BITCOIN

-by G21

Terminologies

- BTC: The conventional symbol/letters used to denote Bitcoin.
- USD: The conventional symbol/letters used to denote the US Dollar.
- Volume: The total number of BTC traded in a given time frame

  OR

  The net worth of the total number of BTC traded in a given time frame.
- High: Highest Price of BTC (reported in USD) on the given date.
- Low: Lowest Price of BTC (reported in USD) on the given date.
- Open: Opening Price of BTC (reported in USD) on the given date.
- Close: Closing Price of BTC (reported in USD) on the given date.
- Return: Calculated on the basis of difference in closing prices of BTC on consecutive days.

Agenda

What this report covers

01 Data Collection

02 Data Preprocessing and Cleaning

03 Data Visualization

04 Data Statistics

05 Hypothesis Testing

Data Collection

Per minute, Per Hour and Per Day

01 Per Minute Data

(BTC_Minute ⟶ top 5 rows)

Unix Timestamp	Date	Symbol	Open	High	Low	Close	Volume
1586390400000	2020-04-09 00:00:00	BTCUSD	7369.600000	7369.600000	7348.470000	7350.410000	7.733528
1586390340000	2020-04-08 23:59:00	BTCUSD	7364.370000	7369.600000	7364.370000	7369.600000	1.310954
1586390280000	2020-04-08 23:58:00	BTCUSD	7360.890000	7367.180000	7360.890000	7364.370000	1.018774
1586390220000	2020-04-08 23:57:00	BTCUSD	7357.620000	7366.120000	7357.620000	7360.890000	0.026251
1586390160000	2020-04-08 23:56:00	BTCUSD	7370.710000	7370.710000	7357.620000	7357.620000	0.444867

02 Per Hour Data

(BTC_Hour ⟶ top 5 rows)

Unix Timestamp	Date	Symbol	Open	High	Low	Close	Volume
1586390400000	2020-04-09 00:00:00	BTCUSD	7369.600000	7369.600000	7338.230000	7338.230000	7.788915
1586386800000	2020-04-08 23:00:00	BTCUSD	7367.280000	7398.920000	7348.470000	7369.600000	66.558293
1586383200000	2020-04-08 22:00:00	BTCUSD	7337.190000	7390.250000	7322.870000	7367.280000	88.486108
1586379600000	2020-04-08 21:00:00	BTCUSD	7327.700000	7370.000000	7311.330000	7337.190000	35.551553
1586376000000	2020-04-08 20:00:00	BTCUSD	7320.110000	7375.720000	7313.660000	7327.700000	153.655811

03 Per Day Data

(BTC_Day ⟶ top 5 rows)

Date	Symbol	Open	High	Low	Close	Volume BTC	Volume USD
2020-04-10	BTCUSD	7315.250000	7315.250000	7315.250000	7315.250000	0.000000	0.000000
2020-04-09	BTCUSD	7369.600000	7378.850000	7115.040000	7315.250000	2237.130000	16310014.530000
2020-04-08	BTCUSD	7201.810000	7432.230000	7152.800000	7369.600000	2483.600000	18138080.270000
2020-04-07	BTCUSD	7336.960000	7468.420000	7078.000000	7201.810000	2333.340000	17047120.320000
2020-04-06	BTCUSD	6775.210000	7369.760000	6771.010000	7336.960000	3727.470000	26533750.170000

Data Collection

Shape and Information related to the datasets

Shape of the Dataset

Name of the Dataset	Shape of the Dataset
BTC_Minute	(2283519, 8)
BTC_Day	(1647, 8)
BTC_Hour	(39465, 8)

Information related to the Dataset

01 BTC_Minute

```
RangeIndex: 2283519 entries, 0 to 2283518
Data columns (total 8 columns):
 #   Column          Dtype
---  ------          -----
 0   Unix Timestamp  int64
 1   Date            object
 2   Symbol          object
 3   Open            float64
 4   High            float64
 5   Low             float64
 6   Close           float64
 7   Volume          float64
dtypes: float64(5), int64(1), object(2)
```

02 BTC_Hour

```
RangeIndex: 39465 entries, 0 to 39464
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Unix Timestamp  39465 non-null  int64
 1   Date            39465 non-null  object
 2   Symbol          39465 non-null  object
 3   Open            39465 non-null  float64
 4   High            39465 non-null  float64
 5   Low             39465 non-null  float64
 6   Close           39465 non-null  float64
 7   Volume          39465 non-null  float64
dtypes: float64(5), int64(1), object(2)
```

03 BTC_Day

```
RangeIndex: 1647 entries, 0 to 1646
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Date         1647 non-null   object
 1   Symbol       1647 non-null   object
 2   Open         1647 non-null   float64
 3   High         1647 non-null   float64
 4   Low          1647 non-null   float64
 5   Close        1647 non-null   float64
 6   Volume BTC   1647 non-null   float64
 7   Volume USD   1647 non-null   float64
dtypes: float64(6), object(2)
```

Data Preprocessing and Cleaning

01

Dealing with Missing Values

When the dataset has fewer missing values, the row with Nan values can be dropped.

02

Conversion of DataType

03

Sorting of the Dataset

Sorting the dataset according to its "Date" for future use.

04

Setting Date Column as ColumnIndex of the Dataset

05

Removing unwanted columns

Data Preprocessing and Cleaning

Dealing with the Missing Values of the datasets

01

Calculating number of missing values present in the datasets

The datasets contain no missing/Nan values

BTC_Minute

BTC_Hour

```
Unix Timestamp    0
Date              0
Symbol            0
Open              0
High              0
Low               0
Close             0
Volume            0
dtype: int64
```

BTC_Day

```
Date          0
Symbol        0
Open          0
High          0
Low           0
Close         0
Volume BTC    0
Volume USD    0
dtype: int64
```

02

Removing row with column value equal to 0

```
Date          2020-04-10
Symbol            BTCUSD
Open             7315.25
High             7315.25
Low              7315.25
Close            7315.25
Volume BTC           0.0
Volume USD           0.0
```

BTC_Day dataset: Removing row with date "2020-04-10"

Data Preprocessing and Cleaning

Conversion of Data-Type, Sorting, Setting Date Column as ColumnIndex of the datasets

03 Conversion of object to datetime

```
RangeIndex: 1647 entries, 0 to 1646
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Date        1647 non-null   object
 1   Symbol      1647 non-null   object
 2   Open        1647 non-null   float64
 3   High        1647 non-null   float64
 4   Low         1647 non-null   float64
 5   Close       1647 non-null   float64
 6   Volume BTC  1647 non-null   float64
 7   Volume USD  1647 non-null   float64
dtypes: float64(6), object(2)
```

```
RangeIndex: 1647 entries, 0 to 1646
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Date        1647 non-null   datetime64[ns]
 1   Symbol      1647 non-null   object
 2   Open        1647 non-null   float64
 3   High        1647 non-null   float64
 4   Low         1647 non-null   float64
 5   Close       1647 non-null   float64
 6   Volume BTC  1647 non-null   float64
 7   Volume USD  1647 non-null   float64
dtypes: datetime64[ns](1), float64(6), object(1)
```

04 Sorting of the dataset

05 Setting Date column as ColumnIndex

	Symbol	Open	High	Low	Close	Volume BTC	Volume USD
Date							
2015-10-08	BTCUSD	242.50	245.00	242.5	243.95	18.80	4595.84
2015-10-09	BTCUSD	243.95	249.97	243.6	245.39	30.99	7651.63
2015-10-10	BTCUSD	245.39	246.30	244.6	246.30	12.17	2984.44
2015-10-11	BTCUSD	246.30	249.50	246.3	249.50	12.22	3021.12
2015-10-12	BTCUSD	249.50	249.50	247.6	247.60	38.28	9493.89

Data Preprocessing and Cleaning

Removing unwanted columns

06

Removing unwanted columns

Date	Open	High	Low	Close	Volume BTC	Volume USD
2015-10-11	244.535000	247.692500	244.250000	246.285000	18.545000	4563.257500
2015-10-18	256.031429	263.710000	254.200000	258.648571	175.141429	46461.315714
2015-10-25	275.254286	281.348571	273.404286	278.242857	337.531429	93461.344286
2015-11-01	306.202857	323.670000	293.718571	312.005714	939.355714	293904.180000
2015-11-08	378.771429	417.978571	359.361429	385.485714	1708.240000	670326.267143

The visualization are mostly based on the BTC_Day dataset

VISUALIZATION

01

Daily analysis of "Close" Prices of Bitcoin

From the graph, one can visualize that at the end of every year the price of bitcoin is decreasing.

02

RETURN

Percentage Change in "Close" price

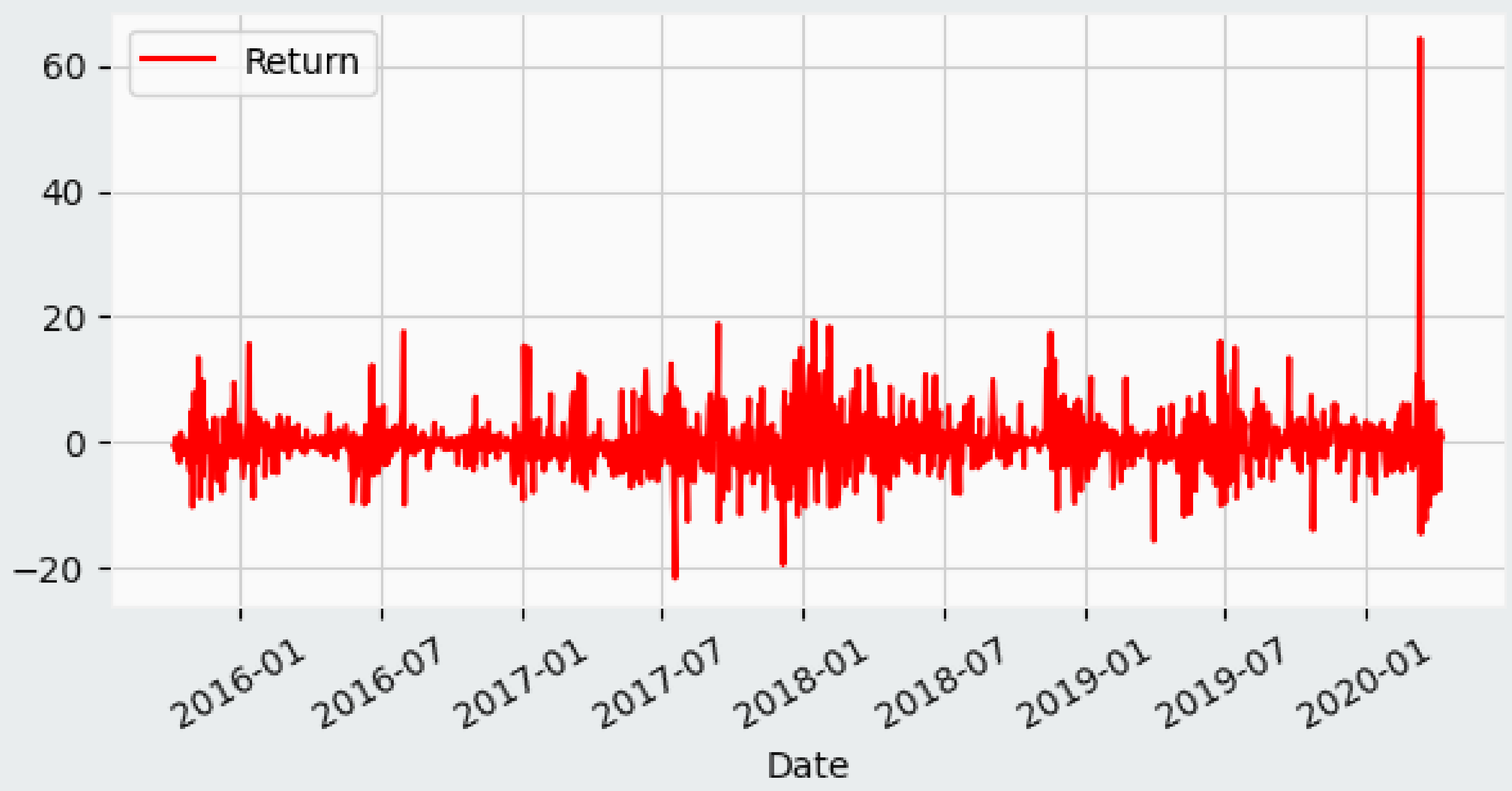
$$r_i = \frac{p_i - p_j}{p_j}$$

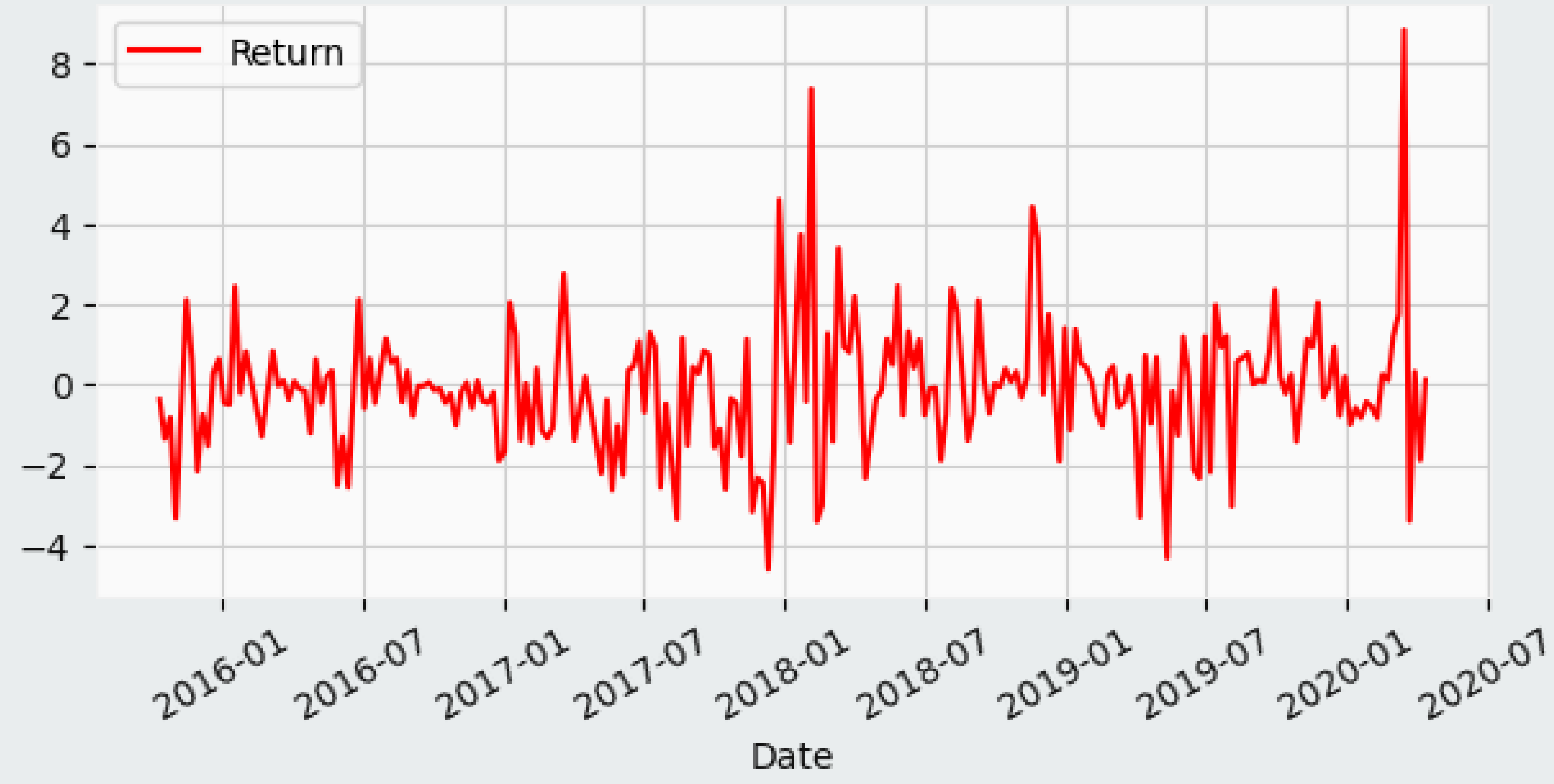
- Daily Return Datasheet

Date	Symbol	Open	High	Low	Close	Volume BTC	Volume USD	Return
2015-10-12 00:00:00	BTCUSD	249.500000	249.500000	247.600000	247.600000	38.280000	9493.890000	-0.960000
2015-10-11 00:00:00	BTCUSD	246.300000	249.500000	246.300000	249.500000	12.220000	3021.120000	0.767367
2015-10-10 00:00:00	BTCUSD	245.390000	246.300000	244.600000	246.300000	12.170000	2984.440000	-1.282565
2015-10-09 00:00:00	BTCUSD	243.950000	249.970000	243.600000	245.390000	30.990000	7651.630000	-0.369468
2015-10-08 00:00:00	BTCUSD	242.500000	245.000000	242.500000	243.950000	18.800000	4595.840000	-0.586821

- Weekly Return Datasheet

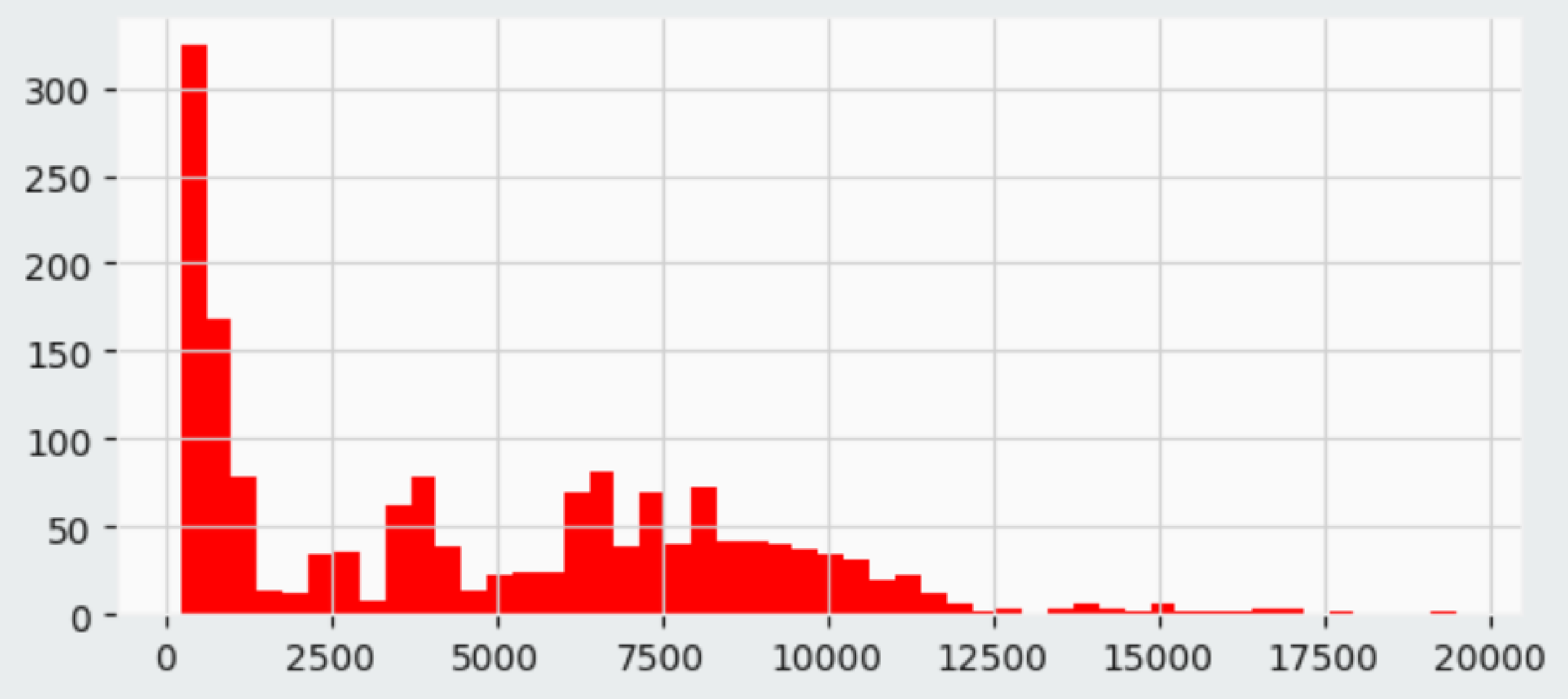
Date	Open	High	Low	Close	Volume BTC	Volume USD	Return
2015-10-11	244.535	247.6925	244.25	246.285	18.545	4563.2575	-0.367872
2015-10-18	256.031429	263.71	254.2	258.648571	175.141429	46461.31571	-1.347957
2015-10-25	275.254286	281.348571	273.404286	278.242857	337.531429	93461.34429	-0.810194
2015-11-01	306.202857	323.67	293.718571	312.005714	939.355714	293904.18	-3.321859
2015-11-08	378.771429	417.978571	359.361429	385.485714	1708.24	670326.2671	-0.407066

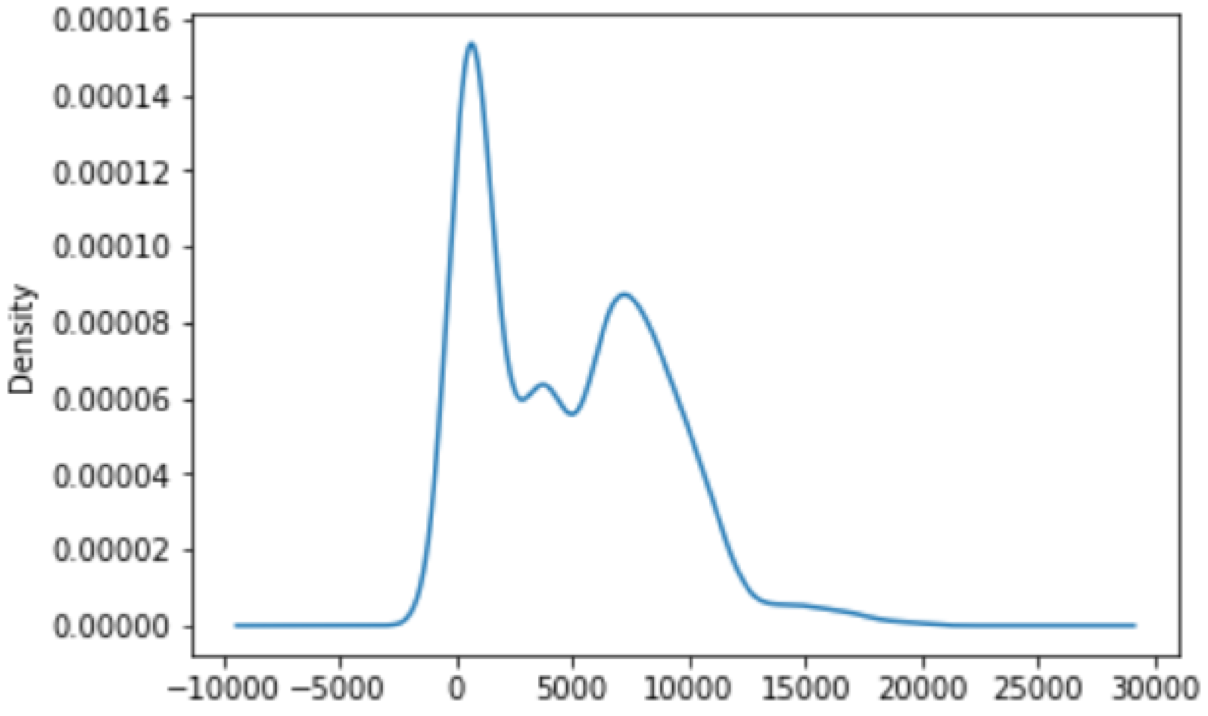




CLOSE

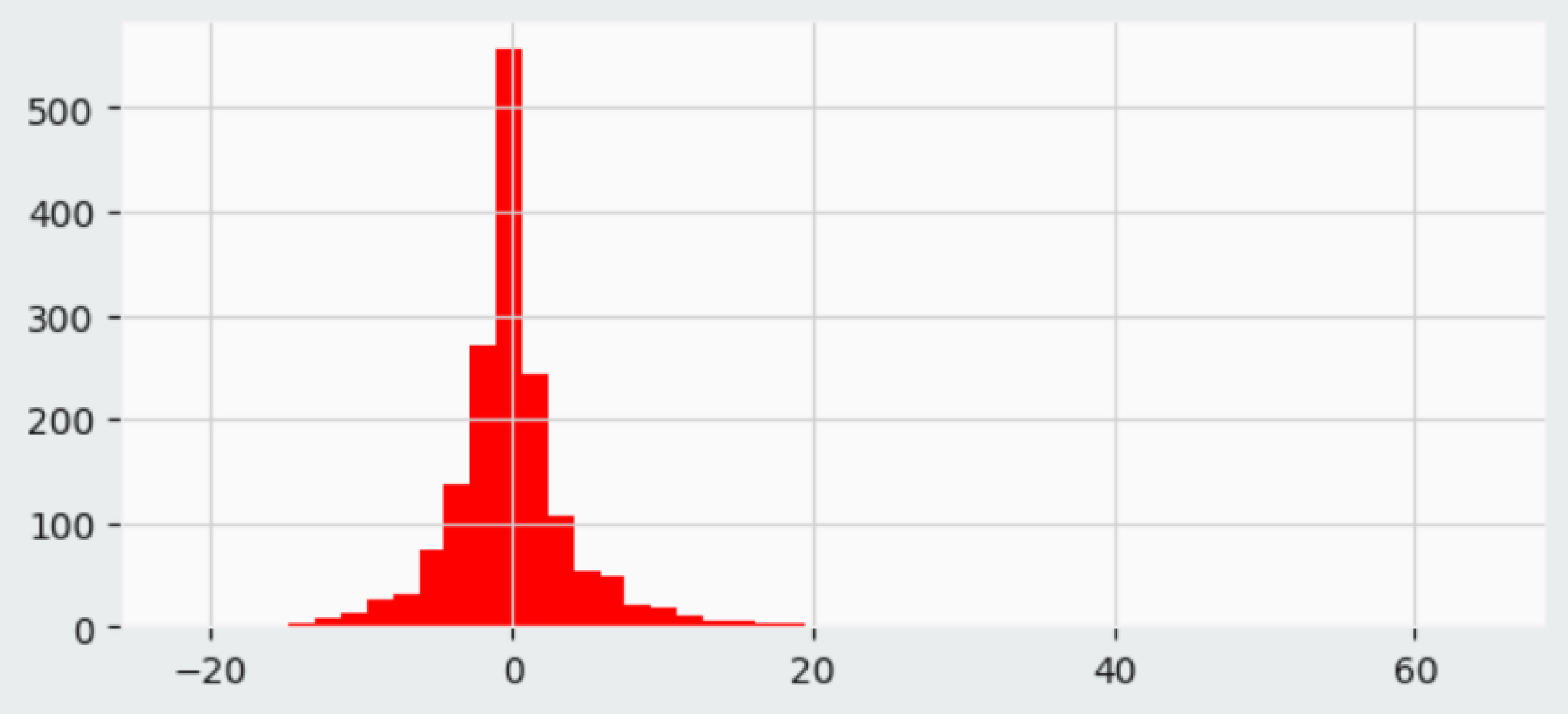
Histogram and Density plot of "Close" price

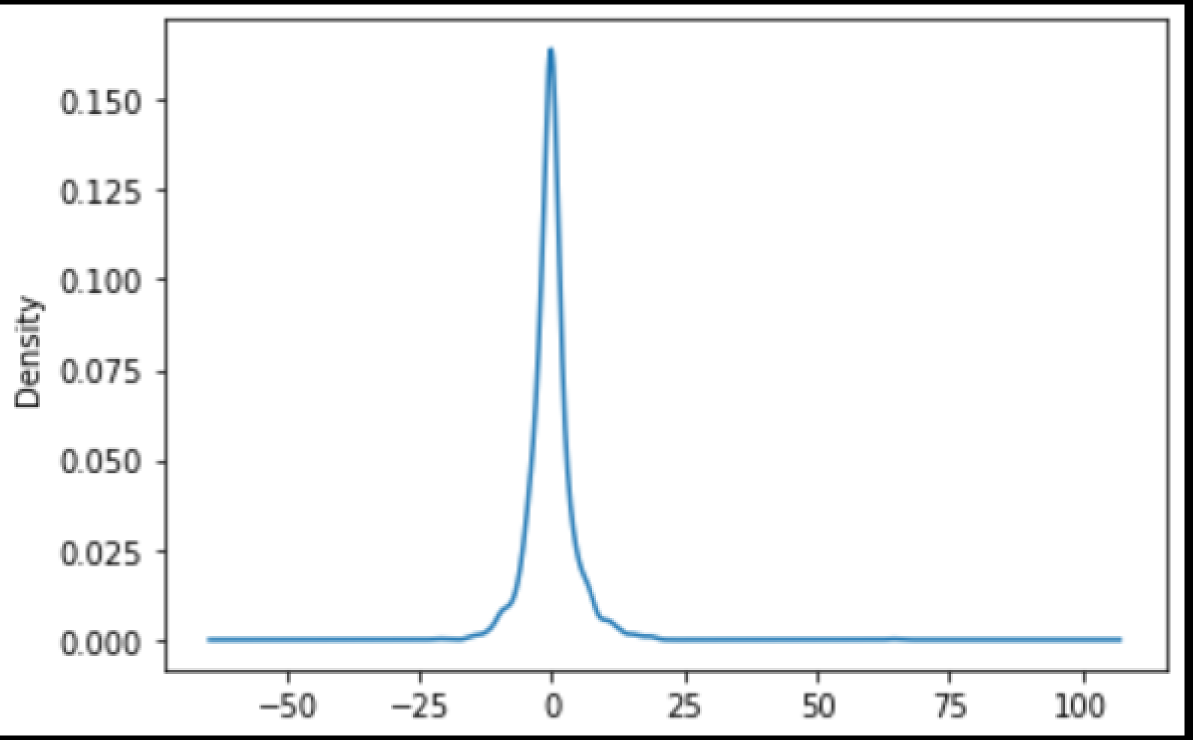




RETURN

Histogram and Density plot of "Return"





STATISTICS

Per minute, Per Hour and Per Day

[dataset.describe()]

01 Per Minute Data

	Open	High	Low	Close	Volume
count	1.37E+06	1.37E+06	1.37E+06	1.37E+06	1.37E+06
mean	6.29E+03	6.29E+03	6.29E+03	6.29E+03	4.40E+00
std	3.68E+03	3.69E+03	3.68E+03	3.68E+03	2.73E+01
min	2.36E+02	2.43E+02	2.36E+02	2.36E+02	2.00E-10
25%	3.53E+03	3.53E+03	3.52E+03	3.53E+03	5.59E-02
50%	6.53E+03	6.54E+03	6.53E+03	6.53E+03	5.01E-01
75%	8.68E+03	8.68E+03	8.67E+03	8.68E+03	3.16E+00
max	2.00E+04	2.00E+04	2.00E+04	2.00E+04	8.26E+03

02 Per Hour Data

	Open	High	Low	Close	Volume
count	39465	39465	39465	39465	39465
mean	4817.778638	4844.889457	4788.678433	4817.963371	158.393961
std	3925.257801	3954.058698	3893.325248	3925.202322	300.774291
min	235.3	243.6	221.7	243.6	9.25436
25%	741.74	743	740.02	741.8	18.164575
50%	4142.12	4165.38	4118.39	4142.59	62.790643
75%	7908.48	7950	7860.75	7908.48	175.504078
max	19869.86	19999	19778.12	19869.86	8526.751048

03 Per Day Data

	Open	High	Low	Close	Volume BTC	Volume USD
count	1647	1647	1647	1647	1647	1.65E+03
mean	4823.493224	4971.083673	4656.335337	4827.596515	3789.357365	2.13E+07
std	3937.740533	4078.52735	3760.33516	3936.503969	4436.695998	3.69E+07
min	242.5	245	236	243.95	12.17	2.98E+03
25%	741.975	751.83	732.03	742.26	1028.94	1.66E+06
50%	4147.1	4295.09	4000	4154.84	2263.37	9.40E+06
75%	7920.51	8139.75	7630.445	7920.51	4918.785	2.46E+07
max	19499.99	19999	18870	19499.99	49229.15	5.61E+08

Volume-Weighted Average Price (VWAP)

- VWAP computes the average based on how many units were bought and sold at various prices throughout the period divided by the total number of units transacted.
- Traders often use VWAP to plan their entry and exit.

VWAP Formula: -

Typical Price(TP) = (High + Low + Close)/3

VWAP = (TP_1 * V_1 + TP_2 * V_2 + TP_n * V_n)/n

where,

V = volume for that period

n = total volume

Typical Price

$$\mathrm{VWAP} = \frac{\sum \mathrm{Price} * \mathrm{Volume}}{\sum \mathrm{Volume}}$$

Variation of VWAP w.r.t Minutes and Hours on 08-04-2020

Simple Moving Average (SMA) and Exponential Moving Average (EMA)

SMA calculates an average of the last n prices, where Ax represents the price in a period, and n represents the number of periods.

SMA = (A1 + A2 + ...An) / n

"**A**" is the average in period n

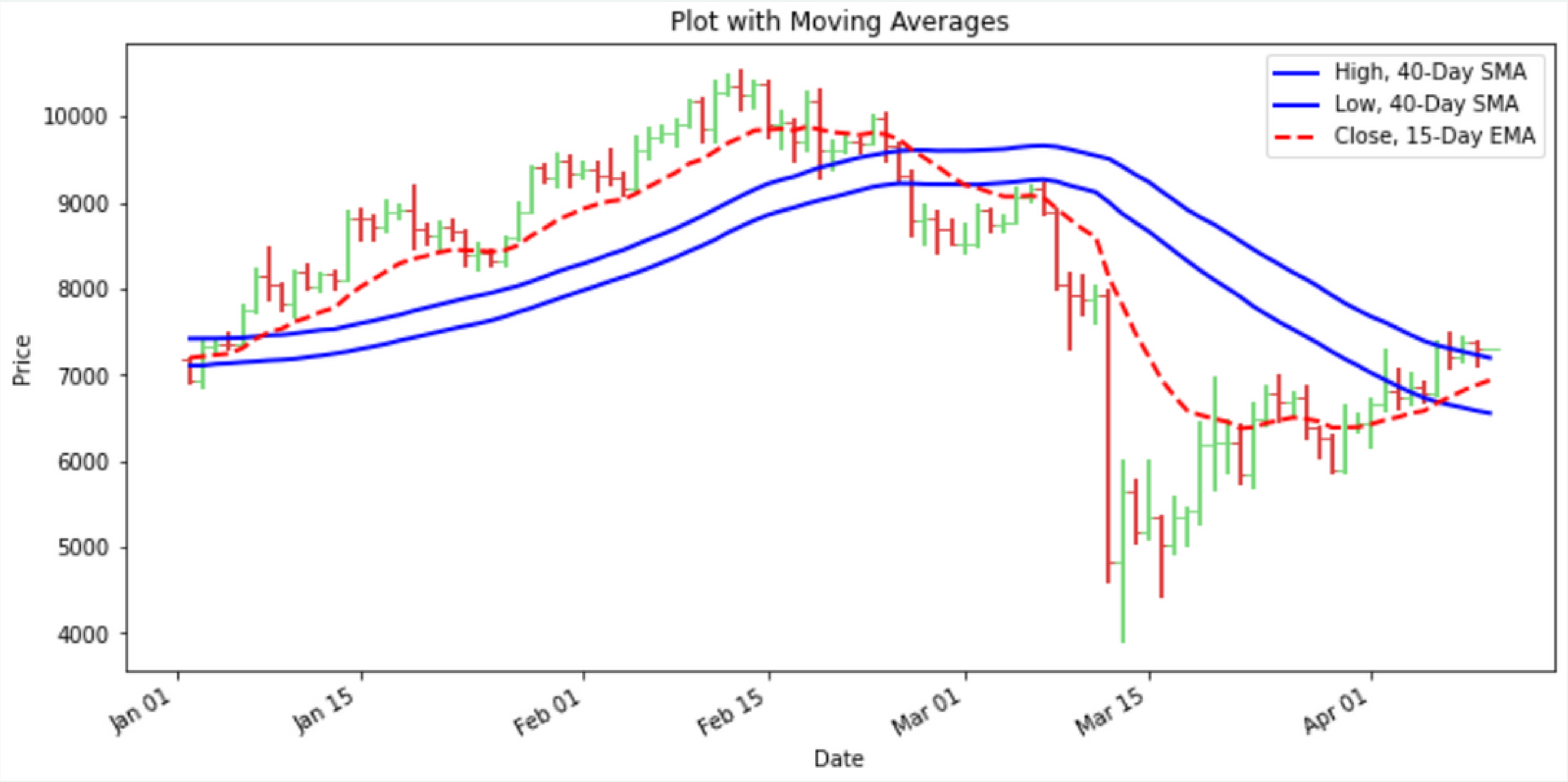
"**n**" is the number of periods

EMA is a weighted average of recent period's prices. It uses an exponentially decreasing weight from each previous price/period.

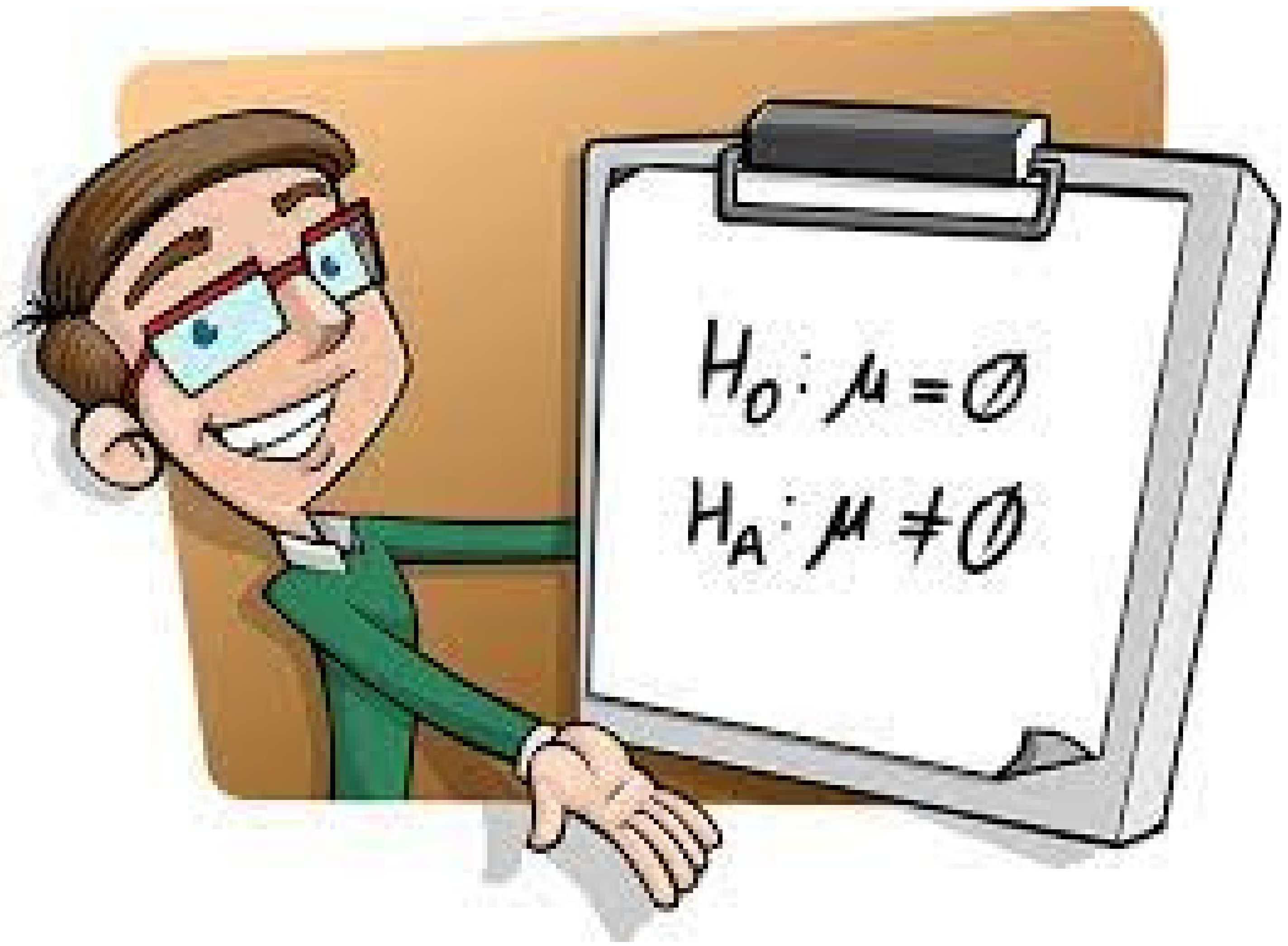
EMA = (2 / n+1) x (Close – Previous EMA) + Previous EMA

Visualization of SMA and EMA

- Buy BTC only when the price bars are completely above the higher 40-Day SMA.
- Sell BTC only when the price bars are completely below the lower 40-Day SMA.
- We do not enter any position (we keep flat on the market) when the prices are between the two 40-Day SMAs, or the last bar is crossing any of them.



HYPOTHESIS TESTING

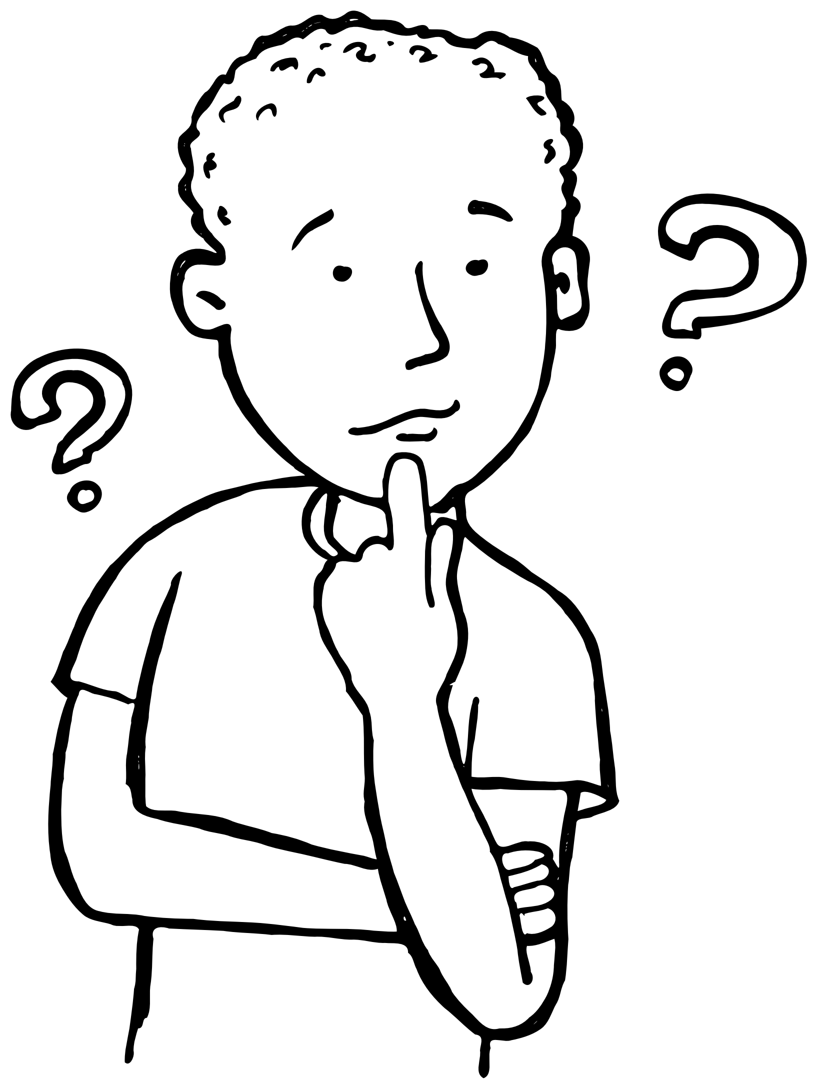


Bitcoin prices dataset is a 'time series' data.

Issues with the dataset:

- Noisiness
- Non-stationarity

Non-stationarity stands for a phenomenon of data distribution changing over time.



```
ADF Statistic: -1.7112456711382251
n_lags: 0.42532788151533524
p-value: 0.42532788151533524
Critial Values:
   1%, -3.4343880265995215
Critial Values:
   5%, -2.8633235546096194
Critial Values:
   10%, -2.56771952639493
```

Augmented Dickey-Fuller (ADF) test is a statistical tool for detecting presence of the unit root .
It can be used to identify crucial properties of a time series, such as non-stationarity or economic bubbles.

H_0: The given time series data of bitcoin prices is non - stationary against the alternative hypothesis

Ha: The given time series data of bitcoin prices is stationary.

Why did we test the time series for non-stationarity?

A glimpse into the near future

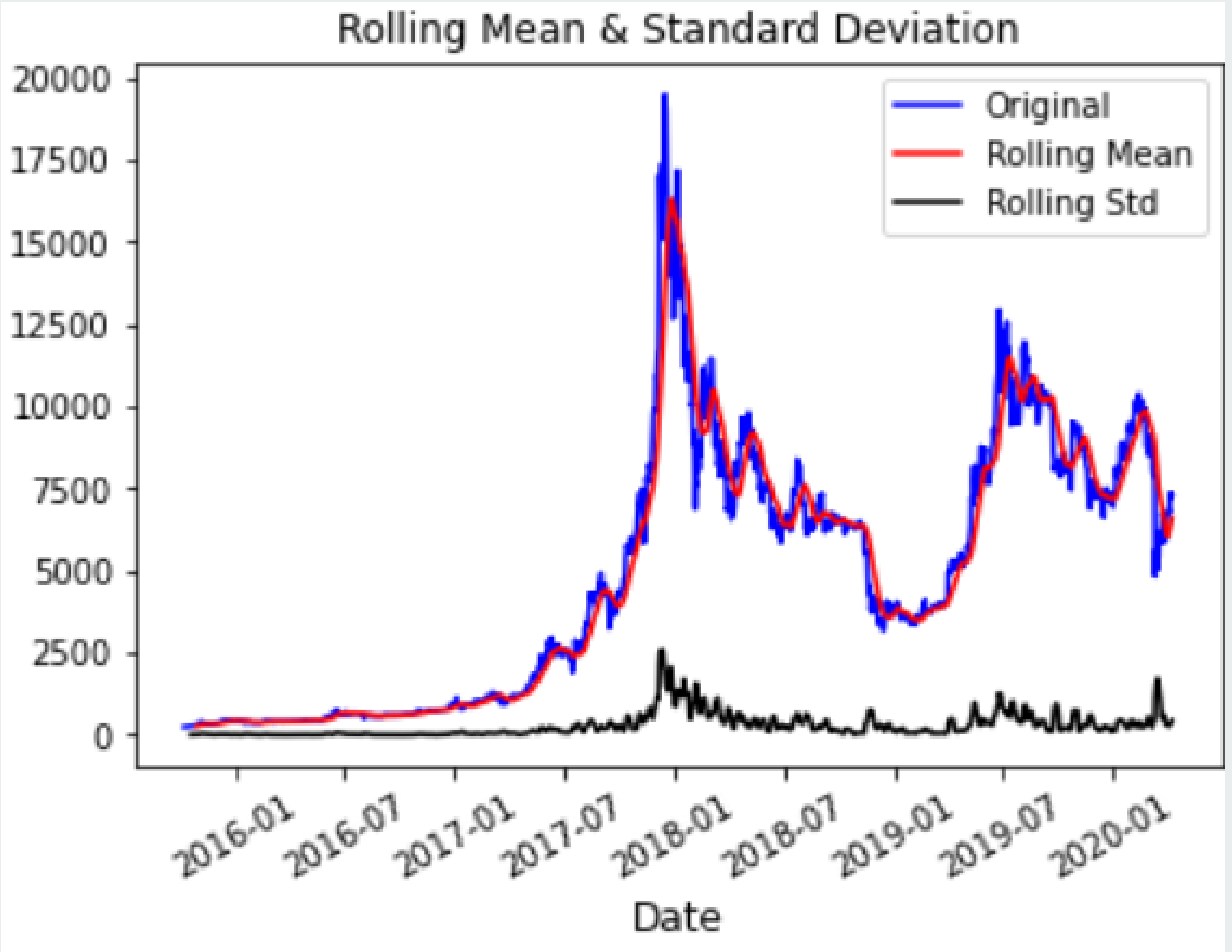
- **Testing various models**
- **Short Term prediction of bitcoin prices**
- **Feature Testing**
- **Local fluctuations in the time-series**

PREDICTION TASK

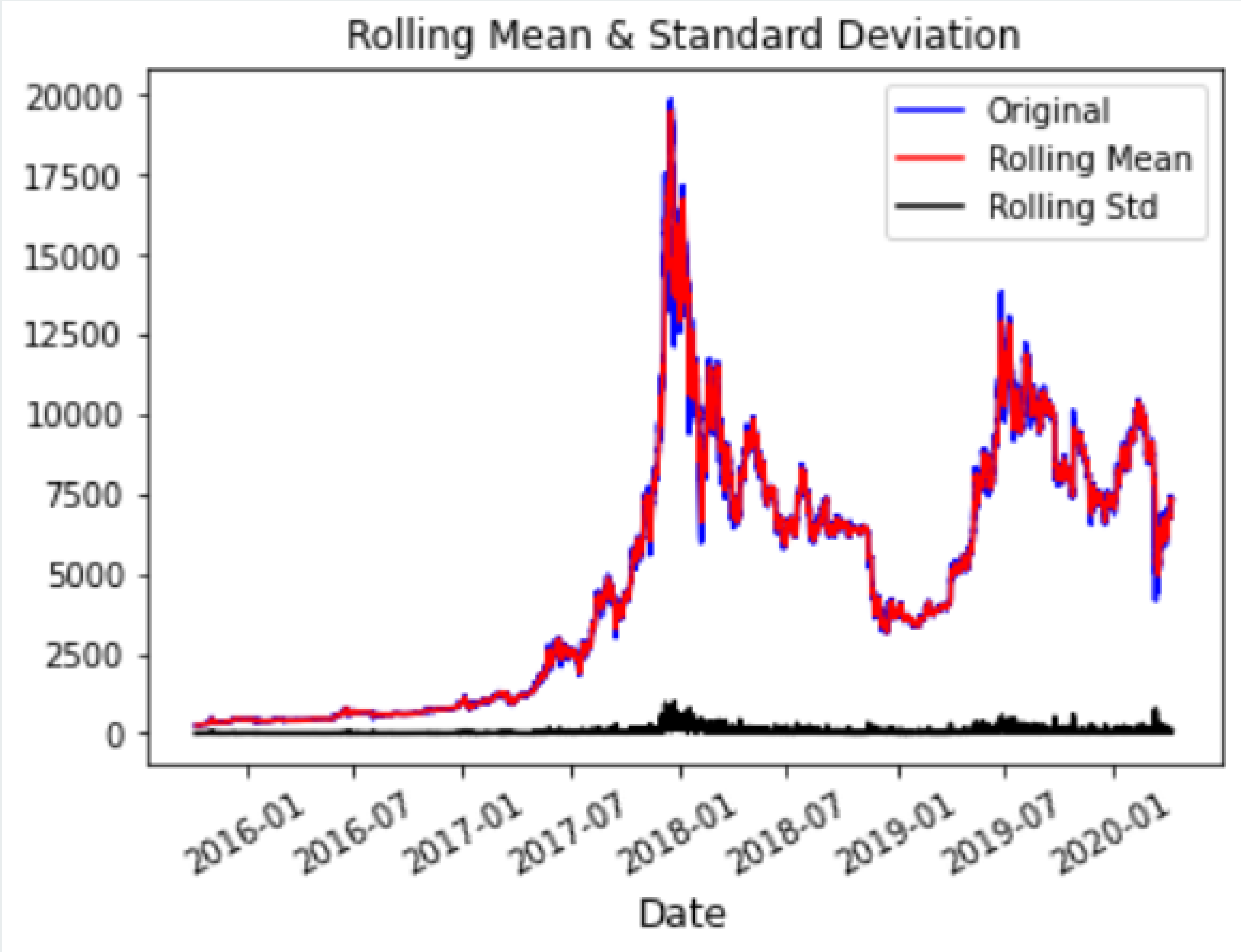
Plotted Time Series Curve

Testing the Stationarity

For BTC_Day Dataset



For BTC_Hour Dataset



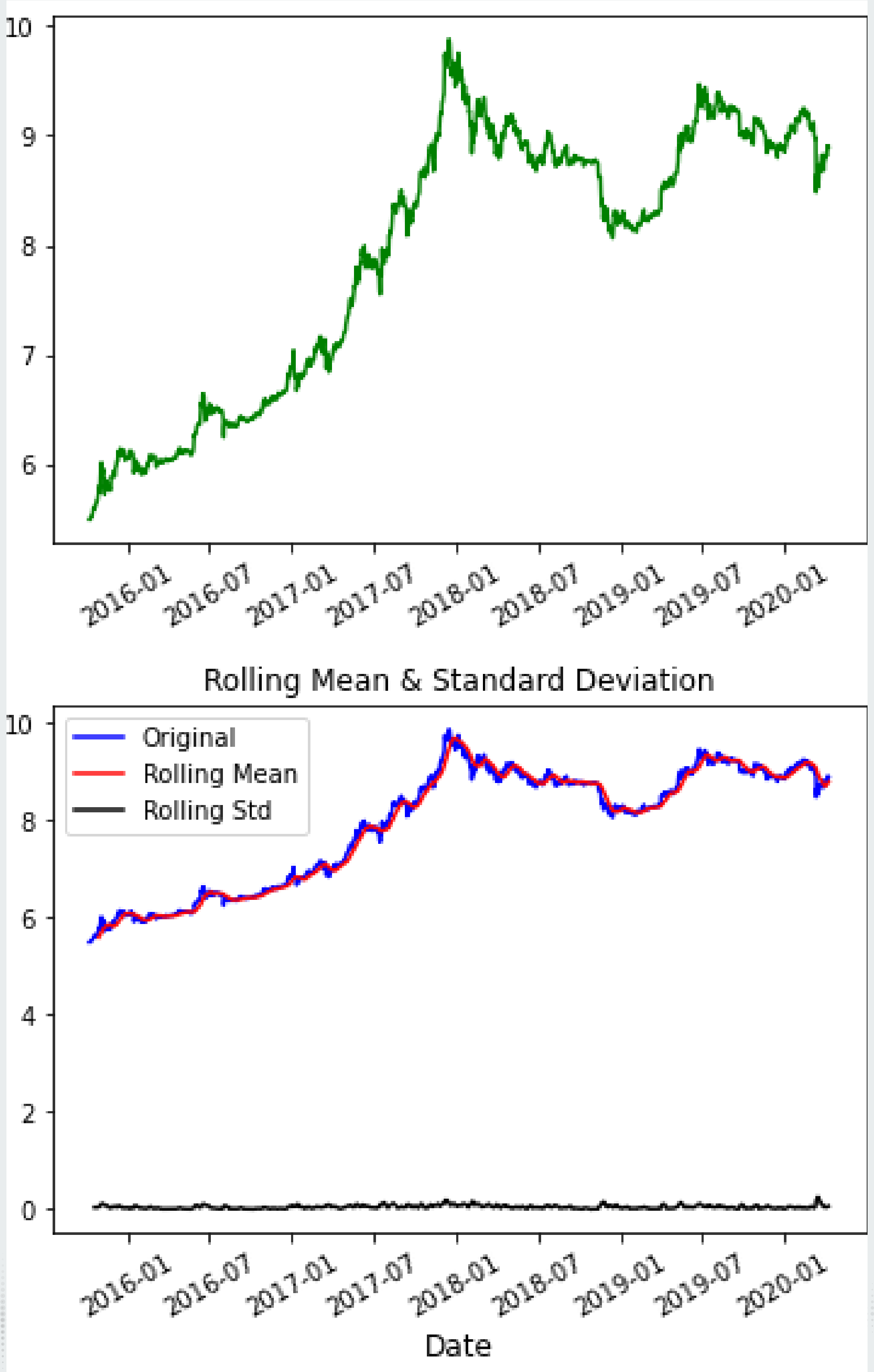
Since the p value is greater than 0.05, the time series is non stationary

Log Transforming the Series

Log transformation is used to unskew highly skewed data. Thus helping in forecasting process.

For BTC_Day Dataset

```
ADF Stastistic: -1.907645
p-value: 0.328473
The graph is non stationary
Critical values:
          1%: -3.434
          5%: -2.863
          10%: -2.568
```

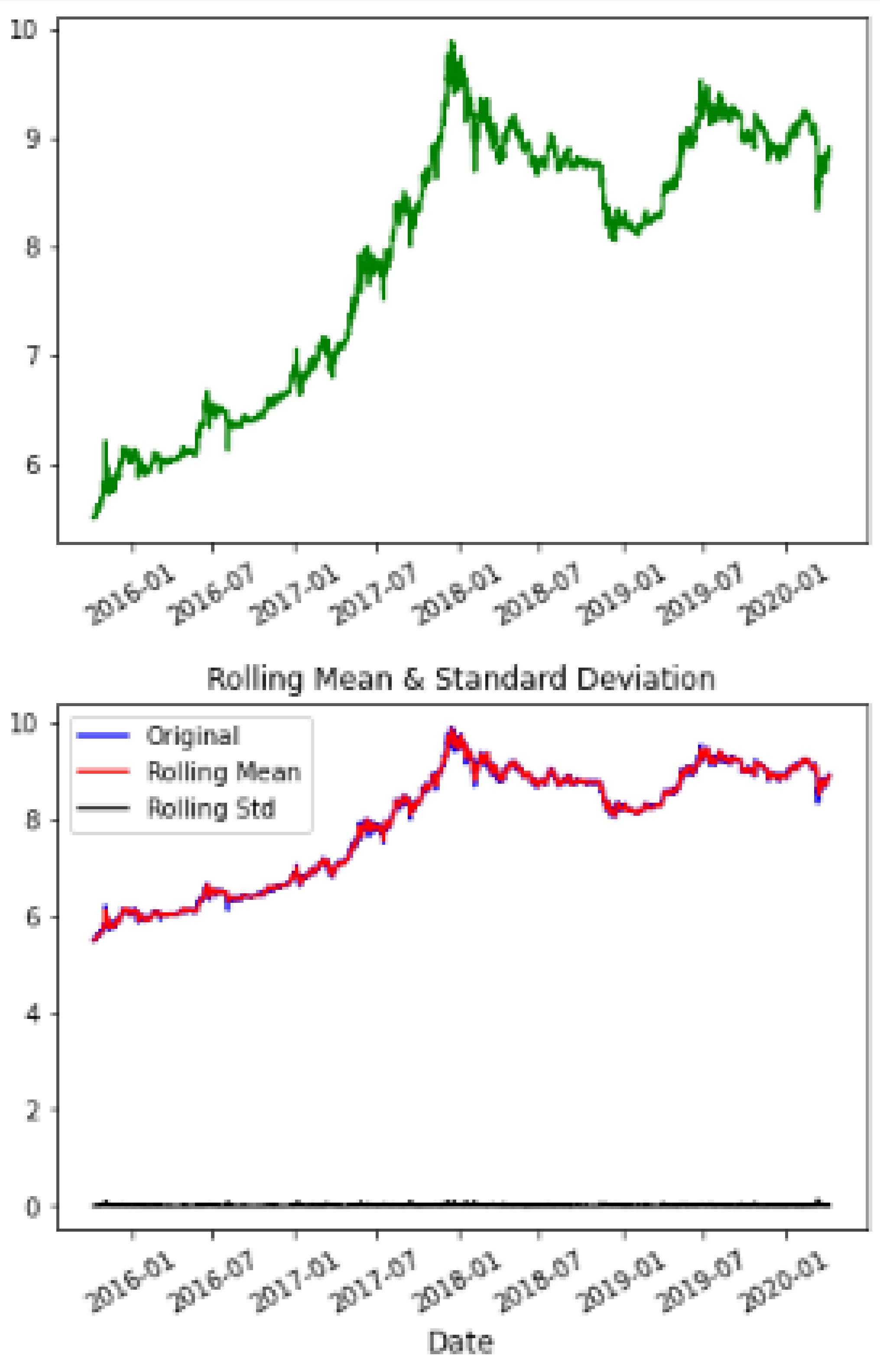


Log Transforming the Series

Log transformation is used to unskew highly skewed data.
Thus helping in forecasting process.

For BTC_Hour Dataset

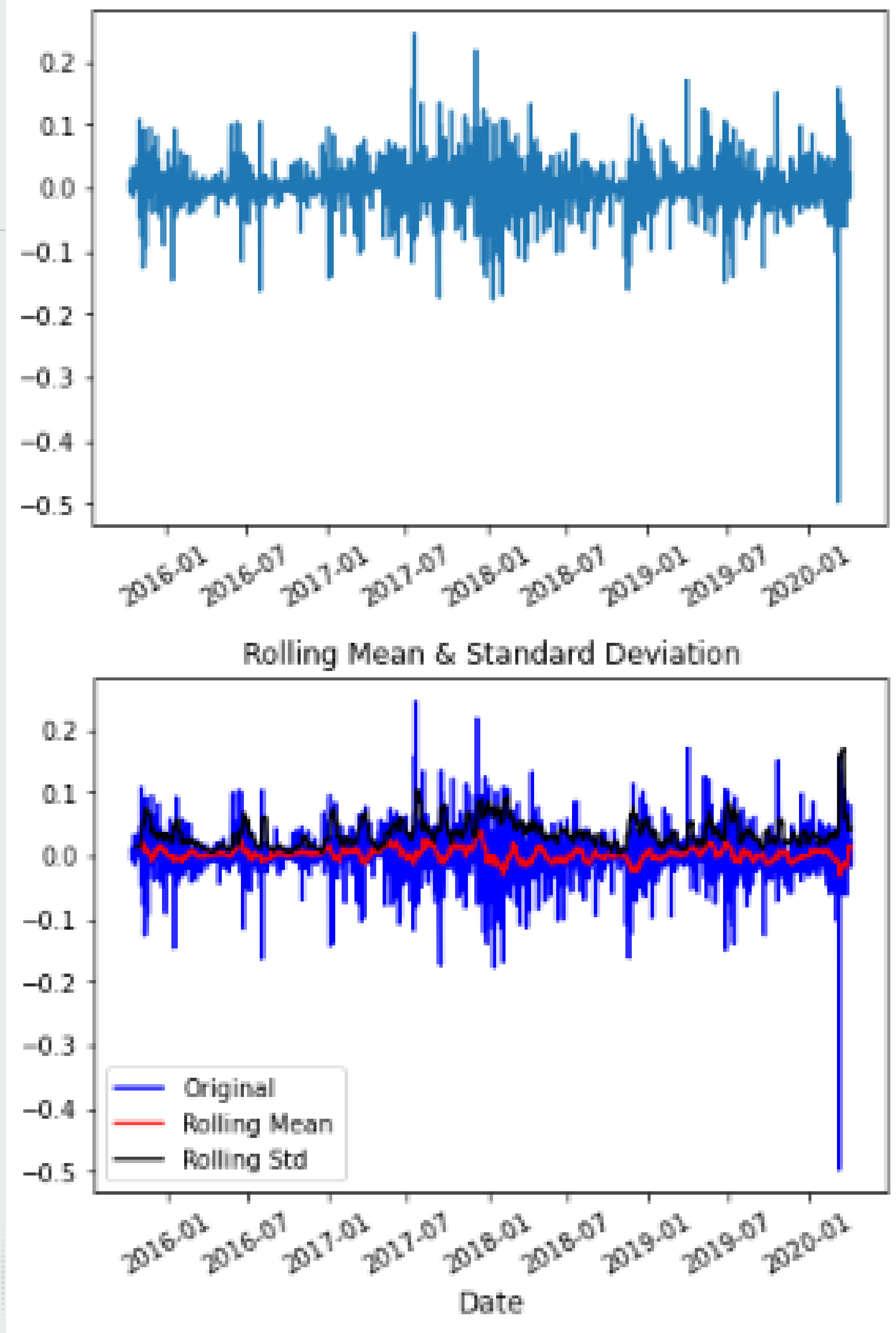
```
ADF Stastistic: -2.018345
p-value: 0.278588
The graph is non stationary
Critical values:
        1%: -3.431
        5%: -2.862
        10%: -2.567
```



Remove Trend and Seasonality with Differencing

For BTC_Day Dataset

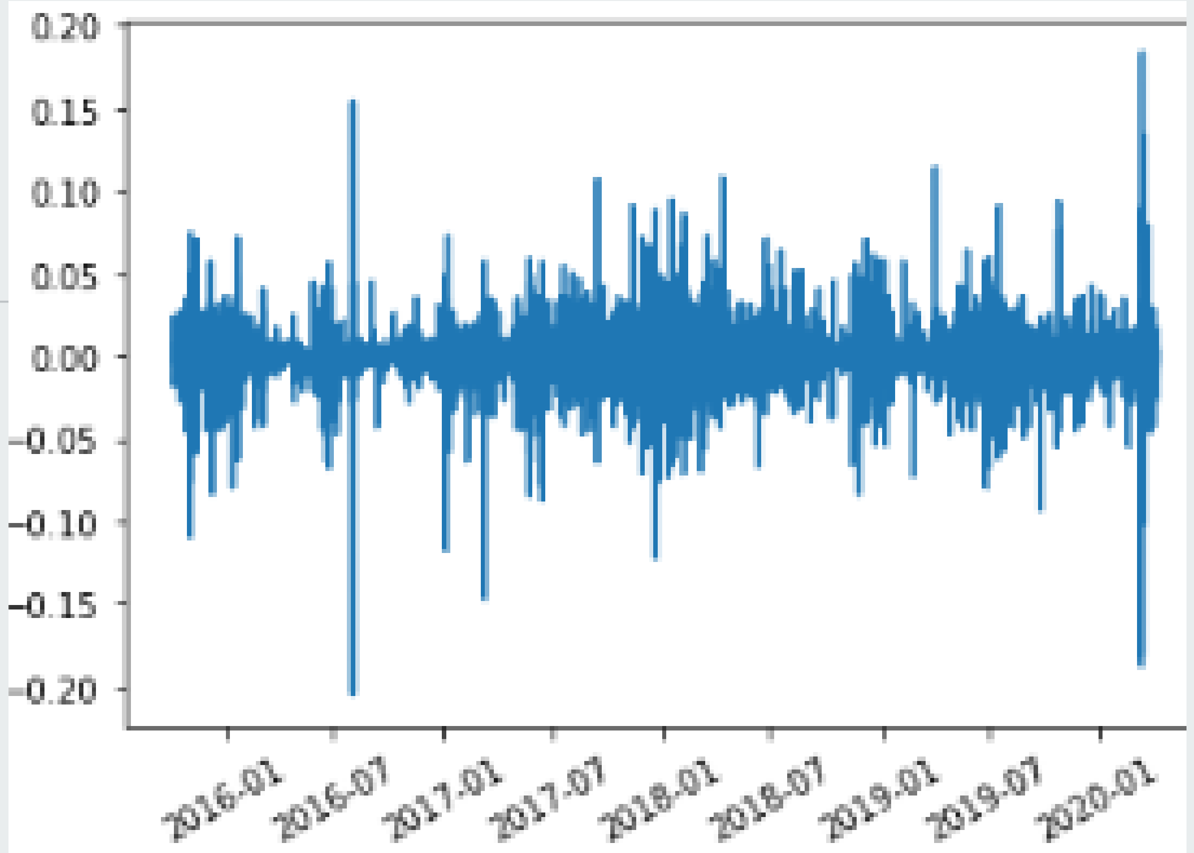
```
ADF Stastistic: -28.291281
p-value: 0.000000
The graph is stationary
Critical values:
        1%: -3.434
        5%: -2.863
        10%: -2.568
```

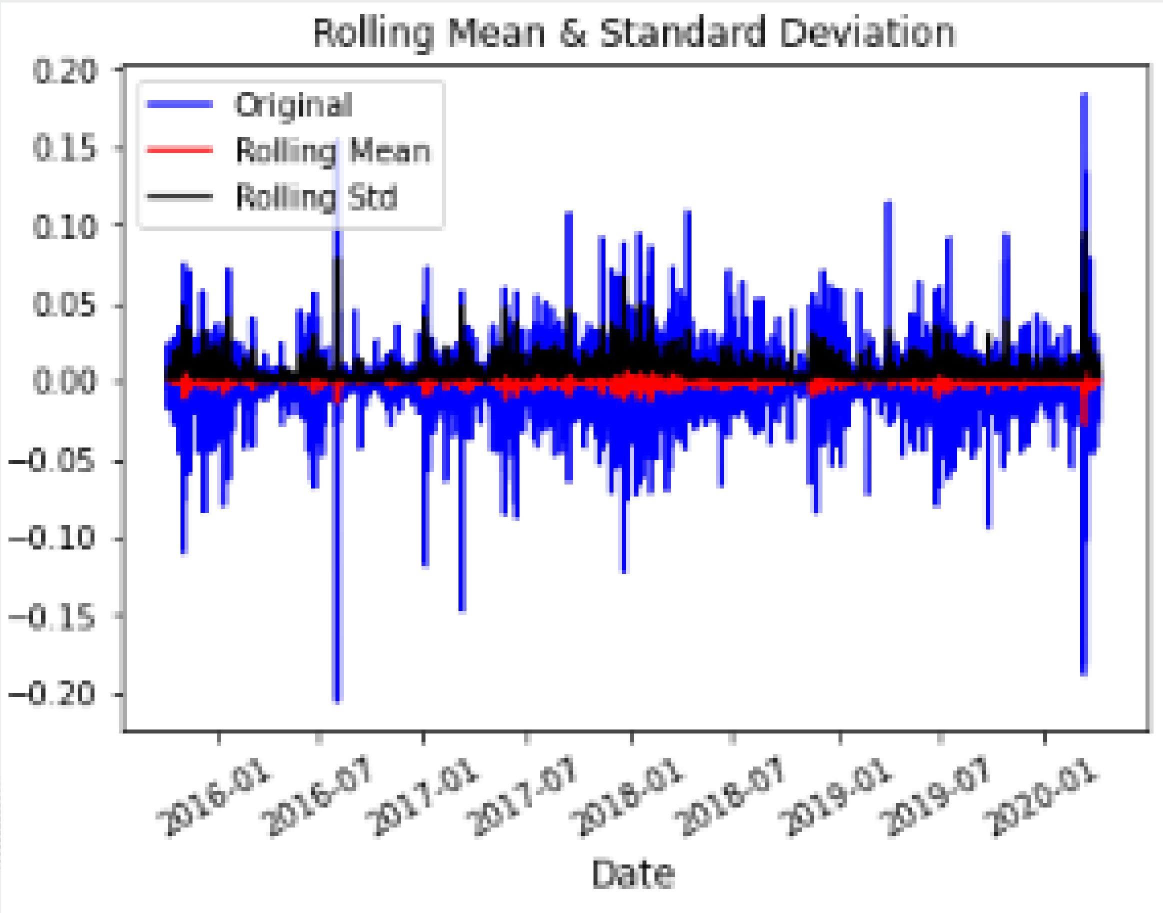


Remove Trend and Seasonality with Differencing

For BTC_Hour Dataset

```
ADF Stastistic: -32.125981
p-value: 0.000000
The graph is stationary
Critical values:
        1%: -3.431
        5%: -2.862
        10%: -2.567
```





Time Series Forecasting Models

Forecasting Models and RSS

- Auto-Regressive Model: Auto regressive model is a time series forecasting model where current values are dependent on past values.
- Moving Average Model: In moving average model the series is dependent on past error terms.
- ARIMA: It is a combination of both AR and MA models. It makes the time series stationary by itself through the process of differencing.

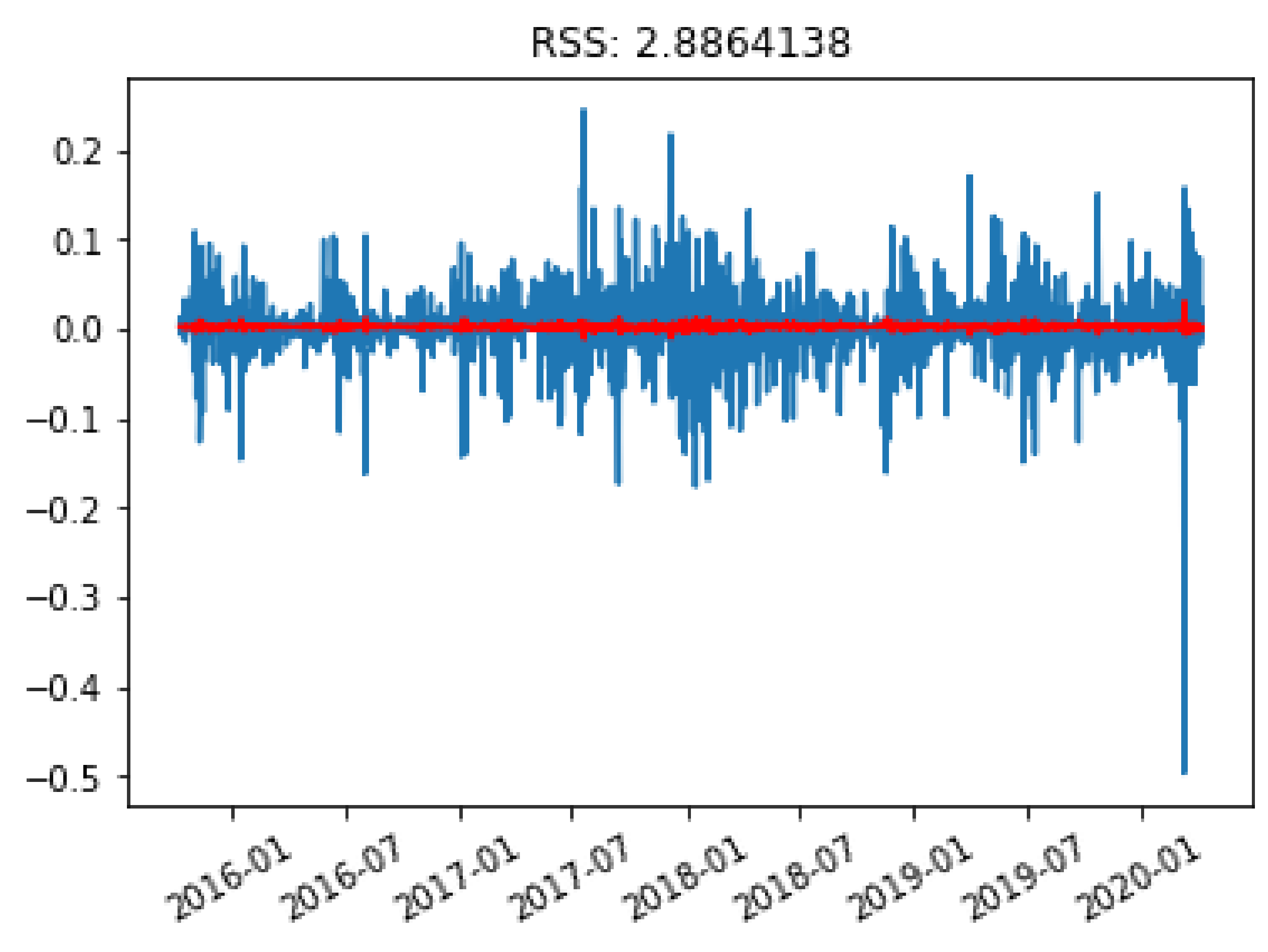
$$RSS = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

RSS = residual sum of squares
y_i = i^th value of the variable to be predicted
$f(x_i)$ = predicted value of y_i
n = upper limit of summation

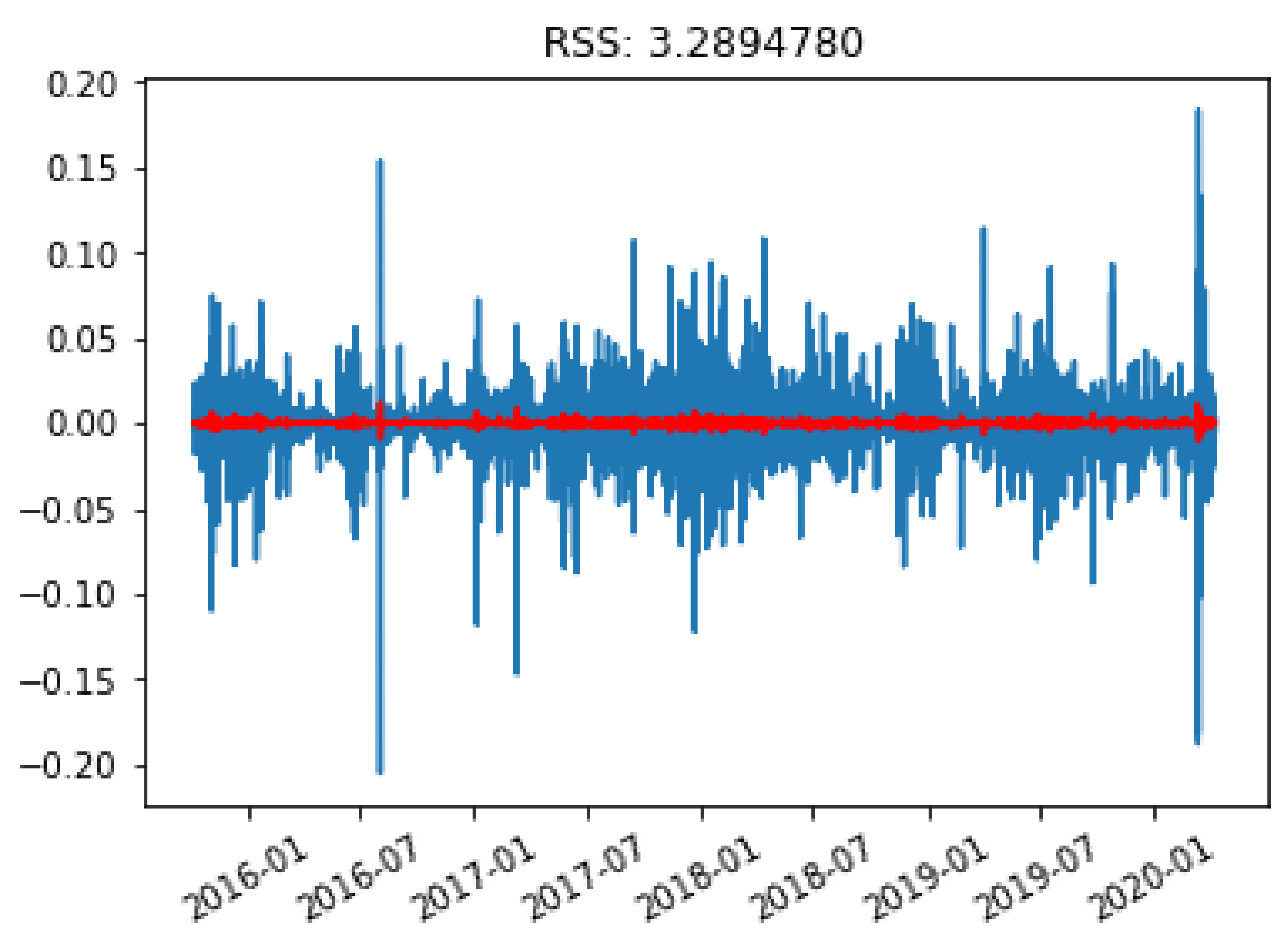
It is a measure of the discrepancy between the data and an estimation model.

Auto Regressive Model

For BTC_Day Dataset

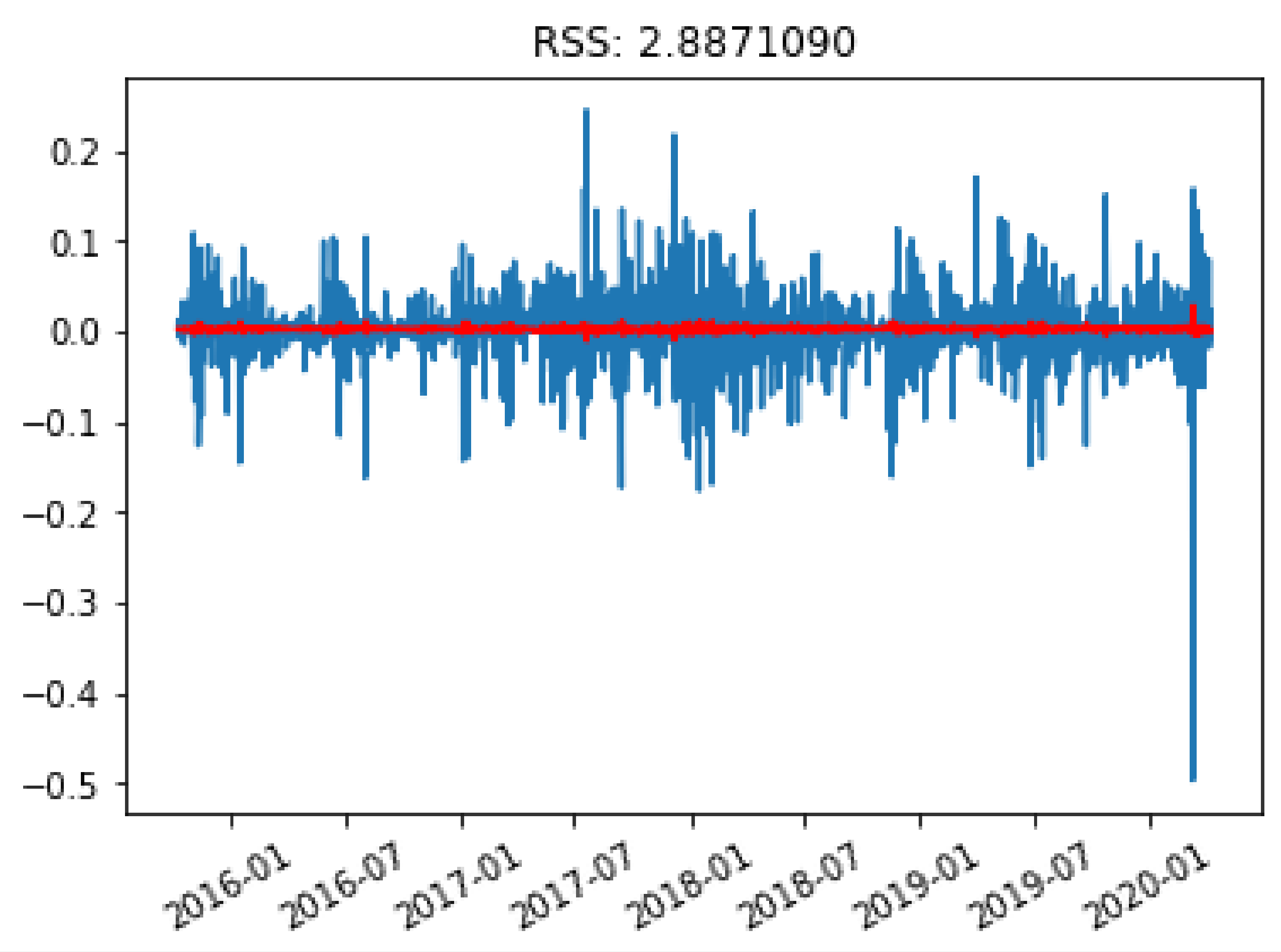


For BTC_Hour Dataset

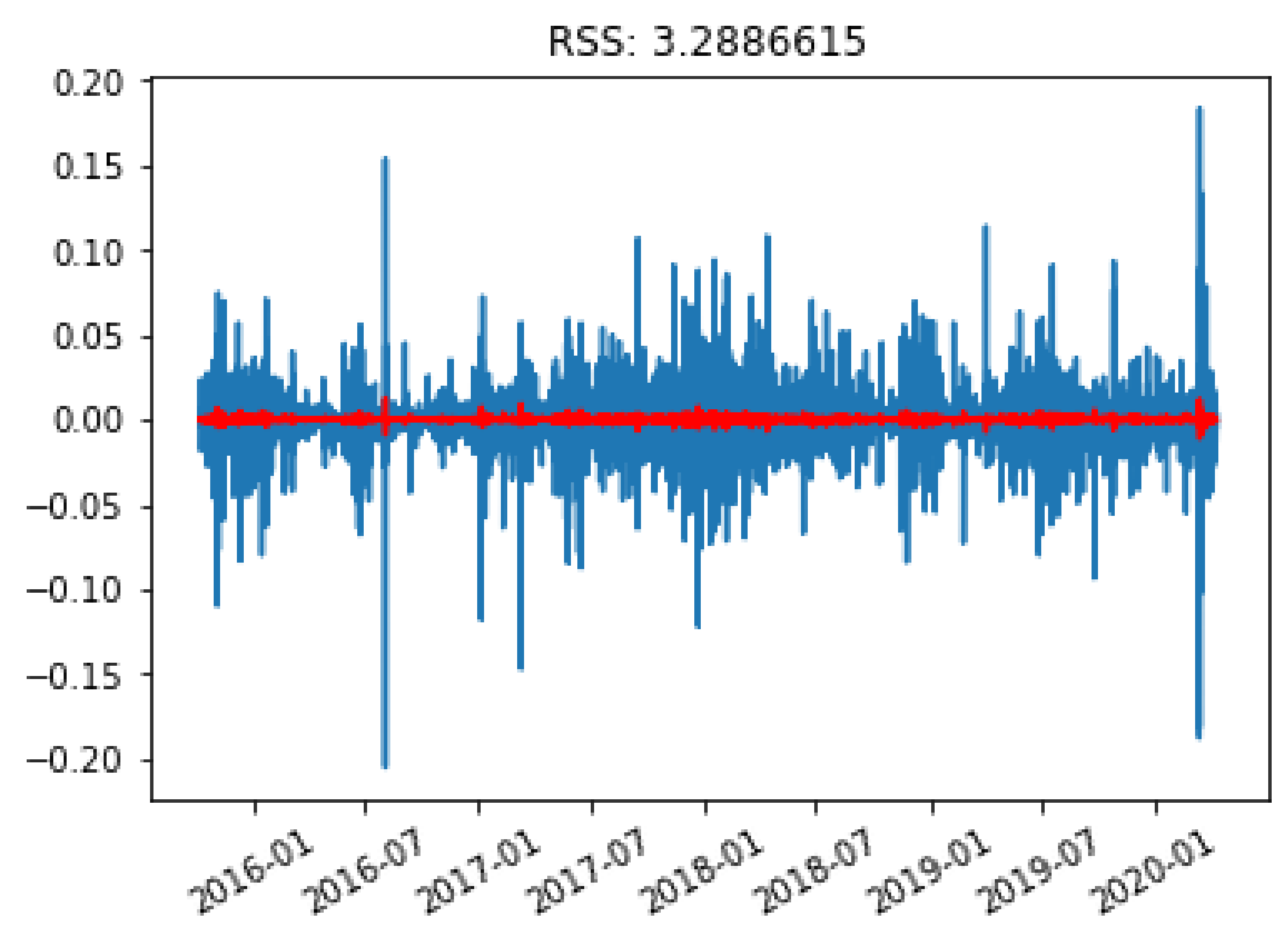


Moving Average Model

For BTC_Day Dataset

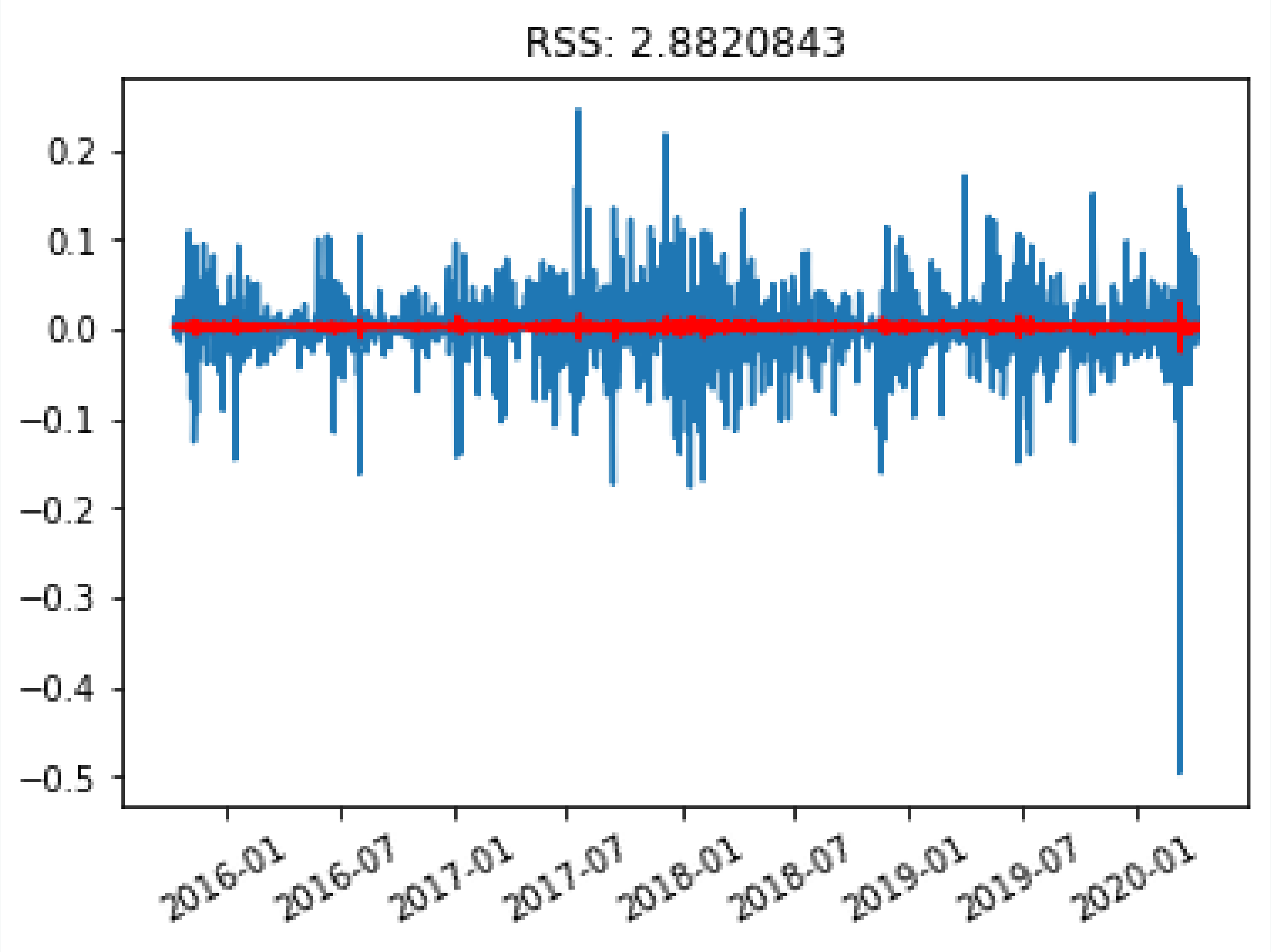


For BTC_Hour Dataset

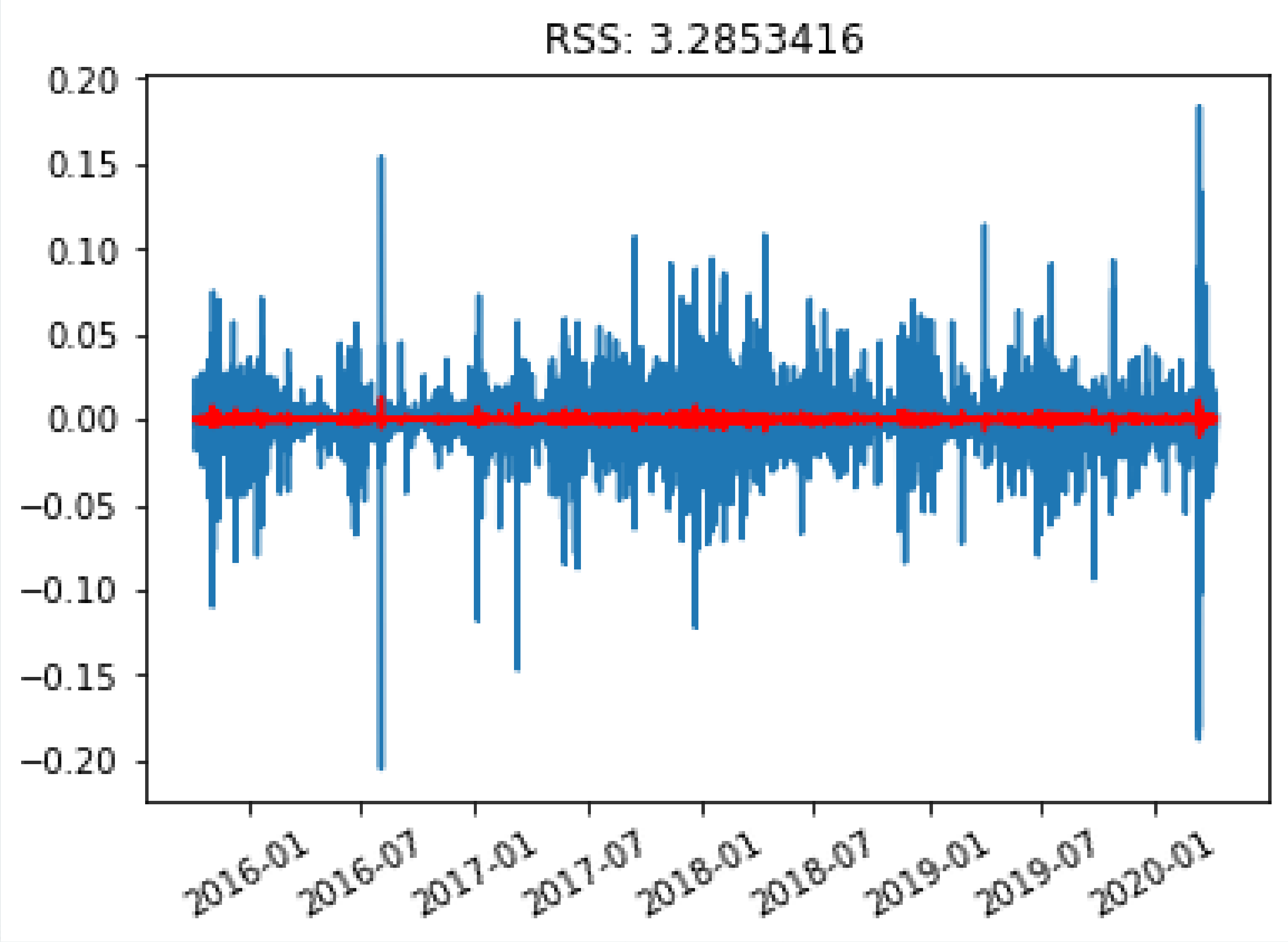


Auto Regressive Integrated Moving Average Model (ARIMA)

For BTC_Day Dataset



For BTC_Hour Dataset



Auto Regressive Integrated Moving Average Model (ARIMA)

For BTC_Day Dataset

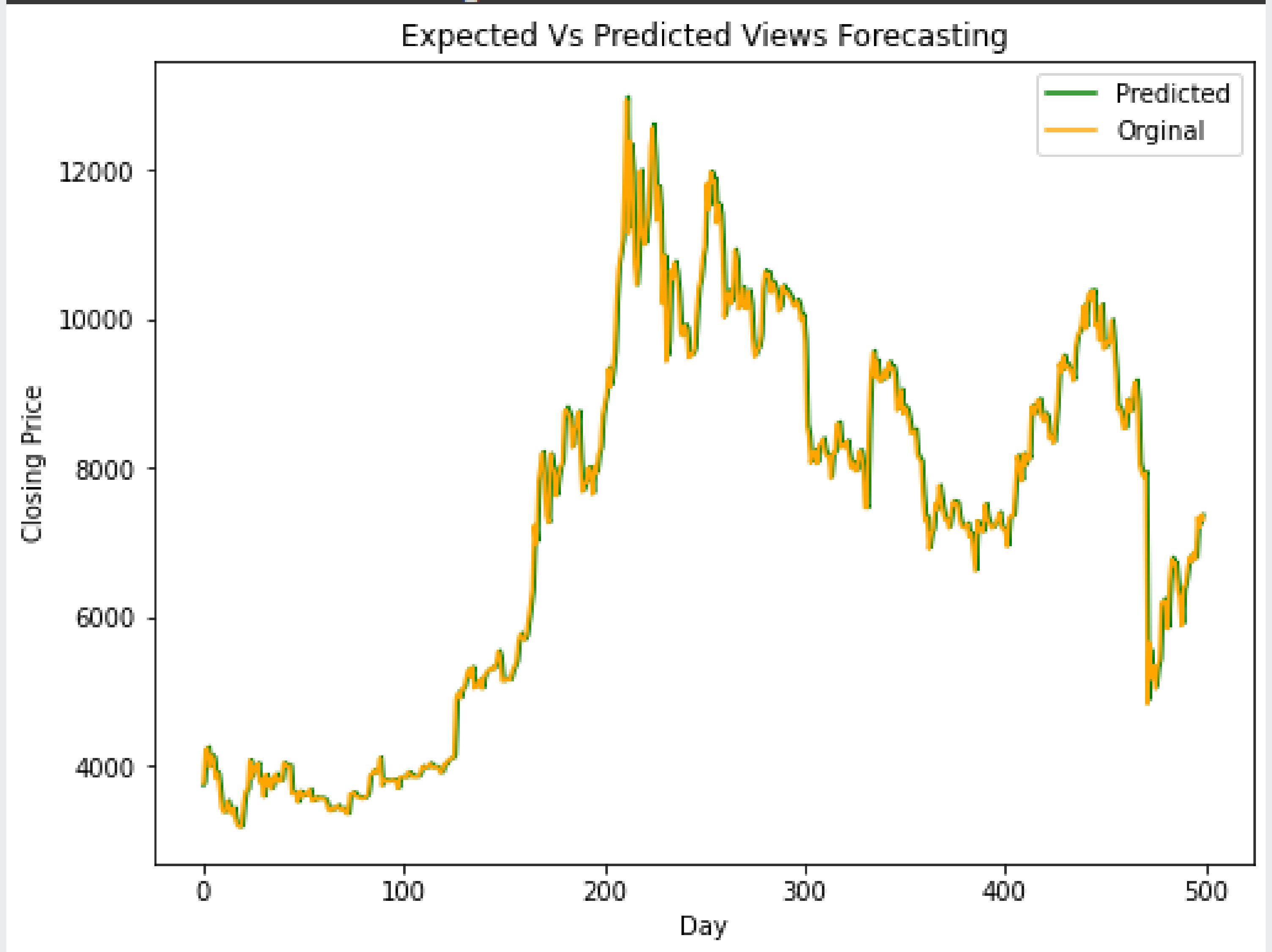
```
size = int(len(ts_log)-500)
# Divide into train and test
train_arima, test_arima = ts_log[0:size], ts_log[size:len(ts_log)]
history = [x for x in train_arima]
predictions = list()
originals = list()
error_list = list()
```

For BTC_Hour Dataset

```
size = int(len(ts_log)-10000)
# Divide into train and test
train_arima, test_arima = ts_log[0:size], ts_log[size:len(ts_log)]
history = [x for x in train_arima]
predictions = list()
originals = list()
error_list = list()
```

For BTC_Day Dataset

```
Mean Error in Predicting Test Case Articles : 2.719307  %
```

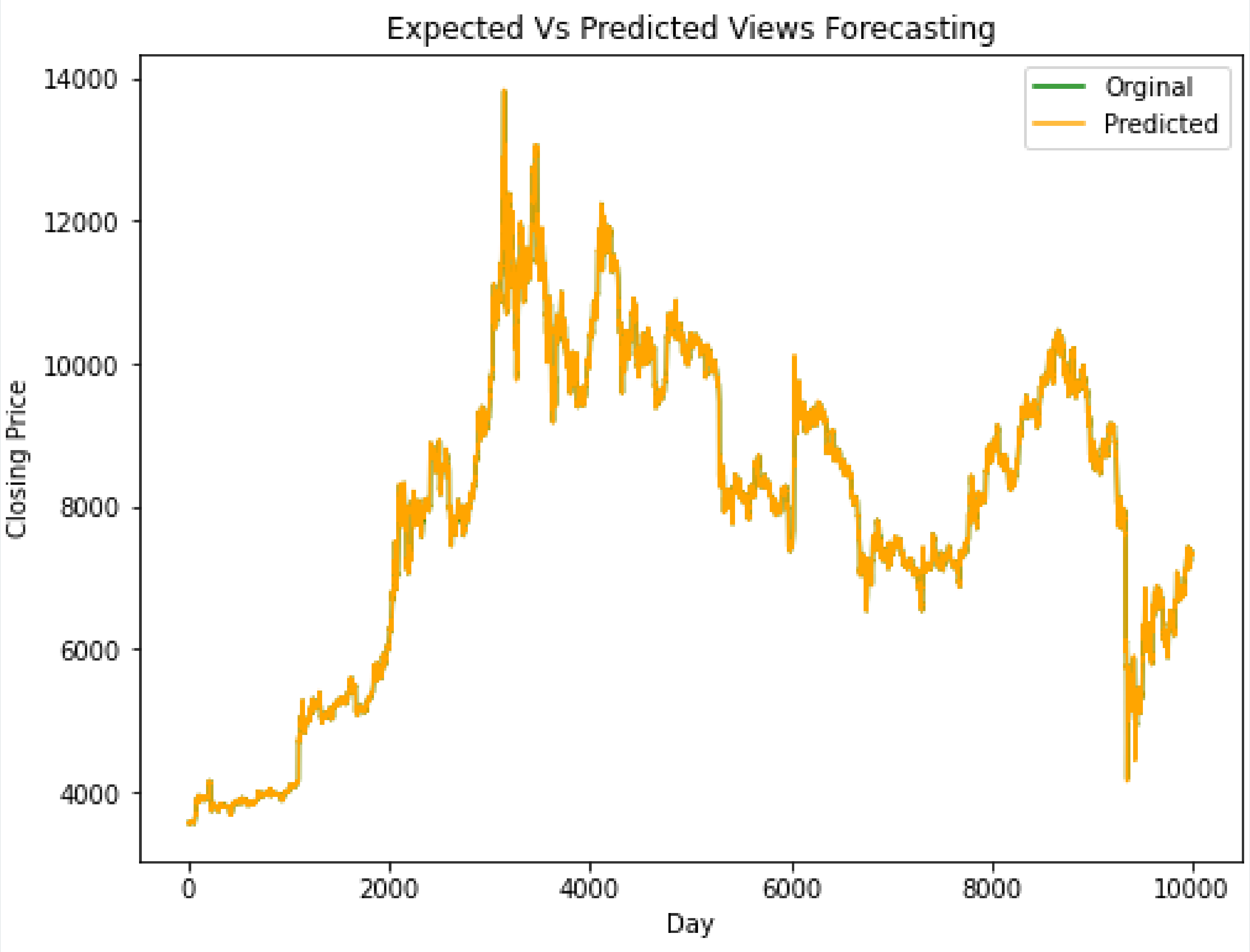



```
Printing Predicted vs Expected Values...


predicted = 3742.756014,   expected = 3771.430000,   error = 0.760295  %
predicted = 3776.510868,   expected = 4223.690000,   error = 10.587404  %
predicted = 4232.496413,   expected = 4240.530000,   error = 0.189448  %
predicted = 4258.103082,   expected = 3984.010000,   error = 6.879829  %
predicted = 3995.042212,   expected = 4141.590000,   error = 3.538443  %
predicted = 4147.941531,   expected = 4101.250000,   error = 1.138471  %
predicted = 4113.368834,   expected = 3835.010000,   error = 7.258360  %
predicted = 3845.287602,   expected = 3903.940000,   error = 1.502390  %
predicted = 3910.253297,   expected = 3691.920000,   error = 5.913814  %
predicted = 3702.796571,   expected = 3436.540000,   error = 7.747809  %
predicted = 3443.288973,   expected = 3377.830000,   error = 1.937900  %
predicted = 3382.760335,   expected = 3394.500000,   error = 0.345844  %
predicted = 3401.424941,   expected = 3529.660000,   error = 3.633071  %
predicted = 3537.450308,   expected = 3411.600000,   error = 3.688894  %
predicted = 3421.400941,   expected = 3356.010000,   error = 1.948473  %
predicted = 3362.516916,   expected = 3431.890000,   error = 2.021425  %
predicted = 3438.742032,   expected = 3265.440000,   error = 5.307157  %
predicted = 3274.324660,   expected = 3195.430000,   error = 2.468984  %
predicted = 3201.008550,   expected = 3184.060000,   error = 0.532294  %
predicted = 3190.336788,   expected = 3193.000000,   error = 0.083408  %
predicted = 3199.860272,   expected = 3502.840000,   error = 8.649545  %
predicted = 3509.592832,   expected = 3667.820000,   error = 4.313929  %
predicted = 3679.668324,   expected = 3683.970000,   error = 0.116767  %
predicted = 3694.321977,   expected = 4077.290000,   error = 9.392710  %
predicted = 4086.475552,   expected = 3839.060000,   error = 6.444691  %
predicted = 3854.363277,   expected = 3979.350000,   error = 3.140883  %
predicted = 3984.390558,   expected = 3948.580000,   error = 0.906922  %
predicted = 3960.235305,   expected = 4036.000000,   error = 1.877222  %
predicted = 4044.563042,   expected = 3777.970000,   error = 7.056516  %
predicted = 3789.793723,   expected = 3808.850000,   error = 0.500316  %
predicted = 3813.531545,   expected = 3588.670000,   error = 6.265874  %
predicted = 3598.910418,   expected = 3889.200000,   error = 7.463992  %
predicted = 3891.075496,   expected = 3729.570000,   error = 4.330405  %
predicted = 3745.771092,   expected = 3826.960000,   error = 2.121499  %
predicted = 3831.442525,   expected = 3692.350000,   error = 3.767046  %
```

For BTC_Hour Dataset

```
Mean Error in Predicting Test Case Articles : 0.462730  %
```




```
Printing Predicted vs Expected Values...


predicted = 3573.242115,   expected = 3570.460000,   error = 0.077920  %
predicted = 3570.745239,   expected = 3579.900000,   error = 0.255727  %
predicted = 3579.372043,   expected = 3573.880000,   error = 0.153672  %
predicted = 3573.910605,   expected = 3580.670000,   error = 0.188775  %
predicted = 3580.454270,   expected = 3585.520000,   error = 0.141283  %
predicted = 3584.931993,   expected = 3556.350000,   error = 0.803689  %
predicted = 3558.090937,   expected = 3556.380000,   error = 0.048109  %
predicted = 3557.513356,   expected = 3560.020000,   error = 0.070411  %
predicted = 3559.777068,   expected = 3564.720000,   error = 0.138663  %
predicted = 3564.265313,   expected = 3566.810000,   error = 0.071343  %
predicted = 3566.487342,   expected = 3565.850000,   error = 0.017873  %
predicted = 3565.832049,   expected = 3566.460000,   error = 0.017607  %
predicted = 3566.457017,   expected = 3565.840000,   error = 0.017304  %
predicted = 3565.857320,   expected = 3566.670000,   error = 0.022785  %
predicted = 3566.639125,   expected = 3567.120000,   error = 0.013481  %
predicted = 3567.057684,   expected = 3582.000000,   error = 0.417150  %
predicted = 3580.992626,   expected = 3586.960000,   error = 0.166363  %
predicted = 3586.047168,   expected = 3582.860000,   error = 0.088956  %
predicted = 3582.938276,   expected = 3586.740000,   error = 0.105994  %
predicted = 3586.642594,   expected = 3582.950000,   error = 0.103060  %
predicted = 3583.049922,   expected = 3583.200000,   error = 0.004188  %
predicted = 3583.331438,   expected = 3586.710000,   error = 0.094197  %
predicted = 3586.467110,   expected = 3586.420000,   error = 0.001314  %
predicted = 3586.302037,   expected = 3585.120000,   error = 0.032971  %
predicted = 3585.217611,   expected = 3589.180000,   error = 0.110398  %
predicted = 3588.961187,   expected = 3586.780000,   error = 0.060812  %
predicted = 3586.780631,   expected = 3595.580000,   error = 0.244727  %
predicted = 3595.089118,   expected = 3596.790000,   error = 0.047289  %
predicted = 3596.365282,   expected = 3597.220000,   error = 0.023761  %
predicted = 3597.144159,   expected = 3595.990000,   error = 0.032096  %
predicted = 3596.054841,   expected = 3591.900000,   error = 0.115673  %
predicted = 3592.219368,   expected = 3592.000000,   error = 0.006107  %
predicted = 3592.153098,   expected = 3594.930000,   error = 0.077245  %
predicted = 3594.731512,   expected = 3599.070000,   error = 0.120545  %
predicted = 3598.680365,   expected = 3594.000000,   error = 0.130227  %
predicted = 3594.174690,   expected = 3594.390000,   error = 0.005990  %
```

Thank You!

-by G21

1) Ankur Saini	2001MM04
2) Bhushan Madushkar	2001MM07
3) Hardik Tiwari	2001MM15
4) Priyanka Kumari	2001MM23
5) Sahil Chaudhari	2001MM29
6) Spandan Prasad	2001MM34