```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima_model import ARIMA
import datetime
import itertools
import warnings
from sklearn.metrics import mean_squared_error
import seaborn as sns
import statsmodels
plt.style.use('fivethirtyeight')
%matplotlib inline
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning:
  import pandas.util.testing as tm
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
df = pd.read_csv("/content/drive/MyDrive/Ineuron/data_stocks.csv")
df.head()
```

| | DATE | SP500 | NASDAQ.AAL | NASDAQ.AAPL | NASDAQ.ADBE | NASDAQ.ADI | NASDAQ.ADP |
|---|---|---|---|---|---|---|---|
| **0** | 1491226200 | 2363.6101 | 42.3300 | 143.6800 | 129.6300 | 82.040 | 102.2300 |
| **1** | 1491226260 | 2364.1001 | 42.3600 | 143.7000 | 130.3200 | 82.080 | 102.1400 |
| **2** | 1491226320 | 2362.6799 | 42.3100 | 143.6901 | 130.2250 | 82.030 | 102.2125 |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 0.0729 | 82.000 | 102.1400 |
| **4** | 1491226440 | 2364.8501 | 42.5378 | 143.6600 | 129.8800 | 82.035 | 102.0600 |

Cut 13 cells. You can now paste them in this or a different notebook. ×

5 rows × 502 columns

```
df["DATE"].dtypes
```

```
dtype('int64')
```

```
df['DATE'] = pd.to_datetime(df['DATE'], unit='s')
```

```
df['DATE'].tail()
```

```
41261   2017-08-31 19:56:00
41262   2017-08-31 19:57:00
41263   2017-08-31 19:58:00
41264   2017-08-31 19:59:00
41265   2017-08-31 20:00:00
Name: DATE, dtype: datetime64[ns]
```

```
df.index = df['DATE']
```

```
df.drop('DATE',axis = 1,inplace=True)
```

```
df.tail()
```

| | SP500 | NASDAQ.AAL | NASDAQ.AAPL | NASDAQ.ADBE | NASDAQ.ADI | NASDAQ.ADP | NASDAQ.A |
|---|---|---|---|---|---|---|---|
| **DATE** | | | | | | | |
| **2017-08-31 19:56:00** | 2472.22 | 44.72 | 164.11 | 155.090 | 83.67 | 106.565 | 11 |
| **2017-08-31 19:57:00** | 2471.77 | 44.73 | 164.12 | 155.160 | 83.65 | 106.590 | 11 |
| **2017-08-31 19:58:00** | 2470.03 | 44.74 | 164.01 | 155.065 | 83.62 | 106.520 | 11 |
| **2017-08-31 19:59:00** | 2471.49 | 44.71 | 163.88 | 154.960 | 83.58 | 106.400 | 11 |
| **2017-08-31 20:00:00** | 2471.49 | 44.74 | 163.98 | 155.160 | 83.69 | 106.470 | 11 |

Cut 13 cells. You can now paste them in this or a different notebook. ×

## NASDAQ.ADP

[ ] ↳ *19 cells hidden*

## NADBAQ.CBOE

```
df_CBOE = df[["NASDAQ.CBOE"]].copy()
df_CBOE.tail()
```

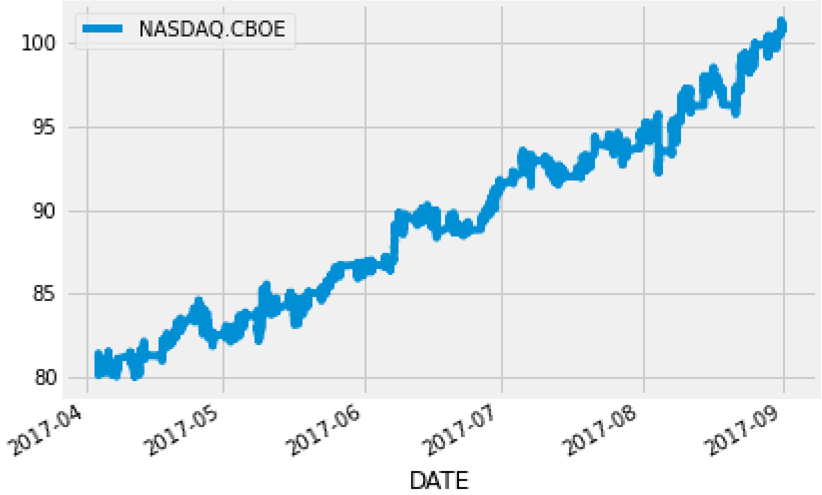
| | NASDAQ.CBOE |
|---|---|
| **DATE** | |
| **2017-08-31 19:56:00** | 100.89 |
| **2017-08-31 19:57:00** | 100.88 |
| **2017-08-31 19:58:00** | 100.86 |
| **2017-08-31 19:59:00** | 100.83 |
| **2017-08-31 20:00:00** | 100.89 |

```
df_CBOE.count()
```

```
NASDAQ.CBOE    41266
dtype: int64
```

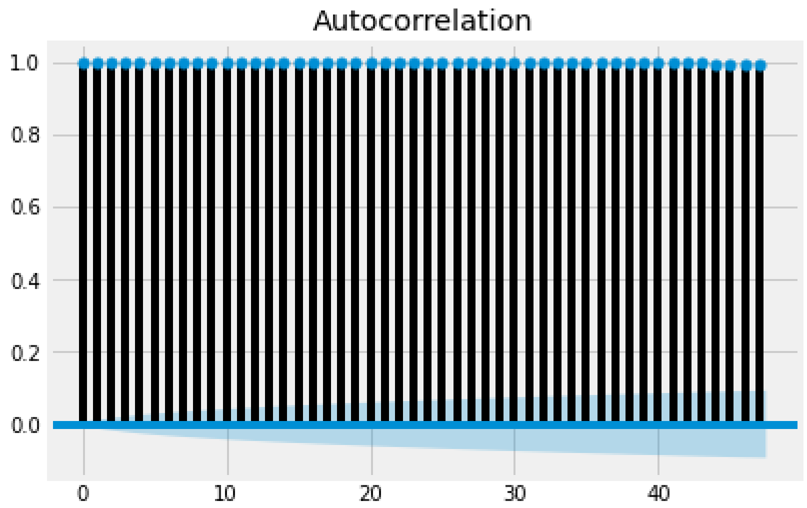
```
df_CBOE.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f98419c03c8>
```



Cut 13 cells. You can now paste them in this or a different notebook.

```
plot_acf(df_CBOE)
```



```
##Converting series to stationary
df_CBOE.shift(1)
```

|  | NASDAQ.CBOE |
|---|---|
| **DATE** |  |
| **2017-04-03 13:30:00** | NaN |
| **2017-04-03 13:31:00** | 81.0300 |
| **2017-04-03 13:32:00** | 81.2100 |
| **2017-04-03 13:33:00** | 81.2100 |
| **2017-04-03 13:34:00** | 81.1300 |
| **...** | ... |
| **2017-08-31 19:56:00** | 100.8899 |
| **2017-08-31 19:58:00** | 100.8800 |
| **2017-08-31 19:59:00** | 100.8600 |
| **2017-08-31 20:00:00** | 100.8300 |

Cut 13 cells. You can now paste them in this or a different notebook. ✕

41266 rows × 1 columns

```
X3 = df_CBOE.values
train3 = X3[0:28886] # 27 data as train data
test3 = X3[28886:] # 9 data as test data
print(train3.size)
print(test3.size)
predictions3 = []
```

```
28886
```

```
     12380
```

```
p3=d3=q3=range(0,2)
pdq3=list(itertools.product(p3,d3,q3))

warnings.filterwarnings('ignore')
for param in pdq3:
    try:
        model_arima3 = ARIMA(train3, order=param)
        model_arima_fit3 = model_arima3.fit()
        print(param,model_arima_fit3.aic)
    except:
        continue
```

```
     (0, 0, 0) 160441.4526311847
     (0, 0, 1) 120929.59019310356
     (0, 1, 0) -96706.67408377743
     (0, 1, 1) -96892.8965002238
     (1, 0, 0) -96696.92223399912
     (1, 0, 1) -96882.56874919325
     (1, 1, 0) -96878.58101700693
     (1, 1, 1) -97008.39440809148
```

```
from statsmodels.tsa.arima_model import ARIMA
model_arima3 = ARIMA(train3, order=(3,1,3))
model_arima_fit3 = model_arima3.fit()
```

```
predictions3 = model_arima_fit3.forecast(steps=12380)[0]
predictions3
```

```
     array([92.5017572 , 92.50262953, 92.5037775 , ..., 97.40717353,
            97.40756968, 97.40796582])
```

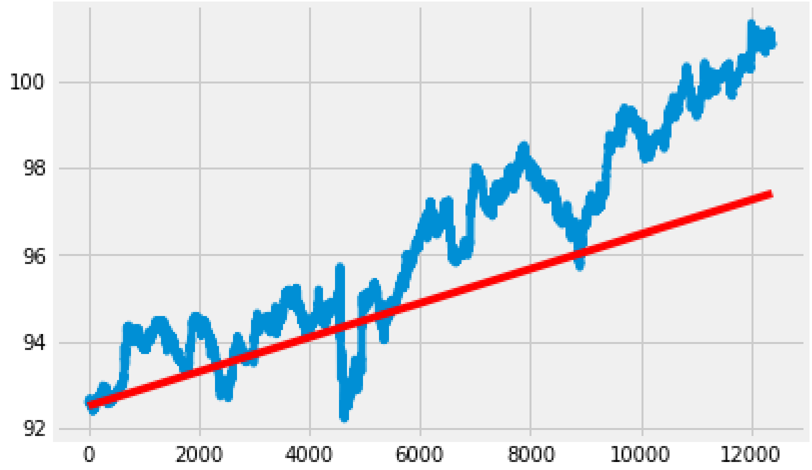
```
mse3 = round(mean_squared_error(test3,predictions3))
```

Cut 13 cells. You can now paste them in this or a different notebook. ✕

```
     3
```

```
plt.plot(test3)
plt.plot(predictions3, color='red')
```

```
[<matplotlib.lines.Line2D at 0x7f98416f6588>]
```



Cut 13 cells. You can now paste them in this or a different notebook. ✕