```
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
3	1491226380	2364.3101	42.3700	143.6400	130.0729	82.000	102.1400
4	1491226440	2364.8501	42.5378	143.6600	129.8800	82.035	102.0600

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

5 rows × 501 columns

NASDAQ.AAPL

```
df_AAPL = df[["NASDAQ.AAPL"]].copy()
df_AAPL.tail()
```

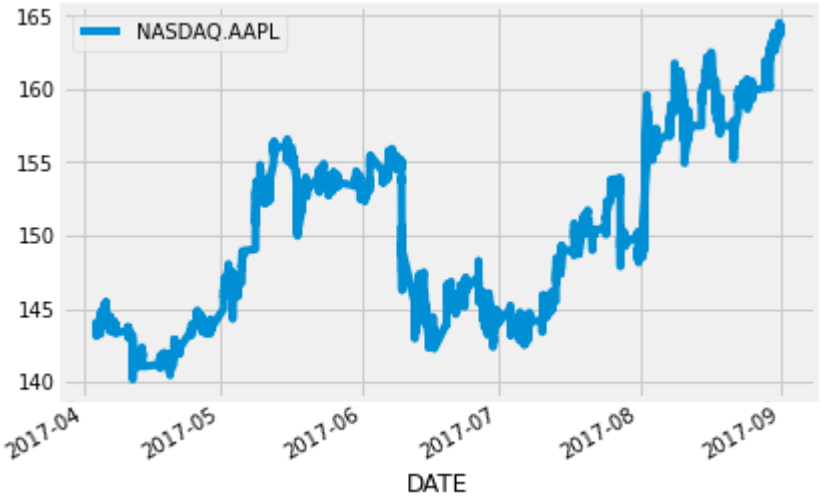
	NASDAQ.AAPL
DATE	
2017-08-31 19:56:00	164.11
2017-08-31 19:57:00	164.12

```
df_AAPL.count()
```

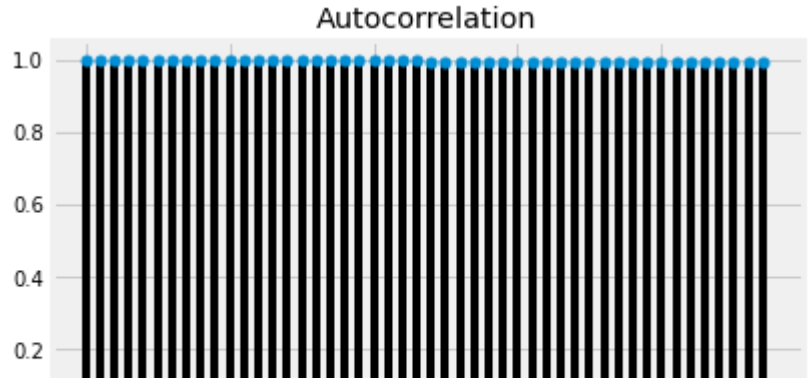
```
NASDAQ.AAPL    41266
dtype: int64
```

```
df_AAPL.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f98410597f0>
```




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




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

DATE	NASDAQ.AAPL
2017-04-03 13:30:00	NaN
2017-04-03 13:31:00	143.6800
2017-04-03 13:32:00	143.7000
2017-04-03 13:33:00	143.6901
2017-04-03 13:34:00	143.6400
...	...
2017-08-31 19:56:00	164.1400
2017-08-31 19:57:00	164.1100
2017-08-31 19:58:00	164.1200
2017-08-31 19:59:00	164.0100
2017-08-31 20:00:00	163.8800

41266 rows × 1 columns

```
X0 = df_AAPL.values
train0 = X0[0:28886] # 27 data as train data
test0 = X0[28886:] # 9 data as test data
print(train0.size)
print(test0.size)
predictions0 = []
```

```
28886
12380
```

```
p0=d0=q0=range(0,2)
pdq0=list(itertools.product(p0,d0,q0))

warnings.filterwarnings('ignore')
```

```
warnings.filterwarnings("ignore")
for param in pdq0:
    try:
        model_arima0 = ARIMA(train0, order=param)
        model_arima_fit0 = model_arima0.fit()
        print(param,model_arima_fit0.aic)
    except:
        continue
```

```
(0, 0, 0) 170326.9720446082
(0, 0, 1) 131018.03599865251
(0, 1, 0) -64708.712006361384
(0, 1, 1) -64706.8773153409
(1, 0, 0) -64703.106010175194
(1, 0, 1) -64701.25996864913
(1, 1, 0) -64706.87029558887
(1, 1, 1) -64714.63157446154
```

```
from statsmodels.tsa.arima_model import ARIMA
model_arima0 = ARIMA(train0, order=(4,1,4))
model_arima_fit0 = model_arima0.fit()
```

```
#p0,d0,q0
#p0 -> Periods taken for auto regressive model
#d0 -> Integrated order, difference
#q0 -> Periods in moving average model
from statsmodels.tsa.arima_model import ARIMA
model_arima0 = ARIMA(train0, order=(3,1,3))
model_arima_fit0 = model_arima0.fit()
print(model_arima_fit0.aic)
```

```
-64730.23498583691
```

```
predictions0 = model_arima_fit0.forecast(steps=12380)[0]
predictions0
```

```
array([150.61066006, 150.61168229, 150.61211528, ..., 153.58143875,
       153.58167866, 153.58191858])
```

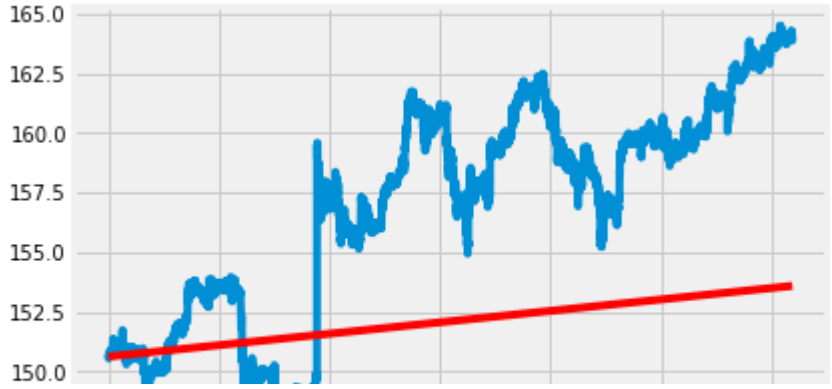
```
res0 = round(mean_squared_error(test0,predictions0))
res0
```

```
38
```

+ Code + Text

```
plt.plot(test0)
plt.plot(predictions0, color='red')
```

```
[<matplotlib.lines.Line2D at 0x7f9840e3c438>]
```



NASDAQ.ADP

↳ *19 cells hidden*

NADBAQ.CBOE

[ ] ↳ *14 cells hidden*

NASDAQ.CSCO

[ ] ↳ *13 cells hidden*

NASDAQ.EBAY

[ ] ↳ *12 cells hidden*