes
2019-06-21	0.7594	0.75940	0.7594	0.7594	0.75970	52.0	158.0	0.76045	-0.00075
2019-06-24	0.7609	0.76115	0.7609	0.7610	0.76075	10.0	207.0	0.75970	0.00105
2019-06-25	0.7615	0.76150	0.7615	0.7615	0.76105	1.0	217.0	0.76075	0.00030
2019-06-26	0.7630	0.76300	0.7630	0.7630	0.76520	2.0	217.0	0.76105	0.00415
2019-06-27	0.7630	0.76300	0.7651	0.7651	0.76635	0.0	217.0	0.76520	0.00115

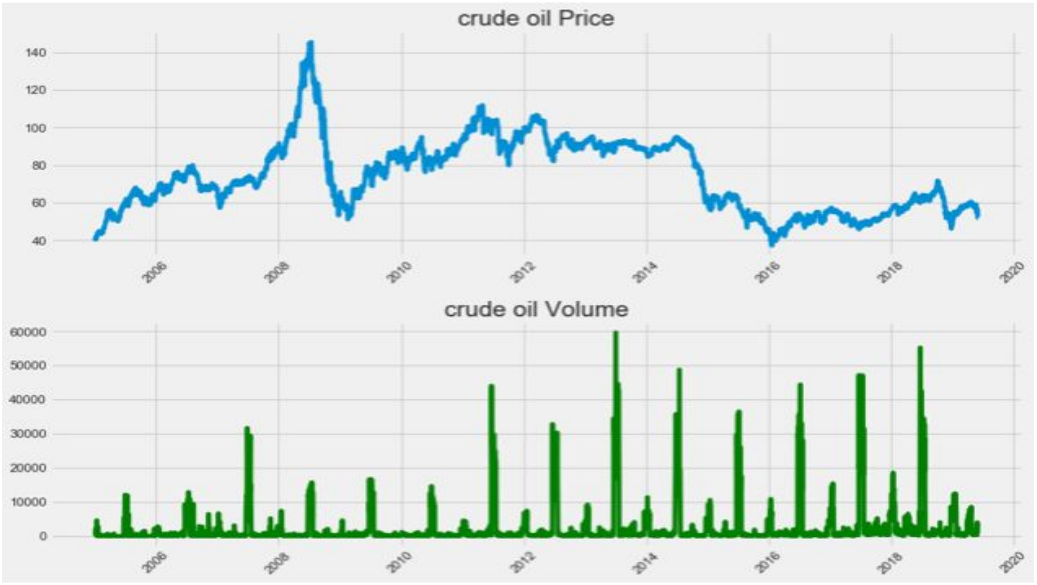
Now both data sets are clean and ready for the analysis. The first approach is Visualizing the data we have to investigate any trends.

Let us now try to visualize our data looking for trends that would help us in our analysis

```
# create a  first sub dataplot t with  the settle price and a second sub dataplot  with the volume and plot both sub dataset
sns.set(rc={"figure.figsize": (12, 8)})
plt.style.use('fivethirtyeight')
plt.subplot(2,1,1)
plt.title('crude oil Price')
plt.plot(df_new['Settle'],color='blue')
plt.subplot(2,1,2)
plt.title('crude oil Volume')
plt.plot(df_new['Volume'],color='green')
plt.tight_layout()
plt.show()
```

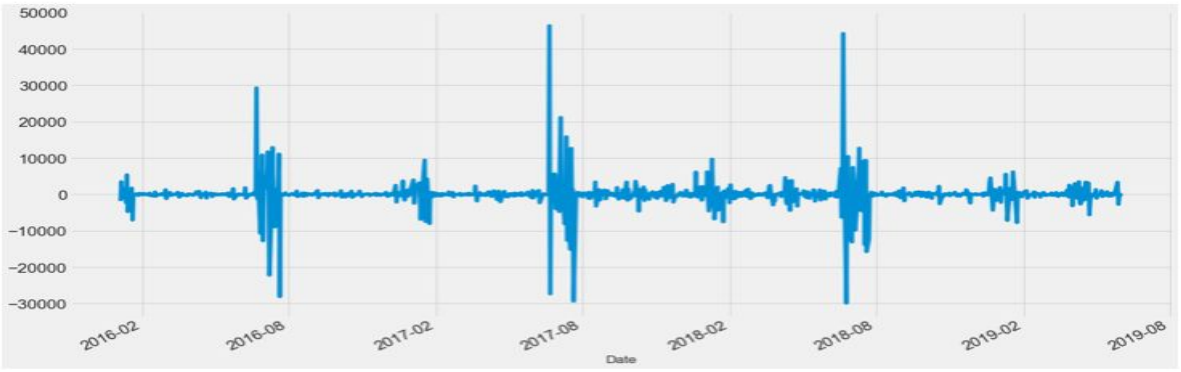


A closer look at the plot shows that the volume has a seasonal pattern where it jumps in July each year with one difference which is the trading volume from year to year



drilling down more to find in which month this spike occurs it is clearly happening in July each year but more aggressively after 2012 the below plot shows seasonality in trading volume of oil each year in july

```
df_volume['2016-01-01':'2019-06-03'].diff().plot(figsize=(20,10), linewidth=5, fontsize=20)
<matplotlib.axes._subplots.AxesSubplot at 0x2d4d99366d8>
```



What drives the volume of crude oil to the highs especially in july of each year? It is the consumption of Oil in the US AND Canada due to summer and people would travel a lot also what is driving the volume in January and february is the oil needed for heat? so future contracts and options on oil are settled in July and February and this what causes the volume to a high spikes

As for the settelment price for oil as the time plot shows that the maximum settelement price is 145.5

```
df_new['Settle']['2008-01-01':'2019-05-31'].max()
```

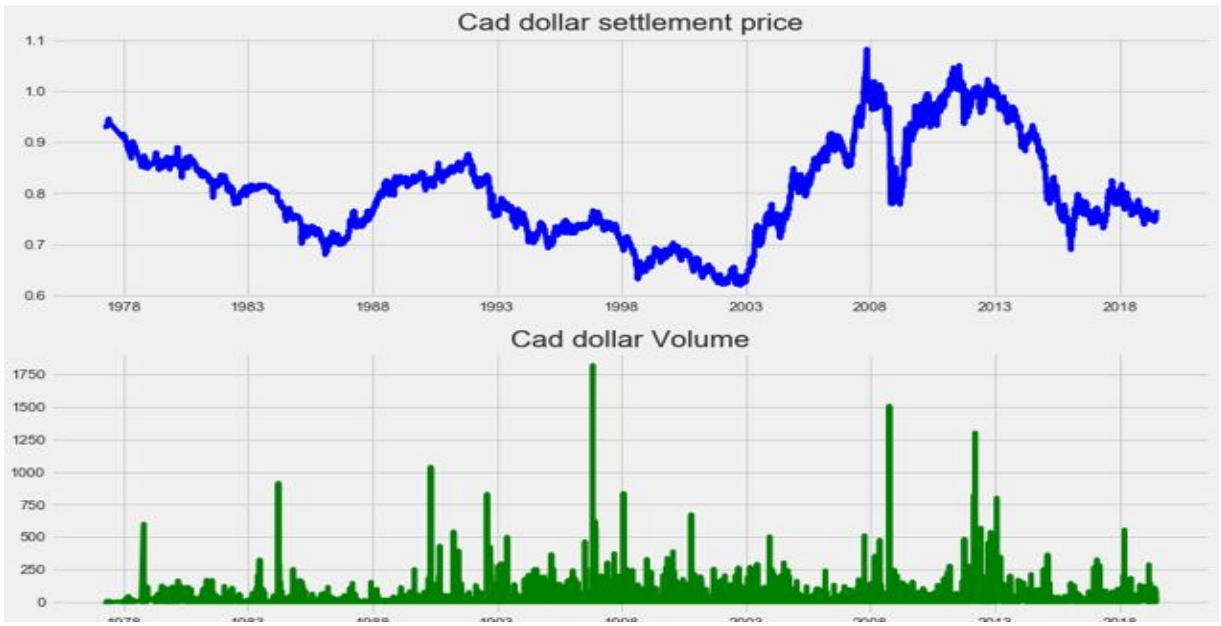
```
145.45
```

As for the settelment price for oil as the time plot shows that the minimum settelement price is 37.22

```
df_new['Settle']['2008-01-01':'2019-05-31'].min()
```

```
37.22
```

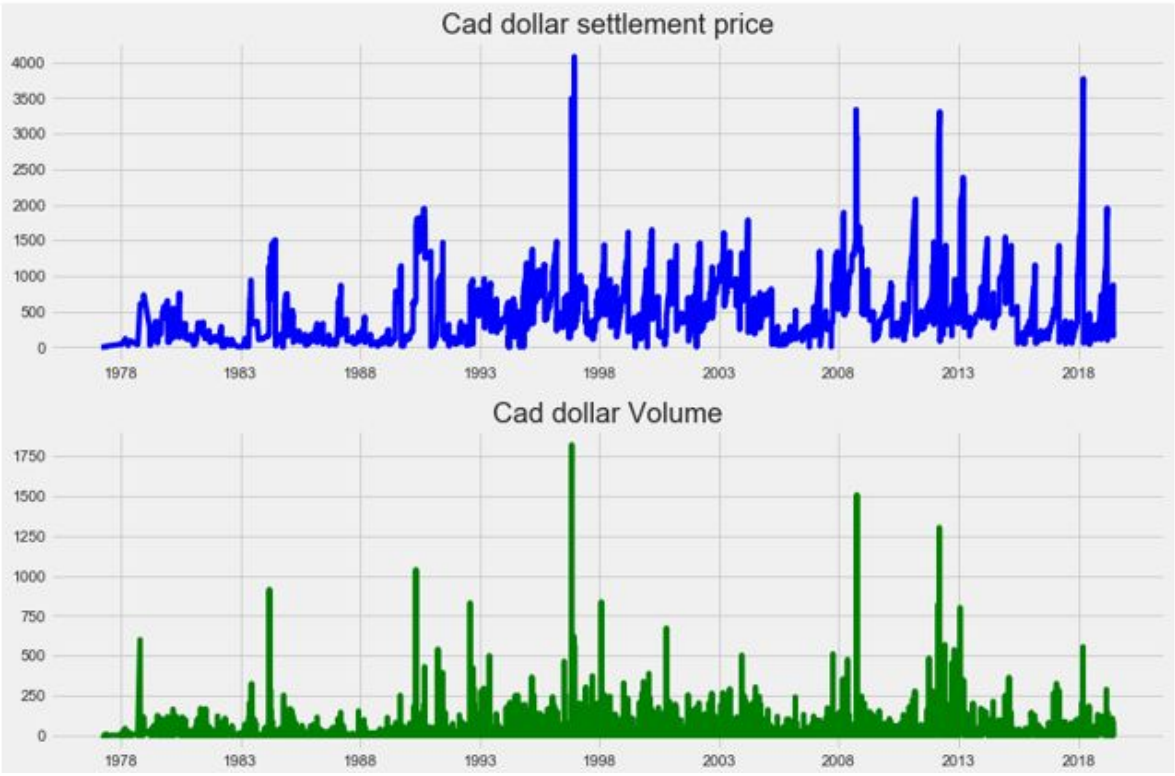
The same approach applies on Canadian Dollar as on Crude oil as the following plots shows



A closer look at the plot shows that the volume has a two big spikes between 2008 - 2010 and 2011 - 2013

what is the relation between the Opened Interest positions and Volume?
The following plots suggests a relationship between volume traded and Previous day open interest or in other words the yield paid to hold the currency. Which proves that Candian Dollar is a carry trade currency

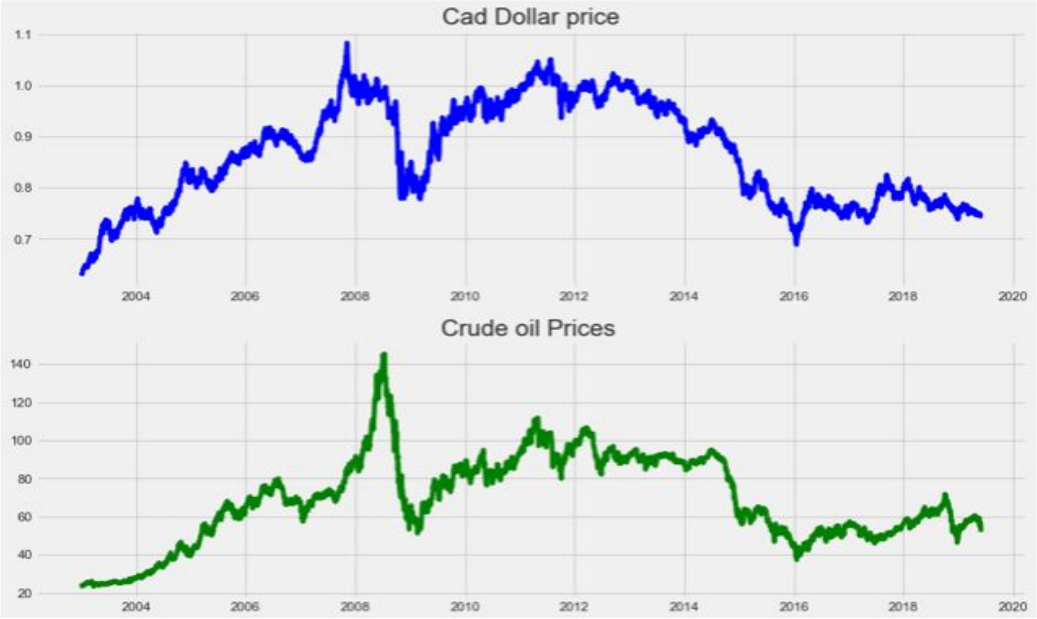
```
sns.set(rc={"figure.figsize": (12, 8)})
plt.style.use('fivethirtyeight')
plt.subplot(2,1,1)
plt.title('Cad dollar settlement price')
plt.plot(df_cadnew['Previous Day Open Interest'],color='blue')
plt.subplot(2,1,2)
plt.title('Cad dollar Volume')
plt.plot(df_cadnew['Volume'],color='green')
plt.tight_layout()
plt.show()
```



Fundamentally speaking we know that the price of commodities is related to supply and demand in the market
in the case of the Canadian/U.S. dollar exchange rate, the price is determined by the demand and supply of both Canadian dollars and U.S. dollars. Because crude oil exports account for a large portion of U.S. currency that's earned by Canada, movements in the price and the volume of crude oil have a significant impact on the flow of U.S. dollars into the Canadian economy. And the relation between prices and volume spikes or were shown in the above plots.

Now let us detect or compute the correlation of Cad to Crude oil.

```
# what is the releation between the Cad and Crude oi?(settle prices)
# Now let us compare the Crude oil and the Canadian dollar prices movements to investigate any releationship starting 2003
sns.set(rc={"figure.figsize": (12, 8)})
plt.style.use('fivethirtyeight')
plt.subplot(2,1,1)
plt.title('Cad Dollar price')
plt.plot(df_cadnew['Settle']['2003-01-01':'2019-05-31'],color='blue')
plt.subplot(2,1,2)
plt.title('Crude oil Prices ')
plt.plot(df_new['Settle']['2003-01-01':'2019-05-31'],color='green')
plt.tight_layout()
plt.show()
#The two data sets seems to be correlated follwoing the same trend.
```



The preliminary take away of this plot is that both Cad and Crude oil are correlated and move in the same direction up or down.
Computing the correlation

```
round(df_new['Settle'].corr(df_cadnew['Settle']),3)
```

```
0.816
```

Looking at the above plot and running or computing a correlation between the two

```
#creating oil_settle
oil_settle=df_new['Settle']['2003-02-07':'2019-05-31']
oil_settle.head()
```

```
Date
2003-02-07    25.43
2003-02-10    25.04
2003-02-11    25.26
2003-02-12    25.32
2003-02-13    25.53
Name: Settle, dtype: float64
```

```
#creating cad_settle
cad_settle=df_cadnew['Settle']['2003-01-01':'2019-05-31']
cad_settle.head()
```

```
Date
2003-01-02    0.6285
2003-01-03    0.6304
2003-01-06    0.6326
2003-01-07    0.6320
2003-01-08    0.6320
Name: Settle, dtype: float64
```

```
print(round(cad_settle.corr(oil_settle),4))
```

```
0.8996
```

So even during the period 2003-2019 the correlation is higher than the whole period rate with a rate of 0.899 versus 0.815.
It would be a good idea to plot Canadian Dollar against crude oil to visualize and detect the relationship between both data sets

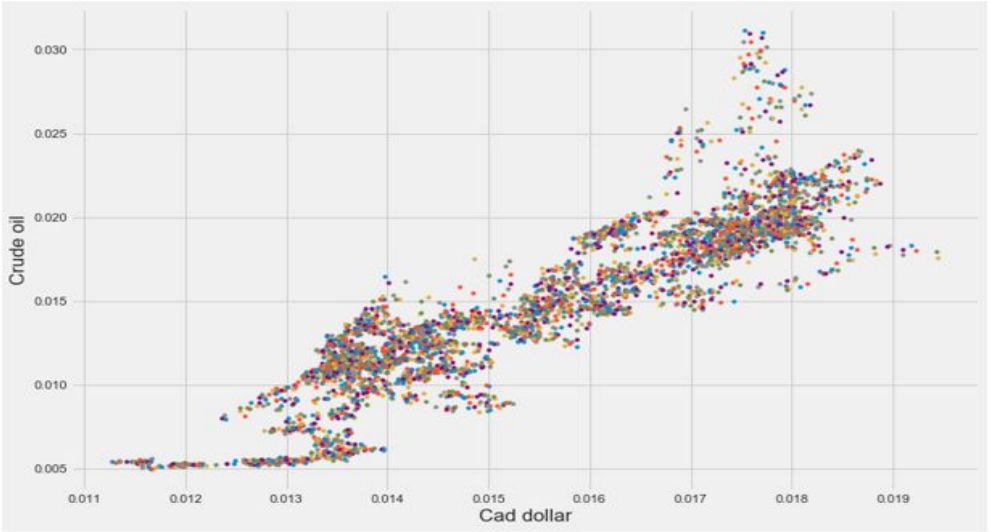
```
df_cadnew[['Settle']]['2003-01-01':'2019-05-31'].info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 4106 entries, 2003-01-02 to 2019-05-31
Data columns (total 1 columns):
Settle    4106 non-null float64
dtypes: float64(1)
memory usage: 64.2 KB
```

```
df_new[['Settle']]['2003-02-07':'2019-05-31'].info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 4106 entries, 2003-02-07 to 2019-05-31
Data columns (total 1 columns):
Settle    4106 non-null float64
dtypes: float64(1)
memory usage: 64.2 KB
```

```
sns.set(rc={"figure.figsize": (12, 8)})
plt.style.use('fivethirtyeight')
_=plt.plot(normalized_X,normalized_Y,marker='.',linestyle='none')
_=plt.plot(normalized_X,normalized_Y,color='r')
plt.xlabel('Cad dollar')
plt.ylabel('Crude oil')
plt.show()
```



Looking at the above plot would summarize the relationship between Canadian dollar and Crude oil. With a positive high correlation rate as crude oil increases the Canadian dollar increase and vice versa.

Now with that being noticed and computed let us try to change this from observation (based on a sample) to generalize this rule throughout Hypothesis Testing.

Since this Data set is a time series data so the independency of data is not a valid condition here so to do a test Hypothesis for the correlation we applied the Block bootstrap to overcome this problem.
First step is constructing a new data frame with both Crude oil Price and Canadian Dollar settlement price

```
df_cadnew.rename(columns={'Settle':'Settle'},inplace=True)# renaming the settelment as Cad
df_new.rename(columns={'Settle':'Crude'},inplace=True)# renaming the Settlement of crude as crude
```

```
df_crd=pd.concat([df_cadnew['Settle'],df_new['Crude']],axis=1)
df_crd.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 10237 entries, 1977-03-11 to 2019-06-28
Data columns (total 2 columns):
Settle    10135 non-null float64
Crude     9107 non-null float64
dtypes: float64(2)
memory usage: 239.9 KB
```

```
df_crd.tail()
```

Date	Settle	Crude
2019-06-24	0.76075	54.78
2019-06-25	0.76105	55.07
2019-06-26	0.76520	55.89
2019-06-27	0.76635	55.82
2019-06-28	0.76695	55.03

```
df_crd=df_crd.dropna() # Dropping the NAN from teh created dataframe
```

```
df_crd.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 9005 entries, 1983-04-07 to 2019-06-28
Data columns (total 2 columns):
Settle    9005 non-null float64
Crude     9005 non-null float64
dtypes: float64(2)
memory usage: 211.1 KB
```

Let us state a null hypothesis and an alternative hypothesis and run a Hypothesis testing using Block Bootstrap since the data is a time series data.
The null Hypothesis is that there is a positive correlation and at least 0.816 between Crude oil and Canadian Dollar.

The alternative Hypothesis is that there is no correlation of at least 0.816 between Crude oil and Canadian Dollar.

Since this Data set is a time series data so the independency of data is not a valid condition here so to perform Hypothesis testing for the correlation we applied the Block bootstrap to overcome this problem

First we introduced a block column to shuffle the data based on this column range.

```
df_crd['block'] = [item for sublist in [[i]* (9000 // 20) for i in range(1, 21)] for item in sublist]
```

```
df_crd.head()
```

Date	Settle	Crude	block
1983-04-07	0.8087	29.45	1
1983-04-08	0.8093	29.90	1
1983-04-11	0.8095	29.80	1
1983-04-12	0.8103	30.40	1
1983-04-13	0.8103	30.45	1

```
block_nums=list(range(1,20))
```

```
bootstrap_block_nums=np.random.choice(block_nums,size=20,replace=True)
```

```
df_crd1=pd.concat(df_crd[df_crd.block==i] for i in bootstrap_block_nums)
df_crd1
```

Date	Settle	Crude	block
2005-01-19	0.8178	43.40	13
2005-01-20	0.8135	43.15	13
2005-01-21	0.8219	44.01	13
2005-01-24	0.8201	44.27	13
2005-01-25	0.8119	45.04	13
2005-01-26	0.8140	44.61	13
2005-01-27	0.8109	44.48	13
2005-01-28	0.8098	43.60	13
2005-01-31	0.8096	44.47	13
2005-02-01	0.8120	44.08	13

We defined in this step a Pearson_r function to compute the correlation between crude oil and Canadian dollar

```
def pearson_r(x,y):
    corr_matrix=np.corrcoef(x,y)
    return corr_matrix[0,1]
```

```
r=pearson_r(Settle,crude)
r
```

```
0.8158645833047963
```

Then we sampled with replacement (Bootstrap) the new data frame df-crd1

```
import  random as random


perm_replicates=np.empty(10000)

for i in range(10000):

    cad_permuted=np.random.permutation(Settle)
    perm_replicates[i]=pearson_r(cad_permuted,crude)

p=np.sum(perm_replicates>=pearson_r(Settle,crude))/len(perm_replicates)
print('p-val= ',p)
```

```
p-val=  0.8187
```

Thus the above test shows that we have the probability of 0.8187 of getting a correlation rate greater than 0.816 the initial correlation of the observation

So there is not enough evidence to reject the Null Hypothesis

The next step is to predict the future price of crude oil and cascade that to Candian Dollar.