

```
In [1]: !pip install nbconvert
```

```
Requirement already satisfied: nbconvert in f:\anaconda3\lib\site-packages (6.0.7)
Requirement already satisfied: nbclient<0.6.0,>=0.5.0 in f:\anaconda3\lib\site-packages (from nbconvert) (0.5.3)
Requirement already satisfied: jupyterlab-pygments in f:\anaconda3\lib\site-packages (from nbconvert) (0.1.2)
Requirement already satisfied: Jinja2>=2.4 in f:\anaconda3\lib\site-packages (from nbconvert) (2.11.3)
Requirement already satisfied: Pygments>=2.4.1 in f:\anaconda3\lib\site-packages (from nbconvert) (2.8.1)
Requirement already satisfied: nbformat>=4.4 in f:\anaconda3\lib\site-packages (from nbconvert) (5.1.3)
Requirement already satisfied: pandocfilters>=1.4.1 in f:\anaconda3\lib\site-packages (from nbconvert) (1.4.3)
Requirement already satisfied: mistune<2,>=0.8.1 in f:\anaconda3\lib\site-packages (from nbconvert) (0.8.4)
Requirement already satisfied: defusedxml in f:\anaconda3\lib\site-packages (from nbconvert) (0.7.1)
Requirement already satisfied: bleach in f:\anaconda3\lib\site-packages (from nbconvert) (3.3.0)
Requirement already satisfied: traitlets>=4.2 in f:\anaconda3\lib\site-packages (from nbconvert) (5.0.5)
Requirement already satisfied: testpath in f:\anaconda3\lib\site-packages (from nbconvert) (0.4.4)
Requirement already satisfied: entrypoints>=0.2.2 in f:\anaconda3\lib\site-packages (from nbconvert) (0.3)
Requirement already satisfied: jupyter-core in f:\anaconda3\lib\site-packages (from nbconvert) (4.7.1)
Requirement already satisfied: MarkupSafe>=0.23 in f:\anaconda3\lib\site-packages (from Jinja2>=2.4->nbconvert) (1.1.1)
Requirement already satisfied: nest-asyncio in f:\anaconda3\lib\site-packages (from nbclient<0.6.0,>=0.5.0->nbconvert) (1.5.1)
Requirement already satisfied: jupyter-client>=6.1.5 in f:\anaconda3\lib\site-packages (from nbclient<0.6.0,>=0.5.0->nbconvert) (6.1.12)
Requirement already satisfied: async-generator in f:\anaconda3\lib\site-packages (from nbclient<0.6.0,>=0.5.0->nbconvert) (1.10)
Requirement already satisfied: pyzmq>=13 in f:\anaconda3\lib\site-packages (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (20.0.0)
Requirement already satisfied: python-dateutil>=2.1 in f:\anaconda3\lib\site-packages (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (2.8.1)
Requirement already satisfied: tornado>=4.1 in f:\anaconda3\lib\site-packages (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (6.1)
Requirement already satisfied: pywin32>=1.0 in f:\anaconda3\lib\site-packages (from jupyter-core->nbconvert) (227)
Requirement already satisfied: ipython-genutils in f:\anaconda3\lib\site-packages (from nbformat>=4.4->nbconvert) (0.2.0)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in f:\anaconda3\lib\site-packages (from nbformat>=4.4->nbconvert) (3.2.0)
Requirement already satisfied: pyparsing>=2.4.0 in f:\anaconda3\lib\site-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert) (2.4.7)
Requirement already satisfied: setuptools in f:\anaconda3\lib\site-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert) (52.0.0.post20210125)
Requirement already satisfied: attrs>=17.4.0 in f:\anaconda3\lib\site-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert) (20.3.0)
Requirement already satisfied: six>=1.11.0 in f:\anaconda3\lib\site-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert) (1.15.0)
Requirement already satisfied: packaging in f:\anaconda3\lib\site-packages (from bleach->nbconvert) (20.9)
Requirement already satisfied: webencodings in f:\anaconda3\lib\site-packages (from bleach->nbconvert) (0.5.1)
Requirement already satisfied: pyparsing>=2.0.2 in f:\anaconda3\lib\site-packages (from packaging->bleach->nbconvert) (2.4.7)
```

In []:

In [2]:

```
#install Quandl
!pip install quandl
```

```
Requirement already satisfied: quandl in f:\anaconda3\lib\site-packages (3.7.0)
Requirement already satisfied: numpy>=1.8 in f:\anaconda3\lib\site-packages (from quandl) (1.20.1)
Requirement already satisfied: pandas>=0.14 in f:\anaconda3\lib\site-packages (from quandl) (1.2.4)
Requirement already satisfied: requests>=2.7.0 in f:\anaconda3\lib\site-packages (from quandl) (2.25.1)
Requirement already satisfied: six in f:\anaconda3\lib\site-packages (from quandl) (1.15.0)
Requirement already satisfied: python-dateutil in f:\anaconda3\lib\site-packages (from quandl) (2.8.1)
Requirement already satisfied: inflection>=0.3.1 in f:\anaconda3\lib\site-packages (from quandl) (0.5.1)
Requirement already satisfied: more-itertools in f:\anaconda3\lib\site-packages (from quandl) (8.7.0)
Requirement already satisfied: pytz>=2017.3 in f:\anaconda3\lib\site-packages (from pandas>=0.14->quandl) (2021.1)
Requirement already satisfied: idna<3,>=2.5 in f:\anaconda3\lib\site-packages (from requests>=2.7.0->quandl) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in f:\anaconda3\lib\site-packages (from requests>=2.7.0->quandl) (1.26.4)
Requirement already satisfied: chardet<5,>=3.0.2 in f:\anaconda3\lib\site-packages (from requests>=2.7.0->quandl) (4.0.0)
Requirement already satisfied: certifi>=2017.4.17 in f:\anaconda3\lib\site-packages (from requests>=2.7.0->quandl) (2020.12.5)
```

In [3]:

```
#IMPORTING NECESSARY LIBRARY
import quandl as qdl
import pandas as pd
import numpy as np
```

In [5]:

```
#GETTING KEY
qdl.ApiConfig.api_key = ""
```

In [66]:

```
#SELECTING STOCK FROM DEFINED DATES
stock=qdl.get('NSE/PNB',start_date='2018-01-01',end_date='2018-01-31')
```

In [7]:

```
print(stock)
```

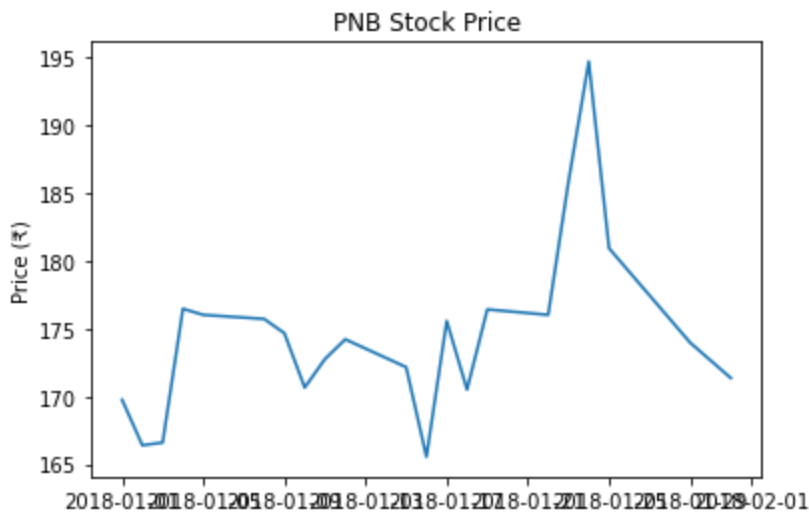
Date	Open	High	Low	Last	Close	Total Trade Quantity \
2018-01-01	172.95	173.40	168.90	169.70	169.75	7869149.0
2018-01-02	170.50	170.90	165.40	166.15	166.40	13729132.0
2018-01-03	168.00	169.30	166.20	166.50	166.60	8534233.0
2018-01-04	167.35	179.70	165.40	178.00	176.45	31613344.0
2018-01-05	180.40	180.50	175.40	176.05	176.00	15500426.0
2018-01-08	177.05	178.15	175.10	175.65	175.70	7302294.0
2018-01-09	175.30	177.90	174.05	174.55	174.65	11152776.0
2018-01-10	174.45	174.90	170.00	170.10	170.65	9021131.0
2018-01-11	170.30	173.60	169.55	173.30	172.75	8683986.0
2018-01-12	173.50	174.90	171.45	174.50	174.20	11223630.0
2018-01-15	174.95	175.85	171.85	172.05	172.15	7104648.0
2018-01-16	172.25	172.95	165.10	165.70	165.55	11916843.0
2018-01-17	165.50	177.00	162.65	176.50	175.55	25299752.0
2018-01-18	170.70	182.75	168.65	170.35	170.50	32923946.0

2018-01-19	170.80	178.00	169.35	176.90	176.40	21519974.0
2018-01-22	176.10	177.30	172.20	176.20	176.00	11370450.0
2018-01-23	176.05	187.80	175.25	186.00	185.80	28176767.0
2018-01-24	185.25	196.35	182.10	195.45	194.65	50753260.0
2018-01-25	197.00	197.00	180.00	181.40	180.90	45808402.0
2018-01-29	181.70	182.35	172.60	173.00	173.95	18481283.0
2018-01-30	173.30	175.90	171.40	172.25	172.65	16914910.0
2018-01-31	171.50	175.20	168.80	170.90	171.35	15313331.0

Date	Turnover (Lacs)
2018-01-01	13480.32
2018-01-02	22963.83
2018-01-03	14308.85
2018-01-04	54653.03
2018-01-05	27508.99
2018-01-08	12881.70
2018-01-09	19639.04
2018-01-10	15507.28
2018-01-11	14922.25
2018-01-12	19450.16
2018-01-15	12340.18
2018-01-16	20066.79
2018-01-17	43070.74
2018-01-18	58071.00
2018-01-19	37574.06
2018-01-22	19896.68
2018-01-23	51344.08
2018-01-24	96318.97
2018-01-25	84712.34
2018-01-29	32703.52
2018-01-30	29341.51
2018-01-31	26320.88

```
In [8]: #LIBRARY FOR VISUALIZATION
import seaborn as sns
import matplotlib.pyplot as plt
```

```
In [9]: #plot to show the closing price of stock
plt.plot(stock.index, stock['Close'])
plt.title('PNB Stock Price')
plt.ylabel('Price (₹)');
plt.show()
plt.figure(figsize=(2000,10))
```



Now Converting into DataFrame for Further Analysis

```
In [10]: df=pd.DataFrame(stock)
```

```
In [11]: df
```

```
Out[11]:
```

	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
Date							
2018-01-01	172.95	173.40	168.90	169.70	169.75	7869149.0	13480.32
2018-01-02	170.50	170.90	165.40	166.15	166.40	13729132.0	22963.83
2018-01-03	168.00	169.30	166.20	166.50	166.60	8534233.0	14308.85
2018-01-04	167.35	179.70	165.40	178.00	176.45	31613344.0	54653.03
2018-01-05	180.40	180.50	175.40	176.05	176.00	15500426.0	27508.99
2018-01-08	177.05	178.15	175.10	175.65	175.70	7302294.0	12881.70
2018-01-09	175.30	177.90	174.05	174.55	174.65	11152776.0	19639.04
2018-01-10	174.45	174.90	170.00	170.10	170.65	9021131.0	15507.28
2018-01-11	170.30	173.60	169.55	173.30	172.75	8683986.0	14922.25
2018-01-12	173.50	174.90	171.45	174.50	174.20	11223630.0	19450.16
2018-01-15	174.95	175.85	171.85	172.05	172.15	7104648.0	12340.18
2018-01-16	172.25	172.95	165.10	165.70	165.55	11916843.0	20066.79
2018-01-17	165.50	177.00	162.65	176.50	175.55	25299752.0	43070.74
2018-01-18	180.70	182.75	168.65	170.35	170.50	32923946.0	58071.00
2018-01-19	170.80	178.00	169.35	176.90	176.40	21519974.0	37574.06
2018-01-22	176.10	177.30	172.20	176.20	176.00	11370450.0	19896.68
2018-01-23	176.05	187.80	175.25	186.00	185.80	28176767.0	51344.08
2018-01-24	185.25	196.35	182.10	195.45	194.65	50753260.0	96318.97
2018-01-25	197.00	197.00	180.00	181.40	180.90	45808402.0	84712.34
2018-01-29	181.70	182.35	172.60	173.00	173.95	18481283.0	32703.52
2018-01-30	173.30	175.90	171.40	172.25	172.65	16914910.0	29341.51
2018-01-31	171.50	175.20	168.80	170.90	171.35	15313331.0	26320.88

```
In [12]: #Converting Into CSV file
df.to_csv("PNB_csv")
```

```
In [13]: data=pd.read_csv("PNB_csv")
```

```
In [14]: data
```

Out[14]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-01-01	172.95	173.40	168.90	169.70	169.75	7869149.0	13480.32
1	2018-01-02	170.50	170.90	165.40	166.15	166.40	13729132.0	22963.83
2	2018-01-03	168.00	169.30	166.20	166.50	166.60	8534233.0	14308.85
3	2018-01-04	167.35	179.70	165.40	178.00	176.45	31613344.0	54653.03
4	2018-01-05	180.40	180.50	175.40	176.05	176.00	15500426.0	27508.99
5	2018-01-08	177.05	178.15	175.10	175.65	175.70	7302294.0	12881.70
6	2018-01-09	175.30	177.90	174.05	174.55	174.65	11152776.0	19639.04
7	2018-01-10	174.45	174.90	170.00	170.10	170.65	9021131.0	15507.28
8	2018-01-11	170.30	173.60	169.55	173.30	172.75	8683986.0	14922.25
9	2018-01-12	173.50	174.90	171.45	174.50	174.20	11223630.0	19450.16
10	2018-01-15	174.95	175.85	171.85	172.05	172.15	7104648.0	12340.18
11	2018-01-16	172.25	172.95	165.10	165.70	165.55	11916843.0	20066.79
12	2018-01-17	165.50	177.00	162.65	176.50	175.55	25299752.0	43070.74
13	2018-01-18	180.70	182.75	168.65	170.35	170.50	32923946.0	58071.00
14	2018-01-19	170.80	178.00	169.35	176.90	176.40	21519974.0	37574.06
15	2018-01-22	176.10	177.30	172.20	176.20	176.00	11370450.0	19896.68
16	2018-01-23	176.05	187.80	175.25	186.00	185.80	28176767.0	51344.08
17	2018-01-24	185.25	196.35	182.10	195.45	194.65	50753260.0	96318.97
18	2018-01-25	197.00	197.00	180.00	181.40	180.90	45808402.0	84712.34
19	2018-01-29	181.70	182.35	172.60	173.00	173.95	18481283.0	32703.52
20	2018-01-30	173.30	175.90	171.40	172.25	172.65	16914910.0	29341.51
21	2018-01-31	171.50	175.20	168.80	170.90	171.35	15313331.0	26320.88

In [67]:

```
#INFO OF DATAFRAME
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 22 entries, 2018-01-01 to 2018-01-31
Data columns (total 7 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Open                                  22 non-null     float64
1   High                                  22 non-null     float64
2   Low                                   22 non-null     float64
3   Last                                  22 non-null     float64
4   Close                                 22 non-null     float64
5   Total Trade Quantity                 22 non-null     float64
6   Turnover (Lacs)                      22 non-null     float64
dtypes: float64(7)
memory usage: 1.4 KB
```

Checking the Null Values

In [68]:

```
df.describe()
```

Out[68]:

	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
count	22.000000	22.000000	22.000000	22.000000	22.000000	2.200000e+01	22.000000
mean	177.13636	177.13636	174.97272	174.60000	174.48181	1.864608e+07	33048.91818

	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
std	6.860267	7.114442	4.799324	6.710582	6.447490	1.237097e+07	23221.962151
min	165.500000	169.300000	162.650000	165.700000	165.550000	7.104648e+06	12340.180000
25%	170.975000	174.900000	168.687500	170.487500	170.825000	9.554042e+06	16493.000000
50%	173.975000	177.150000	170.700000	173.900000	174.075000	1.452123e+07	24642.355000
75%	176.812500	180.300000	173.687500	176.425000	176.000000	2.435481e+07	41696.570000
max	197.000000	197.000000	182.100000	195.450000	194.650000	5.075326e+07	96318.970000

In [15]:

```
#CHECKING THE NULL VALUES
data.isnull()
```

Out[15]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False	False
10	False	False	False	False	False	False	False	False
11	False	False	False	False	False	False	False	False
12	False	False	False	False	False	False	False	False
13	False	False	False	False	False	False	False	False
14	False	False	False	False	False	False	False	False
15	False	False	False	False	False	False	False	False
16	False	False	False	False	False	False	False	False
17	False	False	False	False	False	False	False	False
18	False	False	False	False	False	False	False	False
19	False	False	False	False	False	False	False	False
20	False	False	False	False	False	False	False	False
21	False	False	False	False	False	False	False	False

In [16]:

```
data.isnull().sum()
```

Out[16]:

Date	0
Open	0
High	0
Low	0
Last	0
Close	0
Total Trade Quantity	0

Turnover (Lacs) 0
dtype: int64

There is no null values in this dataset.

```
In [17]: #CORRELATION
data.corr()
```

```
Out[17]:
```

	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
Open	1.000000	0.814953	0.824666	0.468985	0.498857	0.563297	0.590022
High	0.814953	1.000000	0.773203	0.839457	0.848900	0.875591	0.892736
Low	0.824666	0.773203	1.000000	0.713611	0.746765	0.432021	0.473129
Last	0.468985	0.839457	0.713611	1.000000	0.997320	0.732477	0.753342
Close	0.498857	0.848900	0.746765	0.997320	1.000000	0.720901	0.743690
Total Trade Quantity	0.563297	0.875591	0.432021	0.732477	0.720901	1.000000	0.998559
Turnover (Lacs)	0.590022	0.892736	0.473129	0.753342	0.743690	0.998559	1.000000

```
In [18]: #PLOTTING CORRELATION
plt.figure(figsize=(10,10))
sns.heatmap(data.corr(),annot=True,cmap="crest",center=0)
plt.show()
```



1)Open is a DEPENDENT VARIABLE 2)while the rest are INDEPENDENT VARIABLE 3)DATE coloumn Not required For the analysis

```
In [19]: #Date Coloumn is to be removed from the dataset
x=data.loc[:, "High": "Turnover (Lacs)"].values
```

```
In [20]: data.loc[:, "High": "Turnover (Lacs)"]
```

	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	173.40	168.90	169.70	169.75	7869149.0	13480.32
1	170.90	165.40	166.15	166.40	13729132.0	22963.83
2	169.30	166.20	166.50	166.60	8534233.0	14308.85
3	179.70	165.40	178.00	176.45	31613344.0	54653.03
4	180.50	175.40	176.05	176.00	15500426.0	27508.99
5	178.15	175.10	175.65	175.70	7302294.0	12881.70
6	177.00	174.05	174.55	174.65	11152776.0	19639.04

	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
7	174.90	170.00	170.10	170.65	9021131.0	15507.28
8	173.60	169.55	173.30	172.75	8683986.0	14922.25
9	174.90	171.45	174.50	174.20	11223630.0	19450.16
10	175.85	171.85	172.05	172.15	7104648.0	12340.18
11	172.95	165.10	165.70	165.55	11916843.0	20066.79
12	177.00	162.65	176.50	175.55	25299752.0	43070.74
13	182.75	168.65	170.35	170.50	32923946.0	58071.00
14	178.00	169.35	176.90	176.40	21519974.0	37574.06
15	177.30	172.20	176.20	176.00	11370450.0	19896.68
16	187.80	175.25	186.00	185.80	28176767.0	51344.08
17	196.35	182.10	195.45	194.65	50753260.0	96318.97
18	197.00	180.00	181.40	180.90	45808402.0	84712.34
19	182.35	172.60	173.00	173.95	18481283.0	32703.52
20	175.90	171.40	172.25	172.65	16914910.0	29341.51
21	175.20	168.80	170.90	171.35	15313331.0	26320.88

In [21]:

x

Out[21]:

```
array([[1.73400000e+02, 1.68900000e+02, 1.69700000e+02, 1.69750000e+02,
        7.8691490e+06, 1.3480320e+04],
       [1.70900000e+02, 1.65400000e+02, 1.66150000e+02, 1.66400000e+02,
        1.3729132e+07, 2.2963830e+04],
       [1.69300000e+02, 1.66200000e+02, 1.66500000e+02, 1.66600000e+02,
        8.5342330e+06, 1.4308850e+04],
       [1.79700000e+02, 1.65400000e+02, 1.78000000e+02, 1.76450000e+02,
        3.1613344e+07, 5.4653030e+04],
       [1.80500000e+02, 1.75400000e+02, 1.76050000e+02, 1.76000000e+02,
        1.5500426e+07, 2.7508990e+04],
       [1.78150000e+02, 1.75100000e+02, 1.75650000e+02, 1.75700000e+02,
        7.3022940e+06, 1.2881700e+04],
       [1.77900000e+02, 1.74050000e+02, 1.74550000e+02, 1.74650000e+02,
        1.1152776e+07, 1.9639040e+04],
       [1.74900000e+02, 1.70000000e+02, 1.70100000e+02, 1.70650000e+02,
        9.0211310e+06, 1.5507280e+04],
       [1.73600000e+02, 1.69550000e+02, 1.73300000e+02, 1.72750000e+02,
        8.6839860e+06, 1.4922250e+04],
       [1.74900000e+02, 1.71450000e+02, 1.74500000e+02, 1.74200000e+02,
        1.1223630e+07, 1.9450160e+04],
       [1.75850000e+02, 1.71850000e+02, 1.72050000e+02, 1.72150000e+02,
        7.1046480e+06, 1.2340180e+04],
       [1.72950000e+02, 1.65100000e+02, 1.65700000e+02, 1.65550000e+02,
        1.1916843e+07, 2.0066790e+04],
       [1.77000000e+02, 1.62650000e+02, 1.76500000e+02, 1.75550000e+02,
        2.5299752e+07, 4.3070740e+04],
       [1.82750000e+02, 1.68650000e+02, 1.70350000e+02, 1.70500000e+02,
        3.2923946e+07, 5.8071000e+04],
       [1.78000000e+02, 1.69350000e+02, 1.76900000e+02, 1.76400000e+02,
        2.1519974e+07, 3.7574060e+04],
       [1.77300000e+02, 1.72200000e+02, 1.76200000e+02, 1.76000000e+02,
        1.1370450e+07, 1.9896680e+04],
       [1.87800000e+02, 1.75250000e+02, 1.86000000e+02, 1.85800000e+02,
        2.8176767e+07, 5.1344080e+04],
       [1.96350000e+02, 1.82100000e+02, 1.95450000e+02, 1.94650000e+02,
        5.0753260e+07, 9.6318970e+04],
       [1.87000000e+02, 1.80000000e+02, 1.81400000e+02, 1.80900000e+02,
```

```
4.5808402e+07, 8.4712340e+04],
[1.8235000e+02, 1.7260000e+02, 1.7300000e+02, 1.7395000e+02,
1.8481283e+07, 3.2703520e+04],
[1.7590000e+02, 1.7140000e+02, 1.7225000e+02, 1.7265000e+02,
1.6914910e+07, 2.9341510e+04],
[1.7520000e+02, 1.6880000e+02, 1.7090000e+02, 1.7135000e+02,
1.5313331e+07, 2.6320880e+04]])
```

```
In [22]: #Dependent variable
y=data.loc[:, "Open"].values
```

```
In [23]: data.loc[:, "Open"]
```

```
Out[23]: 0      172.95
1      170.50
2      168.00
3      167.35
4      180.40
5      177.05
6      175.30
7      174.45
8      170.30
9      173.50
10     174.95
11     172.25
12     165.50
13     180.70
14     170.80
15     176.10
16     176.05
17     185.25
18     197.00
19     181.70
20     173.30
21     171.50
Name: Open, dtype: float64
```

```
In [24]: y
```

```
Out[24]: array([172.95, 170.5 , 168.  , 167.35, 180.4 , 177.05, 175.3 , 174.45,
170.3 , 173.5 , 174.95, 172.25, 165.5 , 180.7 , 170.8 , 176.1 ,
176.05, 185.25, 197.  , 181.7 , 173.3 , 171.5 ])
```

```
In [25]: type(x), type(y)
```

```
Out[25]: (numpy.ndarray, numpy.ndarray)
```

```
In [26]: from sklearn.model_selection import train_test_split
```

```
In [27]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=0)
```

```
In [28]: x_train
```

```
Out[28]: array([[1.7295000e+02, 1.6510000e+02, 1.6570000e+02, 1.6555000e+02,
1.1916843e+07, 2.0066790e+04],
[1.8235000e+02, 1.7260000e+02, 1.7300000e+02, 1.7395000e+02,
1.8481283e+07, 3.2703520e+04],
[1.7360000e+02, 1.6955000e+02, 1.7330000e+02, 1.7275000e+02,
2.6320880e+06, 1.4922250e+04],
```

```
[1.7790000e+02, 1.7405000e+02, 1.7455000e+02, 1.7465000e+02,
 1.1152776e+07, 1.9639040e+04],
[1.8780000e+02, 1.7525000e+02, 1.8600000e+02, 1.8580000e+02,
 2.8176767e+07, 5.1344080e+04],
[1.8050000e+02, 1.7540000e+02, 1.7605000e+02, 1.7600000e+02,
 1.5500426e+07, 2.7508990e+04],
[1.6930000e+02, 1.6620000e+02, 1.6650000e+02, 1.6660000e+02,
 8.5342330e+06, 1.4308850e+04],
[1.7815000e+02, 1.7510000e+02, 1.7565000e+02, 1.7570000e+02,
 7.3022940e+06, 1.2881700e+04],
[1.9635000e+02, 1.8210000e+02, 1.9545000e+02, 1.9465000e+02,
 5.0753260e+07, 9.6318970e+04],
[1.7490000e+02, 1.7145000e+02, 1.7450000e+02, 1.7420000e+02,
 1.1223630e+07, 1.9450160e+04],
[1.7490000e+02, 1.7000000e+02, 1.7010000e+02, 1.7065000e+02,
 9.0211310e+06, 1.5507280e+04],
[1.9700000e+02, 1.8000000e+02, 1.8140000e+02, 1.8090000e+02,
 4.5808402e+07, 8.4712340e+04],
[1.7970000e+02, 1.6540000e+02, 1.7800000e+02, 1.7645000e+02,
 3.1613344e+07, 5.4653030e+04],
[1.7340000e+02, 1.6890000e+02, 1.6970000e+02, 1.6975000e+02,
 7.8691490e+06, 1.3480320e+04],
[1.7730000e+02, 1.7220000e+02, 1.7620000e+02, 1.7600000e+02,
 1.1370450e+07, 1.9896680e+04],
[1.7700000e+02, 1.6265000e+02, 1.7650000e+02, 1.7555000e+02,
 2.5299752e+07, 4.3070740e+04]])
```

```
In [29]: x_test
```

```
Out[29]: array([[1.7590000e+02, 1.7140000e+02, 1.7225000e+02, 1.7265000e+02,
 1.6914910e+07, 2.9341510e+04],
 [1.7585000e+02, 1.7185000e+02, 1.7205000e+02, 1.7215000e+02,
 7.1046480e+06, 1.2340180e+04],
 [1.7800000e+02, 1.6935000e+02, 1.7690000e+02, 1.7640000e+02,
 2.1519974e+07, 3.7574060e+04],
 [1.8275000e+02, 1.6865000e+02, 1.7035000e+02, 1.7050000e+02,
 3.2923946e+07, 5.8071000e+04],
 [1.7090000e+02, 1.6540000e+02, 1.6615000e+02, 1.6640000e+02,
 1.3729132e+07, 2.2963830e+04],
 [1.7520000e+02, 1.6880000e+02, 1.7090000e+02, 1.7135000e+02,
 1.5313331e+07, 2.6320880e+04]])
```

```
In [30]: y_test
```

```
Out[30]: array([173.3 , 174.95, 170.8 , 180.7 , 170.5 , 171.5 ])
```

```
In [31]: y_train
```

```
Out[31]: array([172.25, 181.7 , 170.3 , 175.3 , 176.05, 180.4 , 168. , 177.05,
 185.25, 173.5 , 174.45, 197. , 167.35, 172.95, 176.1 , 165.5 ])
```

FITTING THE MODEL LINEAR REGRESSION

```
In [57]: from sklearn.linear_model import LinearRegression
```

```
In [58]: LR=LinearRegression()
```

```
In [59]: LR.fit(x_train,y_train)
```

Out[59]: LinearRegression()

In [60]: `LR.score(x_test,y_test)`

Out[60]: 0.9539309730486247

In [61]: `Test_Data=[[49.41,48.56,49.66,50.60,50.12,50.0]]`

In [62]: `Test_Data`

Out[62]: [[49.41, 48.56, 49.66, 50.6, 50.12, 50.0]]

In [63]: `type(Test_Data)`

Out[63]: list

In [64]: `Predicted_Price=LR.predict(Test_Data)`

In [65]: `print(Predicted_Price)`

[51.2995235]

The predicted Price is 51.299 while the real price on that day is 50.55.

CONCLUSION: Linear Regression Model is working fine On this Dataset