```
import numpy as np
import pandas as pd

data=pd.read_csv("ICICIBANK.csv")
print(data)
```

```
            Date     Symbol Series  ...    Trades  Deliverable Volume  %Deliverble
0     03-01-2000  ICICIBANK     EQ  ...       NaN                 NaN          NaN
1     04-01-2000  ICICIBANK     EQ  ...       NaN                 NaN          NaN
2     05-01-2000  ICICIBANK     EQ  ...       NaN                 NaN          NaN
3     06-01-2000  ICICIBANK     EQ  ...       NaN                 NaN          NaN
4     07-01-2000  ICICIBANK     EQ  ...       NaN                 NaN          NaN
...          ...        ...    ...  ...       ...                 ...          ...
5301  26-04-2021  ICICIBANK     EQ  ...  546696.0          16913351.0       0.2702
5302  27-04-2021  ICICIBANK     EQ  ...  233412.0           9056247.0       0.3736
5303  28-04-2021  ICICIBANK     EQ  ...  295406.0          14011476.0       0.4209
5304  29-04-2021  ICICIBANK     EQ  ...  304029.0          10324897.0       0.2989
5305  30-04-2021  ICICIBANK     EQ  ...  247055.0           7292392.0       0.3083

[5306 rows x 15 columns]
```

```
data.head(10)
```

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	T
0	03-01-2000	ICICIBANK	EQ	69.20	74.35	74.75	71.40	74.75	74.75	73.20	286260	2.095
1	04-01-2000	ICICIBANK	EQ	74.75	73.05	78.50	71.00	73.25	73.05	73.38	296264	2.173
2	05-01-2000	ICICIBANK	EQ	73.05	70.00	73.50	67.50	70.00	69.50	70.85	227624	1.612
3	06-01-2000	ICICIBANK	EQ	69.50	71.00	74.00	69.55	69.75	70.05	72.04	275149	1.982
4	07-01-	ICICIBANK	EQ	70.05	69.00	72.50	66.00	67.00	67.40	68.72	138809	9.538

```
data.tail(6)
```

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Vo
5300	23-04-2021	ICICIBANK	EQ	579.20	569.90	580.50	567.05	567.5	569.95	572.81	33749
5301	26-04-2021	ICICIBANK	EQ	569.95	602.00	605.50	588.00	588.7	591.10	596.22	62584
5302	27-04-2021	ICICIBANK	EQ	591.10	593.25	601.95	591.10	599.0	598.75	597.63	24239

```
df.dtypes
```

```
Date                    object
Symbol                  object
Series                  object
Prev Close             float64
Open                   float64
High                   float64
Low                    float64
Last                   float64
Close                  float64
VWAP                   float64
Volume                   int64
Turnover               float64
Trades                 float64
Deliverable Volume     float64
%Deliverble            float64
dtype: object
```

```
print(df.shape)
print(df.size)
```

```
(5306, 15)
79590
```

```
df.keys()
```

```
Index(['Date', 'Symbol', 'Series', 'Prev Close', 'Open', 'High', 'Low', 'Last',
       'Close', 'VWAP', 'Volume', 'Turnover', 'Trades', 'Deliverable Volume',
       '%Deliverble'],
      dtype='object')
```

```
df.isnull()
```

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turno
0	False	False	False	False	False	False	False	False	False	False	False	F
1	False	False	False	False	False	False	False	False	False	False	False	F
2	False	False	False	False	False	False	False	False	False	False	False	F
3	False	False	False	False	False	False	False	False	False	False	False	F
4	False	False	False	False	False	False	False	False	False	False	False	F
...	...	...	...	...	...	...	...	...	...	...	...	
5301	False	False	False	False	False	False	False	False	False	False	False	F
5302	False	False	False	False	False	False	False	False	False	False	False	F
5303	False	False	False	False	False	False	False	False	False	False	False	F

```
df.describe()
```

	Prev Close	Open	High	Low	Last	Close	
count	5306.000000	5306.000000	5306.000000	5306.000000	5306.000000	5306.000000	5306.(
mean	550.895392	551.558538	560.558556	541.534197	551.050980	550.995524	551.
std	368.784064	368.890953	374.079697	363.389664	368.705647	368.725374	368.
min	67.400000	67.000000	70.450000	66.000000	67.000000	67.400000	68.
25%	267.562500	267.400000	271.912500	263.625000	267.400000	267.612500	267.
50%	398.075000	399.000000	406.525000	392.450000	398.700000	398.175000	398.
75%	873.562500	877.000000	888.775000	859.800000	874.600000	873.562500	873.
max	1794.100000	1767.050000	1798.150000	1760.150000	1793.000000	1794.100000	1783.

```
df.duplicated()
```

```
0       False
1       False
2       False
3       False
4       False
        ...
5301    False
5302    False
5303    False
5304    False
```

```
5305    False
```

```
df.drop_duplicates()
```

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Vo
0	03-01-2000	ICICIBANK	EQ	69.20	74.35	74.75	71.40	74.75	74.75	73.20	28
1	04-01-2000	ICICIBANK	EQ	74.75	73.05	78.50	71.00	73.25	73.05	73.38	29
2	05-01-2000	ICICIBANK	EQ	73.05	70.00	73.50	67.50	70.00	69.50	70.85	22
3	06-01-2000	ICICIBANK	EQ	69.50	71.00	74.00	69.55	69.75	70.05	72.04	27
4	07-01-2000	ICICIBANK	EQ	70.05	69.00	72.50	66.00	67.00	67.40	68.72	13
...	...	...	...	...	...	...	...	...	...	...	
5301	26-04-2021	ICICIBANK	EQ	569.95	602.00	605.50	588.00	588.70	591.10	596.22	6258
5302	27-04-2021	ICICIBANK	EQ	591.10	593.25	601.95	591.10	599.00	598.75	597.63	2423
5303	28-04-2021	ICICIBANK	EQ	598.75	598.00	622.80	598.00	619.70	621.35	612.57	3329

```
df.mean()
```

```
Prev Close            5.508954e+02
Open                  5.515585e+02
High                  5.605586e+02
Low                   5.415342e+02
Last                  5.510510e+02
Close                 5.509955e+02
VWAP                  5.511290e+02
Volume                8.224631e+06
Turnover              3.759299e+14
Trades                1.383676e+05
Deliverable Volume    4.183406e+06
%Deliverble           4.734630e-01
dtype: float64
```

```
df.median()
```

```
Prev Close            3.980750e+02
Open                  3.990000e+02
High                  4.065250e+02
Low                   3.924500e+02
Last                  3.987000e+02
Close                 3.981750e+02
VWAP                  3.982350e+02
Volume                3.486648e+06
Turnover              2.923010e+14
Trades                1.101010e+05
Deliverable Volume    1.963117e+06
%Deliverble           4.764000e-01
dtype: float64
```

```
df.mode()
```

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume
0	01-01-2001	ICICIBANK	EQ	270.05	142.0	145.0	120.0	140.0	270.05	102.62	66961.(
1	01-01-2002	NaN	NaN	289.30	NaN	NaN	140.0	NaN	289.30	141.77	404216.(
2	01-01-2003	NaN	NaN	NaN	NaN	NaN	148.0	NaN	NaN	261.39	689289.(
3	01-01-2004	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	286.05	NaN
4	01-01-2008	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	295.31	NaN
...	...	...	...	...	...	...	...	...	...	...	..
5301	31-12-2014	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5302	31-12-2015	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5303	31-12-2018	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
## variance ##
```

```
np.var(df)
```

```
Prev Close            1.359761e+05
Open                  1.360549e+05
High                  1.399092e+05
Low                   1.320272e+05
Last                  1.359182e+05
Close                 1.359328e+05
VWAP                  1.359487e+05
Volume                1.484547e+14
Turnover              2.263557e+29
Trades                9.798737e+09
Deliverable Volume    4.050962e+13
%Deliverble           1.728741e-02
dtype: float64
```

```
## standard deviation ##
np.std(df)
```

```
Prev Close            3.687493e+02
Open                  3.688562e+02
High                  3.740444e+02
Low                   3.633554e+02
Last                  3.686709e+02
Close                 3.686906e+02
VWAP                  3.687122e+02
Volume                1.218420e+07
Turnover              4.757685e+14
Trades                9.898857e+04
Deliverable Volume    6.364717e+06
%Deliverble           1.314816e-01
dtype: float64
```

```
## range
df['Low'].max() - df['Low'].min()
```

```
1694.15
```

```
## q3, q1 = np.percentile(data, []) ##
## iqr = q3 - q1 ##
data['Low'].quantile([0.25,0.5,0.75])
```

```
0.25    263.625
0.50    392.450
0.75    859.800
Name: Low, dtype: float64
```

```
## Skewness =3(Mean- Median)/Standard Deviation ##
df.skew()
```

```
Prev Close             0.751634
Open                   0.748978
```

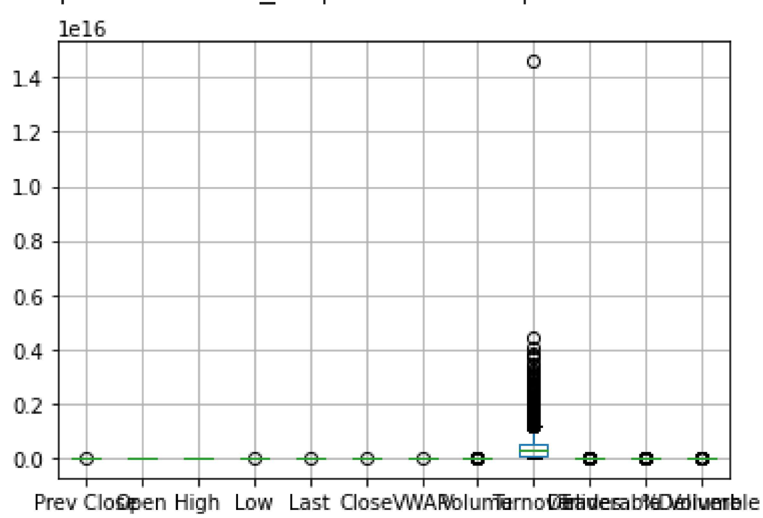
```
High                  0.743097
Low                   0.759964
Last                  0.751635
Close                 0.751599
VWAP                  0.751310
Volume                4.856834
Turnover              7.000400
Trades                2.910304
Deliverable Volume   11.910473
%Deliverble           0.007097
dtype: float64
```

```
## krutosis ##
df.kurtosis()
```

```
Prev Close            -0.432107
Open                  -0.445145
High                  -0.461216
Low                   -0.408504
Last                  -0.431299
Close                 -0.431838
VWAP                  -0.434371
Volume                62.835324
Turnover             157.331069
Trades                12.748884
Deliverable Volume   361.297706
%Deliverble           -0.103833
dtype: float64
```

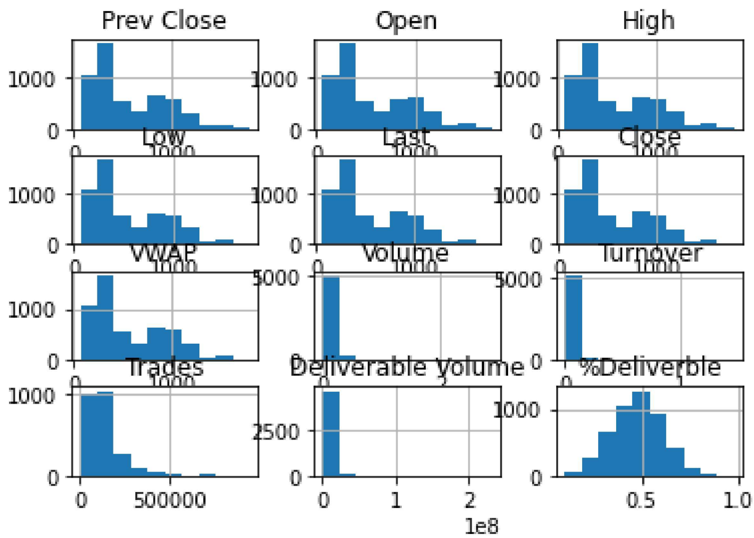
```
import matplotlib.pyplot
df.boxplot()
```

```
/usr/local/lib/python3.7/dist-packages/numpy/core/_asarray.py:83: VisibleDeprecationWarr
  return array(a, dtype, copy=False, order=order)
<matplotlib.axes._subplots.AxesSubplot at 0x7ff1de246a10>
```



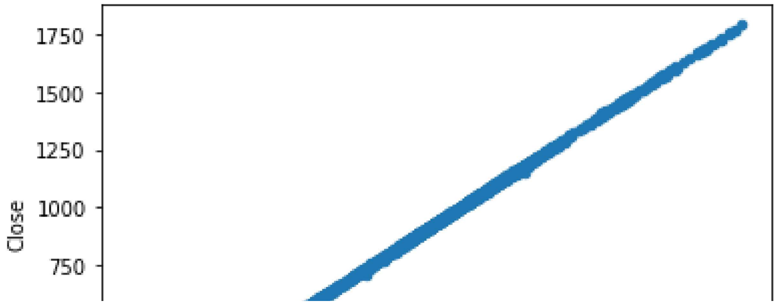

```
df.hist()
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7ff1de0845d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7ff1de0a4e10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7ff1de067490>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7ff1de01cb10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7ff1ddfe01d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7ff1ddf95850>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7ff1ddf4bf50>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7ff1ddf0e550>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7ff1ddf0e590>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7ff1dded00d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7ff1dde46bd0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7ff1dde03290>]],
      dtype=object)
```




```
print(data.plot.scatter('Last','Close'))
print(data.plot.scatter('Low','High'))
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
AxesSubplot(0.125,0.125;0.775x0.755)
```




```
import seaborn as sns
sns.pairplot(data[['Open','High','Low','Last']])
```

```
<seaborn.axisgrid.PairGrid at 0x7ff1dd935e10>
```

```
max_price = data["Close"].max()
print(max_price)
```

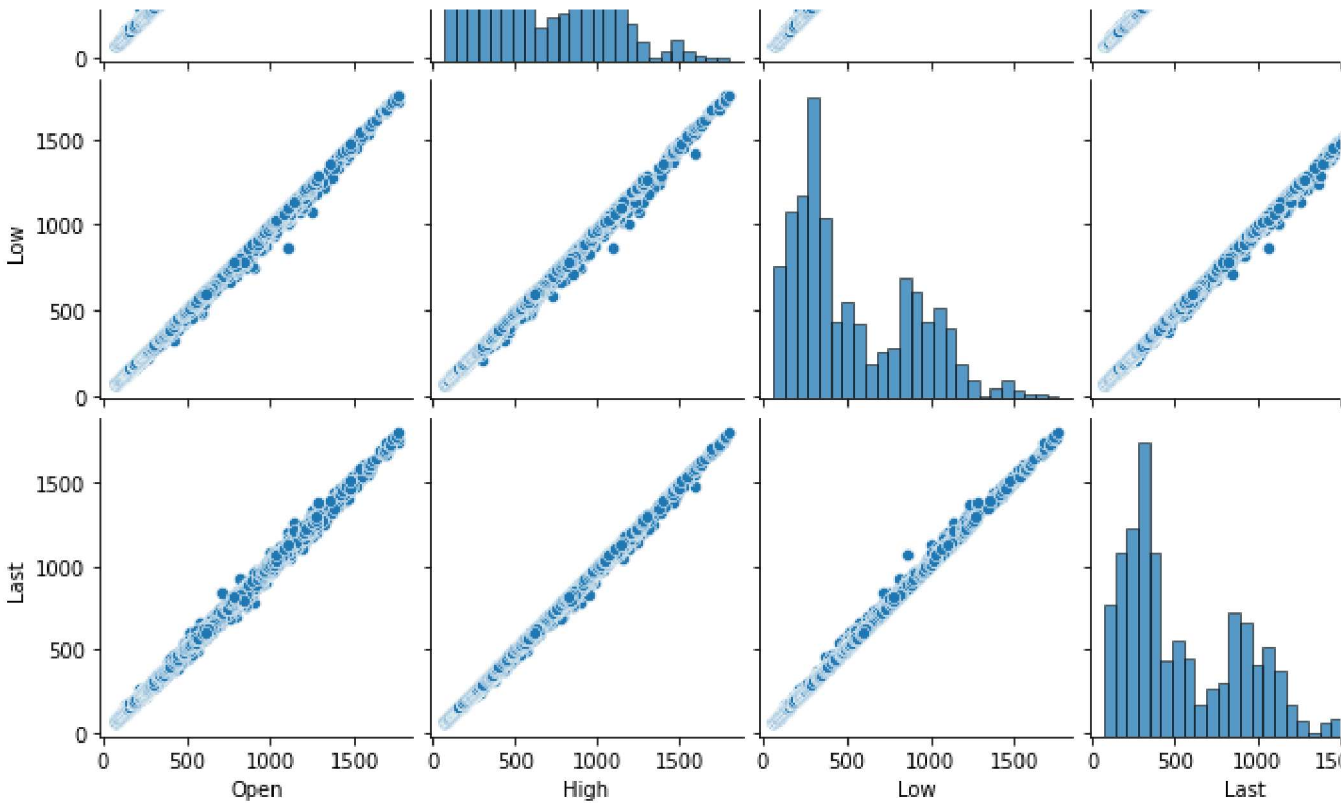
```
1794.1
```

```
min_price = data['Close'].min()
print(min_price)
```

```
67.4
```


```
max_price = data['Open'].max()
print(max_price)
```

```
1767.05
```



✓ 0s completed at 18:51