

```
[In [3]]: import libraries
import pandas as pd
import numpy as np
from sklearn import metrics
import matplotlib.pyplot as plt
import seaborn as sns

[In [4]]: df=pd.read_csv("GOOG.csv")
df.head()
```

```
Out[4]:
```

	symbol	date	close	high	low	open	adjClose	adjHigh	adjLow	adjOpen	adjVolume	divCash	splitFactor	
0	GOOG	2016-06-14 00:00:00+00:00	718.27	722.47	713.1200	716.48	1306065	718.27	722.47	713.1200	716.48	1306065	0.0	1.0
1	GOOG	2016-06-15 00:00:00+00:00	718.82	722.98	717.3100	719.00	1214517	718.92	722.98	717.3100	719.00	1214517	0.0	1.0
2	GOOG	2016-06-16 00:00:00+00:00	710.36	716.65	703.2600	714.91	1982471	710.36	716.65	703.2600	714.91	1982471	0.0	1.0
3	GOOG	2016-06-17 00:00:00+00:00	691.72	708.82	688.4515	706.65	3402357	691.72	708.82	688.4515	706.65	3402357	0.0	1.0
4	GOOG	2016-06-20 00:00:00+00:00	693.71	702.48	693.4100	696.77	2082538	693.71	702.48	693.4100	696.77	2082538	0.0	1.0

```
[In [5]]: df.shape
Out[5]: (1258, 14)
```

```
[In [6]]: df=df.drop(columns=[
    'symbol','adjClose','adjHigh','adjLow','adjOpen','adjVolume','divCash','splitFactor'
])
sns.boxplot(y='close',data=df)
df.head()
```

```
Out[6]:
```

	date	close	high	low	open	volume
0	2016-06-14 00:00:00+00:00	718.27	722.47	713.1200	716.48	1306065
1	2016-06-15 00:00:00+00:00	718.82	722.98	717.3100	719.00	1214517
2	2016-06-16 00:00:00+00:00	710.36	716.65	703.2600	714.91	1982471
3	2016-06-17 00:00:00+00:00	691.72	708.82	688.4515	706.65	3402357
4	2016-06-20 00:00:00+00:00	693.71	702.48	693.4100	696.77	2082538

```
[In [7]]: #Are there any duplicate values
df.duplicated().sum().any()
Out[7]: False
```

```
[In [8]]: #Checking & reviewing DataFrame information
df.isnull().values.any()
Out[8]: False
```

```
[In [9]]: df.describe()
Out[9]:
```

	close	high	low	open	volume
count	1258.000000	1258.000000	1258.000000	1258.000000	1.280000e+03
mean	1216.3117067	1227.430934	1204.176430	1215.260779	1.6015900e+06
std	383.333358	387.570072	378.777094	382.446995	6.860172e+05
min	668.260000	672.300000	663.284000	671.000000	3.467530e+05
50%	960.802500	968.750000	952.182500	959.005000	1.173520e+06
25%	1132.460000	1143.935000	1117.915000	1131.150000	1.412880e+06
75%	1360.595000	1374.345000	1348.557500	1361.075000	1.812150e+06
max	2521.600000	2526.990000	2498.290000	2524.920000	6.207027e+06

```
[In [10]]: df['date'] = pd.to_datetime(df['date'])
df.head()
```

```
Out[10]:
```

	date	close	high	low	open	volume
0	2016-06-14 00:00:00+00:00	718.27	722.47	713.1200	716.48	1306065
1	2016-06-15 00:00:00+00:00	718.82	722.98	717.3100	719.00	1214517
2	2016-06-16 00:00:00+00:00	710.36	716.65	703.2600	714.91	1982471
3	2016-06-17 00:00:00+00:00	691.72	708.82	688.4515	706.65	3402357
4	2016-06-20 00:00:00+00:00	693.71	702.48	693.4100	696.77	2082538

```
[In [11]]: df['date'] = pd.to_datetime(df['date'])
df.head()
```

```
Out[11]:
```

	date	close	high	low	open	volume
0	2016-06-14 00:00:00+00:00	718.27	722.47	713.1200	716.48	1306065
1	2016-06-15 00:00:00+00:00	718.82	722.98	717.3100	719.00	1214517
2	2016-06-16 00:00:00+00:00	710.36	716.65	703.2600	714.91	1982471
3	2016-06-17 00:00:00+00:00	691.72	708.82	688.4515	706.65	3402357
4	2016-06-20 00:00:00+00:00	693.71	702.48	693.4100	696.77	2082538

```
[In [12]]: df['date'] = df['date'].dt.strftime('%Y-%m-%d')
df.head()
```

```
Out[12]:
```

	date	close	high	low	open	volume
0	2016-06-14	718.27	722.47	713.1200	716.48	1306065
1	2016-06-15	718.82	722.98	717.3100	719.00	1214517
2	2016-06-16	710.36	716.65	703.2600	714.91	1982471
3						