```
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
					0729	82.000	102.1400
4	1491226440	2364.8501	42.5378	143.6600	129.8800	82.035	102.0600

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ✕

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()
```

DATE	SP500	NASDAQ.AAL	NASDAQ.AAPL	NASDAQ.ADBE	NASDAQ.ADI	NASDAQ.ADP	NASDAQ.A
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

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ✕

NASDAQ.ADP

[ ] ↳ 21 cells hidden

NASDAQ.EBAY

```
df_EBAY = df[["NASDAQ.EBAY"]].copy()
```
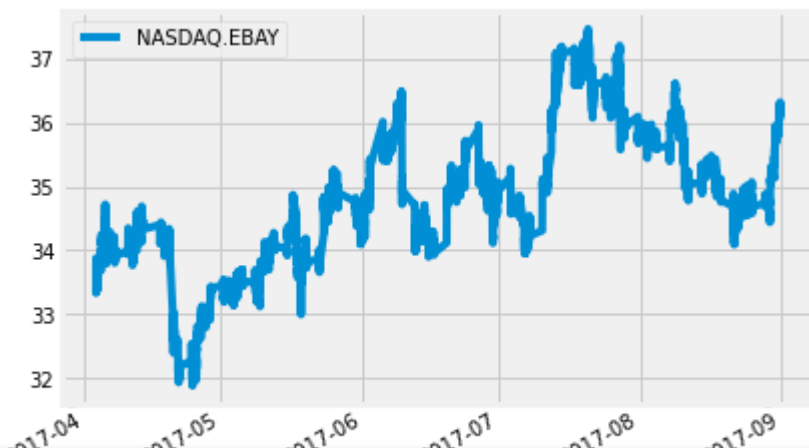
```
df_EBAY.tail()
```

| | NASDAQ.EBAY |
DATE	
2017-08-31 19:56:00	36.135
2017-08-31 19:57:00	36.130
2017-08-31 19:58:00	36.130
2017-08-31 19:59:00	36.120
2017-08-31 20:00:00	36.130

```
df_EBAY.count()
```

```
    NASDAQ.EBAY    41266
    dtype: int64
```
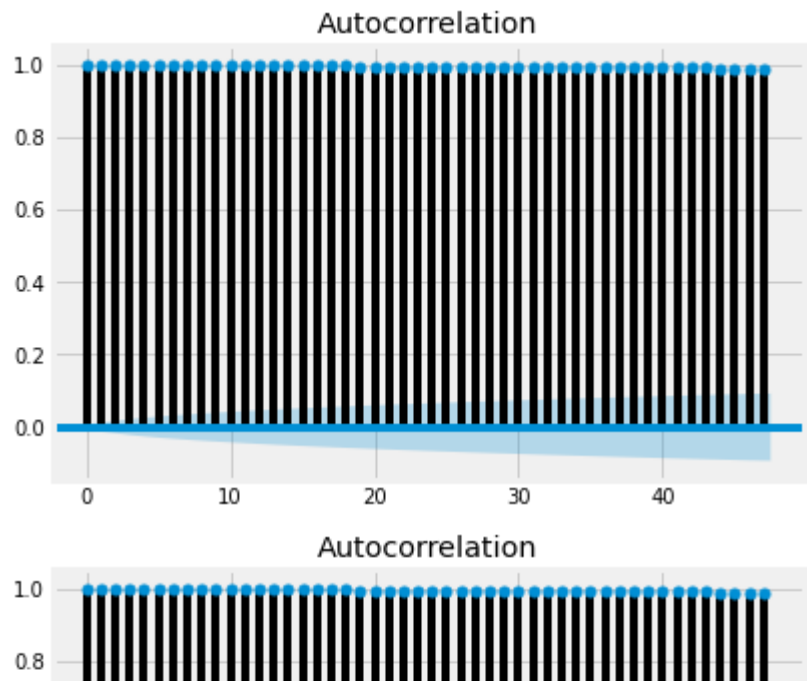
```
df_EBAY.plot()
```

```
    <matplotlib.axes._subplots.AxesSubplot at 0x7f984aa24e48>
```


To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ✕

```
from statsmodels.graphics.tsaplots import plot_acf
plot_acf(df_EBAY)
```

Autocorrelation

Autocorrelation


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

	NASDAQ.EBAY
DATE	
2017-04-03 13:30:00	NaN
2017-04-03 13:31:00	33.3975
2017-04-03 13:32:00	33.3950
2017-04-03 13:33:00	33.4100
2017-04-03 13:34:00	33.3350
...	...
2017-08-31 19:57:00	36.1350
2017-08-31 19:58:00	36.1300
2017-08-31 19:59:00	36.1300
2017-08-31 20:00:00	36.1200

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ✕

41266 rows × 1 columns

```
X5 = df_EBAY.values
train5 = X5[0:28886] # 27 data as train data
test5 = X5[28886:] # 9 data as test data
print(train5.size)
print(test5.size)
predictions5 = []
```

```
      28886
      12380
```

```
p5=d5=q5=range(0,2)
pdq5=list(itertools.product(p5,d5,q5))

warnings.filterwarnings('ignore')
for param in pdq5:
    try:
        model_arima5 = ARIMA(train5, order=param)
        model_arima_fit5 = model_arima5.fit()
        print(param,model_arima_fit5.aic)
    except:
        continue
```

```
  (0, 0, 0) 83955.30612486275
  (0, 0, 1) 44870.471253968884
  (0, 1, 0) -135861.67620322717
  (0, 1, 1) -135872.0650324924
  (1, 0, 0) -135857.4009523872
  (1, 0, 1) -135867.62575643833
  (1, 1, 0) -135872.54671056976
  (1, 1, 1) -135876.14912516676
```

```
from statsmodels.tsa.arima_model import ARIMA
model_arima5 = ARIMA(train5, order=(4,1,4))
model_arima_fit5 = model_arima5.fit()
```

```
predictions5 = model_arima_fit5.forecast(steps=12380)[0]
predictions5
```

```
  array([37.01556121, 37.01587908, 37.01520994, ..., 38.55472764,
         38.55485247, 38.5549773 ])
```

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu   ✕

```
res5
```

```
  6
```

```
plt.plot(test5)
plt.plot(predictions5, color='red')
```
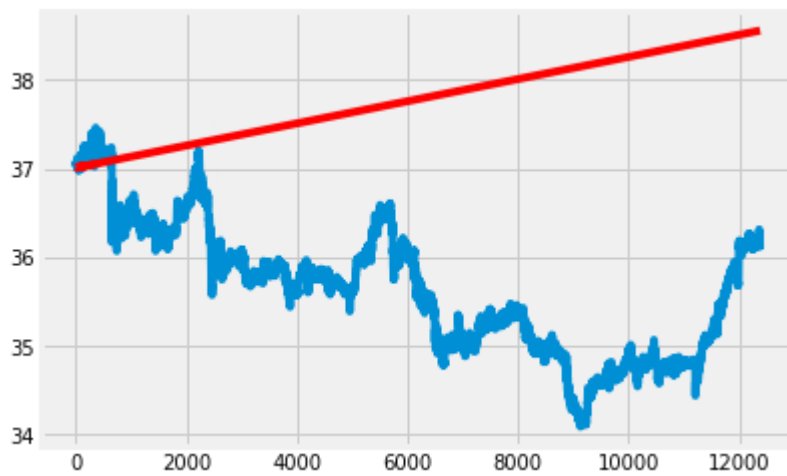
⤷

```
[<matplotlib.lines.Line2D at 0x7f9841a9c630>]
```


To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ✕