

```
In [3]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from scipy import stats
from scipy.stats import probplot
```

```
In [4]: data = pd.read_csv("/Users/rupikapendyala/Desktop/coin_Bitcoin.csv")
data.head()
```

```
Out[4]:
```

	SNo	Name	Symbol	Date	High	Low	Open	Close	Volume	Marketcap
0	1	Bitcoin	BTC	2013-04-29 23:59:59	147.488007	134.000000	134.444000	144.539993	0.0	1.603769e+09
1	2	Bitcoin	BTC	2013-04-30 23:59:59	146.929993	134.050003	144.000000	139.000000	0.0	1.542813e+09
2	3	Bitcoin	BTC	2013-05-01 23:59:59	139.889999	107.720001	139.000000	116.989998	0.0	1.298955e+09
3	4	Bitcoin	BTC	2013-05-02 23:59:59	125.599998	92.281898	116.379997	105.209999	0.0	1.168517e+09
4	5	Bitcoin	BTC	2013-05-03 23:59:59	108.127998	79.099998	106.250000	97.750000	0.0	1.085995e+09

```
In [5]: data = data.drop("Name", axis=1)
data = data.drop("Symbol", axis=1)
data = data.drop("SNo", axis=1)
```

```
In [6]: data.head()
```

```
Out[6]:
```

	Date	High	Low	Open	Close	Volume	Marketcap
0	2013-04-29 23:59:59	147.488007	134.000000	134.444000	144.539993	0.0	1.603769e+09
1	2013-04-30 23:59:59	146.929993	134.050003	144.000000	139.000000	0.0	1.542813e+09
2	2013-05-01 23:59:59	139.889999	107.720001	139.000000	116.989998	0.0	1.298955e+09
3	2013-05-02 23:59:59	125.599998	92.281898	116.379997	105.209999	0.0	1.168517e+09
4	2013-05-03 23:59:59	108.127998	79.099998	106.250000	97.750000	0.0	1.085995e+09

```
In [7]: data.isnull().sum()
```

```
Out[7]: Date      0
        High      0
        Low       0
        Open      0
        Close     0
        Volume    0
        Marketcap 0
        dtype: int64
```

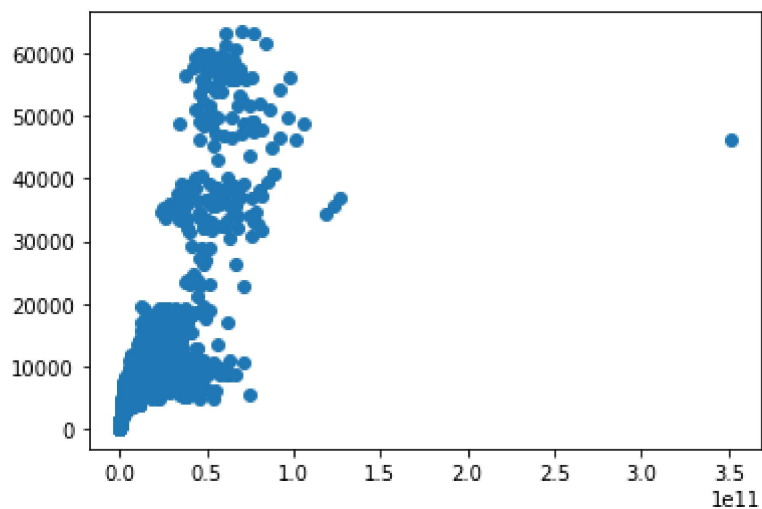
```
In [8]: data.shape
```

```
Out[8]: (2991, 7)
```

```
In [9]: cols = ['High', 'Low', 'Open', 'Close', 'Volume', 'Marketcap']
```

```
In [10]: import matplotlib.pyplot as plt
x = data['Volume']
y = data['Close']
plt.scatter(x.values, y.values)
```

```
Out[10]: <matplotlib.collections.PathCollection at 0x7fe12b8ec550>
```

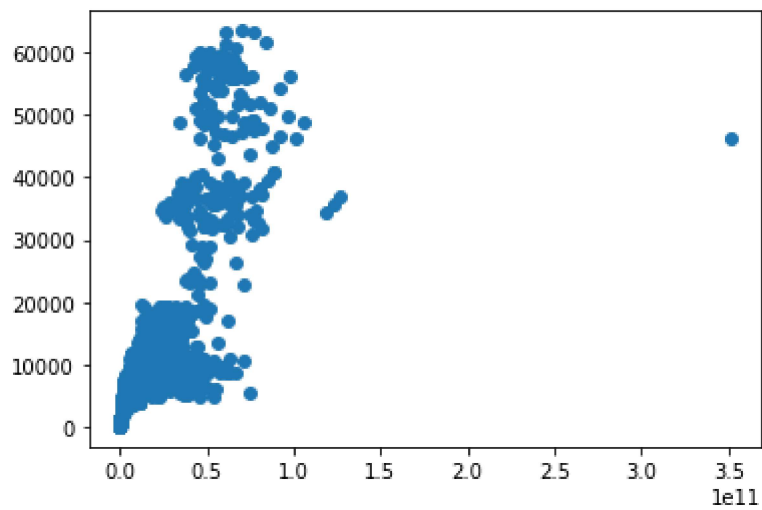


```
In [ ]:
```

```
In [ ]:
```

```
In [11]: import matplotlib.pyplot as plt
x = data['Volume']
y = data['Close']
plt.scatter(x.values, y.values)
```

```
Out[11]: <matplotlib.collections.PathCollection at 0x7fe158ea5970>
```



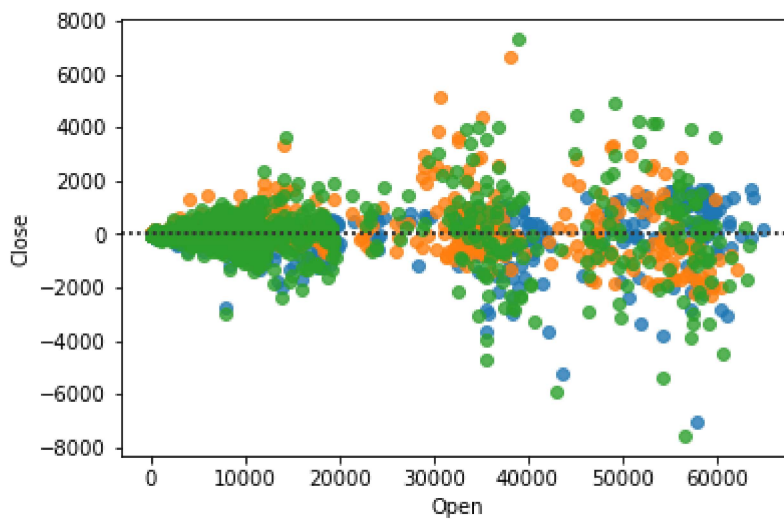
```
In [17]: output = data['Close']

sns.residplot(x=data["High"], y=output)

sns.residplot(x=data["Low"], y=output)

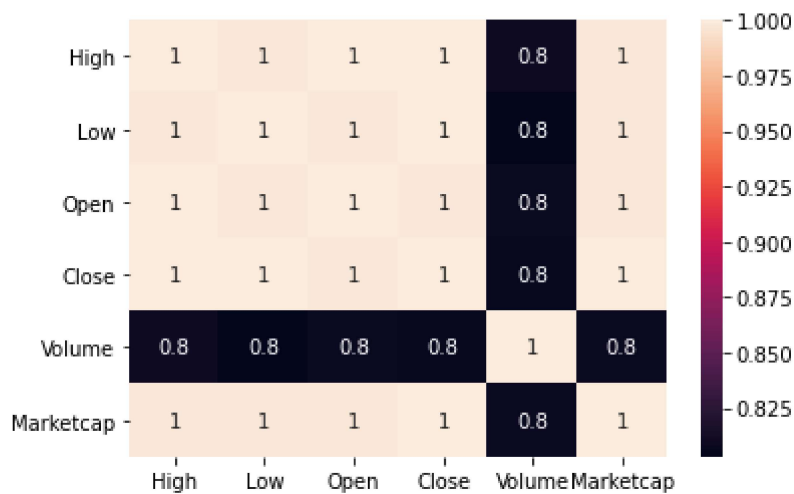
sns.residplot(x=data["Open"], y=output)

plt.show()
```



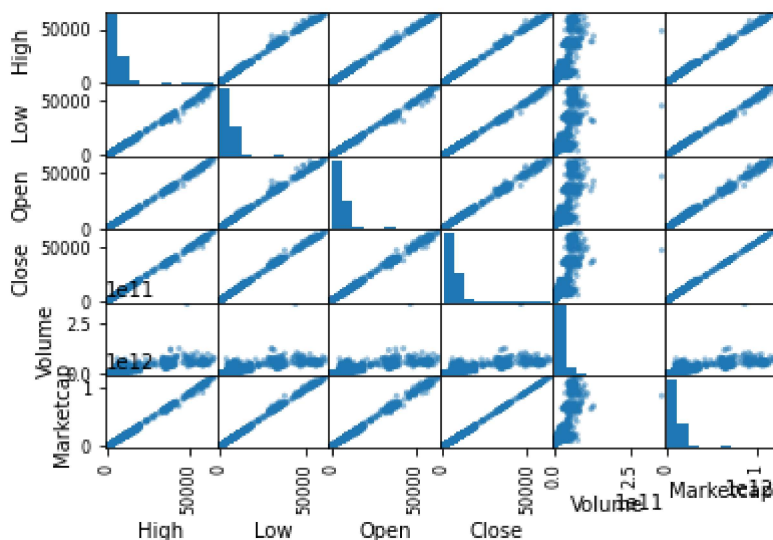
```
In [18]: sns.heatmap(data.corr(), annot = True, fmt='.1g')
```

```
Out[18]: <AxesSubplot:>
```



```
In [19]: from matplotlib import pyplot
from pandas import read_csv
from pandas.plotting import scatter_matrix

scatter_matrix(data)
pyplot.show()
```



```
In [24]: model = sm.formula.ols(formula= "Close ~ High + Low + Open + Marketcap + Volume", data=
multi_reg = model.fit()
print(multi_reg.summary())
```

```

OLS Regression Results
=====
Dep. Variable:          Close    R-squared:                1.000
Model:                  OLS      Adj. R-squared:           1.000
Method:                 Least Squares    F-statistic:          2.384e+06
Date:                   Tue, 29 Mar 2022    Prob (F-statistic):      0.00
Time:                   21:24:58    Log-Likelihood:        -19755.
No. Observations:        2991    AIC:                   3.952e+04
Df Residuals:            2985    BIC:                   3.956e+04
Df Model:                 5
Covariance Type:         nonrobust
=====
coef    std err          t      P>|t|      [0.025    0.975]
=====

```

```
-----
```

Intercept	110.4062	4.480	24.646	0.000	101.623	119.190
High	0.5386	0.012	46.152	0.000	0.516	0.561
Low	0.3113	0.009	33.722	0.000	0.293	0.329
Open	-0.3001	0.010	-28.853	0.000	-0.320	-0.280
Marketcap	2.396e-08	4.88e-10	49.108	0.000	2.3e-08	2.49e-08
Volume	-1.564e-09	3.07e-10	-5.097	0.000	-2.17e-09	-9.62e-10

```
=====
```

Omnibus:	1276.981	Durbin-Watson:	1.264
Prob(Omnibus):	0.000	Jarque-Bera (JB):	75069.035
Skew:	-1.215	Prob(JB):	0.00
Kurtosis:	27.422	Cond. No.	3.34e+11

```
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

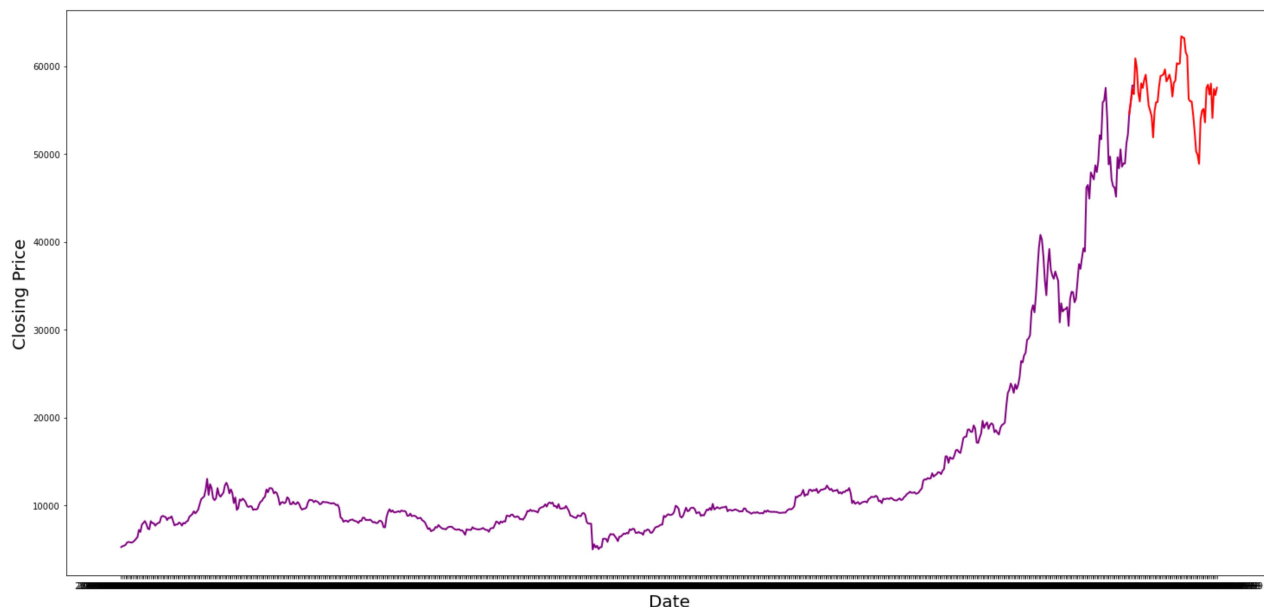
[2] The condition number is large, 3.34e+11. This might indicate that there are strong multicollinearity or other numerical problems.

```
In [48]: features = ['Open', 'High', 'Low', 'Volume', 'Marketcap']
output = 'Close'
x_train, x_test, y_train, y_test = train_test_split(data[features], data[output], test_si
model = LinearRegression()
model.fit(x_train, y_train)
model.score(x_test, y_test)
```

Out[48]: 0.9997603046952602

```
In [71]: shifteddata = data.shift(periods=60).tail(60)
predict = model.predict(shifteddata[features])
plt.figure(figsize = (25, 12))
plt.plot(data["Date"][-800:-117], data["Close"][-800:-117], color='purple', lw=2)
plt.plot(shifteddata["Date"], predict, color='red', lw=2)
plt.xlabel("Date", size=20)
plt.ylabel("Closing Price", size=20)
```

Out[71]: Text(0, 0.5, 'Closing Price')



In [40]:

```
2591    2020-06-02 23:59:59
2592    2020-06-03 23:59:59
2593    2020-06-04 23:59:59
2594    2020-06-05 23:59:59
2595    2020-06-06 23:59:59
      ...
2926    2021-05-03 23:59:59
2927    2021-05-04 23:59:59
2928    2021-05-05 23:59:59
2929    2021-05-06 23:59:59
2930    2021-05-07 23:59:59
Name: Date, Length: 340, dtype: object
```

In [ ]: