```
import numpy as np
import math
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns


url = 'https://raw.githubusercontent.com/mwitiderrick/stockprice/master/NSE-TATAGLOBAL.csv
data = pd.read_csv(url)
data
```

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-09-28	234.05	235.95	230.20	233.50	233.75	3069914	7162.35
1	2018-09-27	234.55	236.80	231.10	233.80	233.25	5082859	11859.95
2	2018-09-26	240.00	240.00	232.50	235.00	234.25	2240909	5248.60
3	2018-09-25	233.30	236.75	232.00	236.25	236.10	2349368	5503.90
4	2018-09-24	233.55	239.20	230.75	234.00	233.30	3423509	7999.55
...	...	...	...	...	...	...	...	...
2030	2010-07-27	117.60	119.50	112.00	118.80	118.65	586100	694.98
2031	2010-07-26	120.10	121.00	117.10	117.10	117.60	658440	780.01

Describing the Dataset

```
data.describe()
```

```
data.tail()
```

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
2030	2010-07-27	117.6	119.50	112.00	118.80	118.65	586100	694.98
2031	2010-07-26	120.1	121.00	117.10	117.10	117.60	658440	780.01
2032	2010-07-23	121.8	121.95	120.25	120.35	120.65	281312	340.31

```
data.dtypes
```

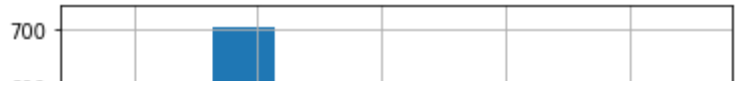
```
Date                   object
Open                  float64
High                  float64
Low                   float64
Last                  float64
Close                 float64
Total Trade Quantity    int64
Turnover (Lacs)       float64
dtype: object
```

```
data['Date'].value_counts()
```

```
2018-09-28    1
2013-04-10    1
2013-03-20    1
2013-03-21    1
2013-03-22    1
             ..
2016-01-11    1
2016-01-12    1
2016-01-13    1
2016-01-14    1
2010-07-21    1
Name: Date, Length: 2035, dtype: int64
```

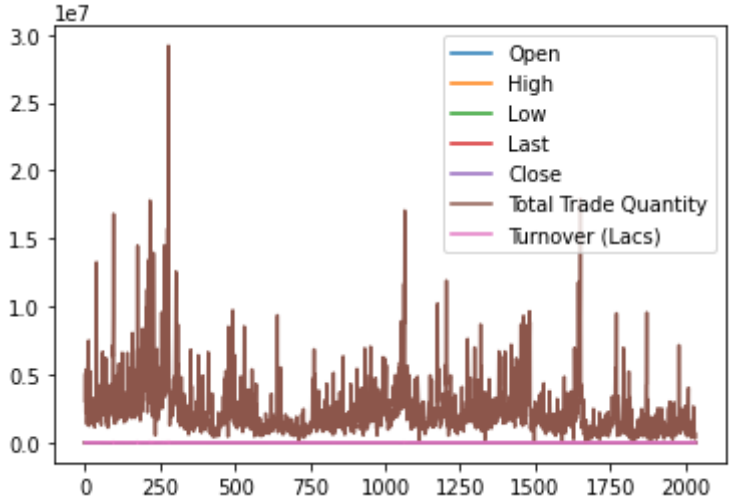
```
data['High'].hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe90c5373d0>
```




```
plt.figure(figsize=(20,8))
data.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe90b22cf50>
<Figure size 1440x576 with 0 Axes>
```




```
data_set = data.filter(['Close'])
dataset = data.values
training_data_len=math.ceil(len(data) * 8)
training_data_len
```

```
16280
```

```
dataset
```

```
array([['2018-09-28', 234.05, 235.95, ..., 233.75, 3069914, 7162.35],
       ['2018-09-27', 234.55, 236.8, ..., 233.25, 5082859, 11859.95],
       ['2018-09-26', 240.0, 240.0, ..., 234.25, 2240909, 5248.6],
       ...,
       ['2010-07-23', 121.8, 121.95, ..., 120.65, 281312, 340.31],
       ['2010-07-22', 120.3, 122.0, ..., 120.9, 293312, 355.17],
       ['2010-07-21', 122.1, 123.0, ..., 121.55, 658666, 803.56]],
      dtype=object)
```

```
data = data.iloc[:, 0:5]
data
```

	Date	Open	High	Low	Last
0	2018-09-28	234.05	235.95	230.20	233.50
1	2018-09-27	234.55	236.80	231.10	233.80
2	2018-09-26	240.00	240.00	232.50	235.00
3	2018-09-25	233.30	236.75	232.00	236.25
4	2018-09-24	233.55	239.20	230.75	234.00
...	...	...	...	...	...
2030	2010-07-27	117.60	119.50	112.00	118.80

```
training_set = data.iloc[:, 1:2].values
training_set
```

```
array([[234.05],
       [234.55],
       [240.  ],
       ...,
       [121.8 ],
       [120.3 ],
       [122.1 ]])
```

Scalling of Data Set

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (0, 1))
data_training_scaled = scaler.fit_transform(training_set)
```

```
features_set = []
labels = []
for i in range(60, 586):
  features_set.append(data_training_scaled[i - 60:i, 0])
  labels.append(data_training_scaled[i, 0])
```

```
features_set, labels = np.array(features_set), np.array(labels)
```

```
features_set = np.reshape(features_set, (features_set.shape[0], features_set.shape[1], 1))
features_set.shape
```

```
(526, 60, 1)
```

Building The LSTM

```
import tensorflow as tf
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dense
from tensorflow.python.keras.layers import LSTM
```

```
model = Sequential()


model.compile(optimizer='adam', loss='mean_squared_error')


model.fit(features_set, labels, epochs=50, batch_size=20)
```

```
Epoch 23/50
27/27 [==============================] - 0s 3ms/step - loss: 0.0118
Epoch 24/50
27/27 [==============================] - 0s 5ms/step - loss: 0.0118
Epoch 25/50
27/27 [==============================] - 0s 3ms/step - loss: 0.0118
Epoch 26/50
27/27 [==============================] - 0s 3ms/step - loss: 0.0118
Epoch 27/50
27/27 [==============================] - 0s 4ms/step - loss: 0.0118
Epoch 28/50
27/27 [==============================] - 0s 1ms/step - loss: 0.0118
Epoch 29/50
27/27 [==============================] - 0s 2ms/step - loss: 0.0118
Epoch 30/50
27/27 [==============================] - 0s 4ms/step - loss: 0.0118
Epoch 31/50
27/27 [==============================] - 0s 3ms/step - loss: 0.0118
Epoch 32/50
27/27 [==============================] - 0s 3ms/step - loss: 0.0118
Epoch 33/50
27/27 [==============================] - 0s 2ms/step - loss: 0.0118
Epoch 34/50
27/27 [==============================] - 0s 3ms/step - loss: 0.0118
Epoch 35/50
27/27 [==============================] - 0s 2ms/step - loss: 0.0118
Epoch 36/50
27/27 [==============================] - 0s 2ms/step - loss: 0.0118
Epoch 37/50
27/27 [==============================] - 0s 2ms/step - loss: 0.0118
Epoch 38/50
27/27 [==============================] - 0s 2ms/step - loss: 0.0118
Epoch 39/50
27/27 [==============================] - 0s 1ms/step - loss: 0.0118
Epoch 40/50
27/27 [==============================] - 0s 2ms/step - loss: 0.0118
Epoch 41/50
27/27 [==============================] - 0s 2ms/step - loss: 0.0118
Epoch 42/50
27/27 [==============================] - 0s 2ms/step - loss: 0.0118
Epoch 43/50
27/27 [==============================] - 0s 2ms/step - loss: 0.0118
Epoch 44/50
27/27 [==============================] - 0s 2ms/step - loss: 0.0118
Epoch 45/50
27/27 [==============================] - 0s 2ms/step - loss: 0.0118
Epoch 46/50
27/27 [==============================] - 0s 2ms/step - loss: 0.0118
Epoch 47/50
27/27 [==============================] - 0s 3ms/step - loss: 0.0118
```

```
Epoch 48/50
27/27 [==============================] - 0s 3ms/step - loss: 0.0118
Epoch 49/50
27/27 [==============================] - 0s 2ms/step - loss: 0.0118
Epoch 50/50
27/27 [==============================] - 0s 1ms/step - loss: 0.0118
<tensorflow.python.keras.callbacks.History at 0x7fe89e798a90>
```

```
data_testing_complete = pd.read_csv(url)
data_testing_processed = data_testing_complete.iloc[:, 1:2]
data_testing_processed
```

	Open
0	234.05
1	234.55
2	240.00
3	233.30
4	233.55
...	...
2030	117.60
2031	120.10
2032	121.80
2033	120.30
2034	122.10

2035 rows × 1 columns

Prediction of the Data

```
data_total = pd.concat((data['Open'], data['Open']), axis=0)
```

```
test_inputs = data_total[len(data_total) - len(data) - 60:].values
test_inputs.shape
```

```
(2095,)
```

```
test_inputs = test_inputs.reshape(-1, 1)
test_inputs = scaler.transform(test_inputs)
```

```
test_feature = []
for i in range(60, 89):
  test_feature.append(test_inputs[i-60:i, 0])
```

```
test_feature = np.array(test_feature)
test_feature = np.reshape(test_feature, (test_feature.shape[0] - test_feature.shape[1], 1)
test_feature.shape
```

```
(1740, 1)
```

```
predictions = model.predict(test_feature)
```

```
predictions
```

```
array([[0.20600162],
       [0.21654502],
       [0.21654502],
       ...,
       [0.67234385],
       [0.6605839 ],
       [0.64760745]], dtype=float32)
```

```
x_train = data[0:1256]
y_train = data[1:1257]
print(x_train.shape)
print(y_train.shape)
```

```
(1256, 5)
(1256, 5)
```

```
x_train
```

	Date	Open	High	Low	Last
0	2018-09-28	234.05	235.95	230.20	233.50
1	2018-09-27	234.55	236.80	231.10	233.80
2	2018-09-26	240.00	240.00	232.50	235.00
3	2018-09-25	233.30	236.75	232.00	236.25
4	2018-09-24	233.55	239.20	230.75	234.00
...	...	...	...	...	...
1251	2013-09-04	142.00	145.35	140.65	143.60
1252	2013-09-03	144.10	145.20	140.70	141.80
1253	2013-09-02	139.40	144.40	139.35	144.00
1254	2013-08-30	138.10	140.65	136.70	139.20
1255	2013-08-29	137.00	140.40	137.00	137.10

1256 rows × 5 columns

```
np.random.seed(1)
```

```
np.random.randn(3, 3)
```

```
array([[ 1.62434536, -0.61175641, -0.52817175],
       [-1.07296862,  0.86540763, -2.3015387 ],
       [ 1.74481176, -0.7612069 ,  0.3190391 ]])
```

Drawing a Single number from the Normal Distribution

```
np.random.normal(1)
```

```
0.7506296245225899
```

Drawing 5 numbers from Normal Distribution

```
np.random.normal(5)
```

```
6.4621079370449745
```

```
np.random.seed(42)
```

```
np.random.normal(size=1000, scale=100).std()
```

```
97.87262077473541
```

Ploting Results

```
plt.figure(figsize=(18,6))
plt.title("Stock Market Price Prediction")
plt.plot(data_testing_complete['Close'])
plt.xlabel('Date', fontsize=18)
plt.ylabel('Total Trade Quantity', fontsize=18)
plt.show()
```

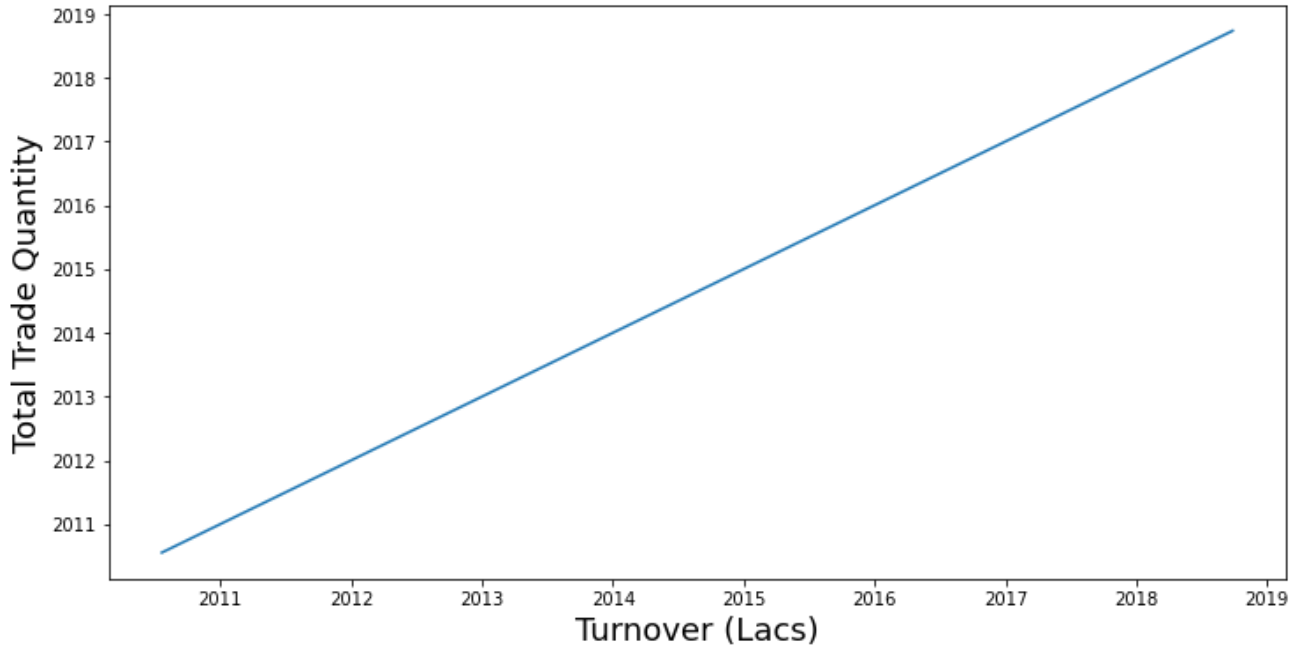
Analyze the Closing price from the dataframe

```
data["Date"] = pd.to_datetime(data.Date)
data.index = data['Date']

plt.figure(figsize=(20, 10))
plt.plot(data["Open"], label='ClosePriceHist')
```

```
[<matplotlib.lines.Line2D at 0x7fe89e45e990>]
```


```
plt.figure(figsize=(12,6))
plt.plot(data['Date'])
plt.xlabel('Turnover (Lacs)', fontsize=18)
plt.ylabel('Total Trade Quantity', fontsize=18)
plt.show()
```



Analyze the Closing price from the

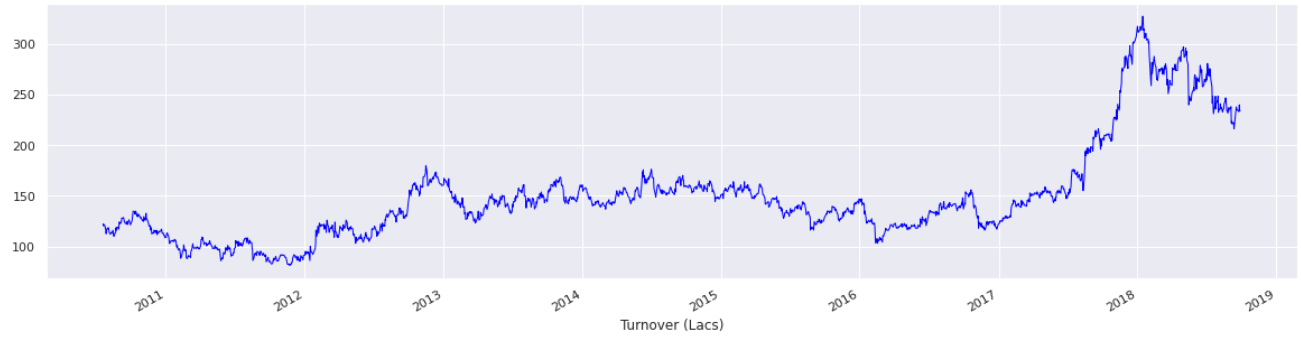
```
data["Turnover (Lacs)"] = pd.to_datetime(data.Date)
data.index = data['Turnover (Lacs)']

plt.figure(figsize=(20, 10))
plt.plot(data["Turnover (Lacs)"], label='ClosePriceHist')
```

```
[<matplotlib.lines.Line2D at 0x7fe89e281e10>]
```


```
sns.set(rc = {'figure.figsize': (20, 5)})
data['Open'].plot(linewidth = 1,color='blue')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe89e2ade50>
```




```
data.columns
```

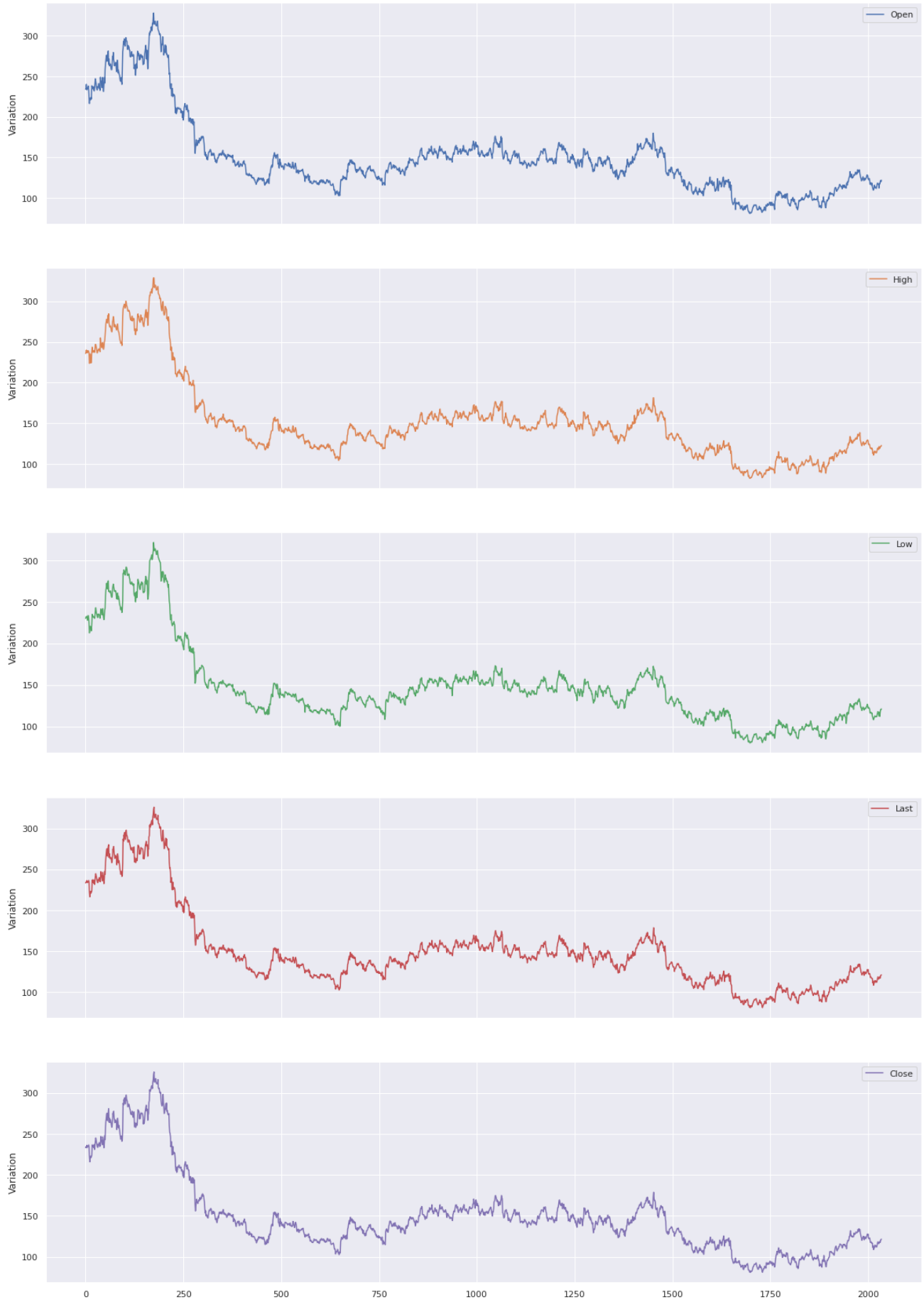
```
Index(['Date', 'Open', 'High', 'Low', 'Last', 'Turnover (Lacs)'], dtype='object')
```

```
df = pd.read_csv(url)
df
```

Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)

```
cols_plot = ['Open','High','Low','Last','Close']
axes = df[cols_plot].plot(alpha = 1, figsize=(20, 30), subplots = True)

for ax in axes:
    ax.set_ylabel('Variation')
```


Open
High
Low
Last
Close
Variation
300
250
200
150
100
0
250
500
750
1000
1250
1500
1750
2000

✓ 6s completed at 3:40 PM