In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
import datetime
import plotly.graph_objects as go
from scipy.stats import linregress
import warnings
```

In [2]:

```
btc_day = pd.read_csv(r"C:\Users\Sahil\Desktop\IITP\VScode\Bitcoin\BTCUSD_day.csv")
btc_day.head()
```

Out[2]:

	Date	Symbol	Open	High	Low	Close	Volume BTC	Volume USD
0	2020-04-10	BTCUSD	7315.25	7315.25	7315.25	7315.25	0.00	0.00
1	2020-04-09	BTCUSD	7369.60	7378.85	7115.04	7315.25	2237.13	16310014.53
2	2020-04-08	BTCUSD	7201.81	7432.23	7152.80	7369.60	2483.60	18138080.27
3	2020-04-07	BTCUSD	7336.96	7468.42	7078.00	7201.81	2333.34	17047120.32
4	2020-04-06	BTCUSD	6775.21	7369.76	6771.01	7336.96	3727.47	26533750.17

In [3]:

```
btc_min = pd.read_csv(r"C:\Users\Sahil\Desktop\IITP\VScode\Bitcoin\BTCUSD_1min.csv",
index_col='Date', parse_dates=True)
btc_min.head()
```

Out[3]:

	Unix Timestamp	Symbol	Open	High	Low	Close	Volume
Date							
2020-04-09 00:00:00	1586390400000	BTCUSD	7369.60	7369.60	7348.47	7350.41	7.733528
2020-04-08 23:59:00	1586390340000	BTCUSD	7364.37	7369.60	7364.37	7369.60	1.310954
2020-04-08 23:58:00	1586390280000	BTCUSD	7360.89	7367.18	7360.89	7364.37	1.018774
2020-04-08 23:57:00	1586390220000	BTCUSD	7357.62	7366.12	7357.62	7360.89	0.026251
2020-04-08 23:56:00	1586390160000	BTCUSD	7370.71	7370.71	7357.62	7357.62	0.444867

In [4]:

```
btc_hr = pd.read_csv(r"C:\Users\Sahil\Desktop\IITP\VScode\Bitcoin\BTCUSD_1hr.csv")
btc_hr.head()
```

Out[4]:

	Unix Timestamp	Date	Symbol	Open	High	Low	Close	Volume
0	1586390400000	2020-04-09 00:00:00	BTCUSD	7369.60	7369.60	7338.23	7338.23	7.788915
1	1586386800000	2020-04-08 23:00:00	BTCUSD	7367.28	7398.92	7348.47	7369.60	66.558293
2	1586383200000	2020-04-08 22:00:00	BTCUSD	7337.19	7390.25	7322.87	7367.28	88.486108
3	1586379600000	2020-04-08 21:00:00	BTCUSD	7327.70	7370.00	7311.33	7337.19	35.551553
4	1586376000000	2020-04-08 20:00:00	BTCUSD	7320.11	7375.72	7313.66	7327.70	153.655811

In [5]:

```
print('Shape of btc_hr dataset is: ',btc_hr.shape)
print('Shape of btc_min dataset is: ',btc_min.shape)
print('Shape of btc_day dataset is: ',btc_day.shape)
```

```
Shape of btc_hr dataset is:  (39465, 8)
Shape of btc_min dataset is:  (2283519, 7)
Shape of btc_day dataset is:  (1647, 8)
```

In [6]:

```
btc_day.info()
```

```
<class 'pandas.core.frame.DataFrame'>
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)
memory usage: 103.1+ KB
```

In [ ]:

```
btc_day['Date'] = pd.to_datetime(btc_day['Date'])
```

In [4]:

```
btc_day.isnull().sum()
```

Out[4]:

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

In [5]:

```
btc_day.describe()
```

Out[5]:

	Open	High	Low	Close	Volume BTC	Volume USD
count	1647.000000	1647.000000	1647.000000	1647.000000	1647.000000	1.647000e+03
mean	4823.493224	4971.083673	4656.335337	4827.596515	3789.357365	2.129125e+07
std	3937.740533	4078.527350	3760.335160	3936.503969	4436.695998	3.694968e+07
min	242.500000	245.000000	236.000000	243.950000	0.000000	0.000000e+00
25%	741.975000	751.830000	732.030000	742.260000	1028.940000	1.659287e+06
50%	4147.100000	4295.090000	4000.000000	4154.840000	2263.370000	9.395394e+06
75%	7920.510000	8139.750000	7630.445000	7920.510000	4918.785000	2.464551e+07
max	19499.990000	19999.000000	18870.000000	19499.990000	49229.150000	5.606539e+08

In [10]:

```
print(min(btc_day['Date']),"||||",max(btc_day['Date']))
```

```
2015-10-08 00:00:00 |||| 2020-04-10 00:00:00
```

In [6]:

```
btc_day_copy = btc_day.copy()
btc_day_copy.drop(btc_day_copy[btc_day_copy['Volume BTC'] == 0].index, inplace =
True)
btc_day_copy.head()
```

Out[6]:

	Date	Symbol	Open	High	Low	Close	Volume BTC	Volume USD
1	2020-04-09	BTCUSD	7369.60	7378.85	7115.04	7315.25	2237.13	16310014.53
2	2020-04-08	BTCUSD	7201.81	7432.23	7152.80	7369.60	2483.60	18138080.27
3	2020-04-07	BTCUSD	7336.96	7468.42	7078.00	7201.81	2333.34	17047120.32
4	2020-04-06	BTCUSD	6775.21	7369.76	6771.01	7336.96	3727.47	26533750.17
5	2020-04-05	BTCUSD	6870.20	6907.90	6678.60	6775.21	1275.49	8662210.80

In [7]:

```
btc_day_copy.shape
```

Out[7]:

```
(1643, 8)
```

In [43]:

```
#creating return series
btc_day_copy['Return'] = btc_day_copy['Close'].pct_change()*100
```

In [44]:

```
btc_day_copy.head()
```

Out[44]:

	Date	Symbol	Open	High	Low	Close	Volume BTC	Volume USD	Return
1	2020-04-09	BTCUSD	7369.60	7378.85	7115.04	7315.25	2237.13	16310014.53	NaN
2	2020-04-08	BTCUSD	7201.81	7432.23	7152.80	7369.60	2483.60	18138080.27	0.742968
3	2020-04-07	BTCUSD	7336.96	7468.42	7078.00	7201.81	2333.34	17047120.32	-2.276786
4	2020-04-06	BTCUSD	6775.21	7369.76	6771.01	7336.96	3727.47	26533750.17	1.876612
5	2020-04-05	BTCUSD	6870.20	6907.90	6678.60	6775.21	1275.49	8662210.80	-7.656441

In [42]:

```
np.mean(btc_day_copy['Return'])
```

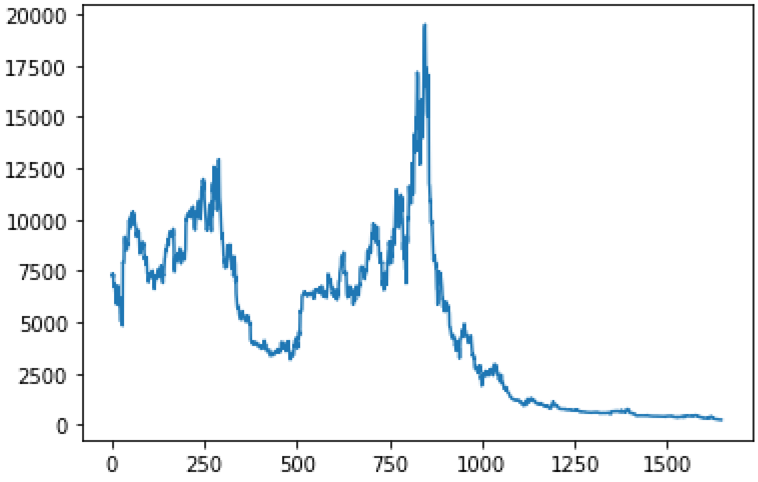
Out[42]:

```
-0.11739233132494797
```

Visualization

In [10]:

```
#plotting closing prices
ax1 = btc_day_copy['Close'].plot(style = ['-'])
```

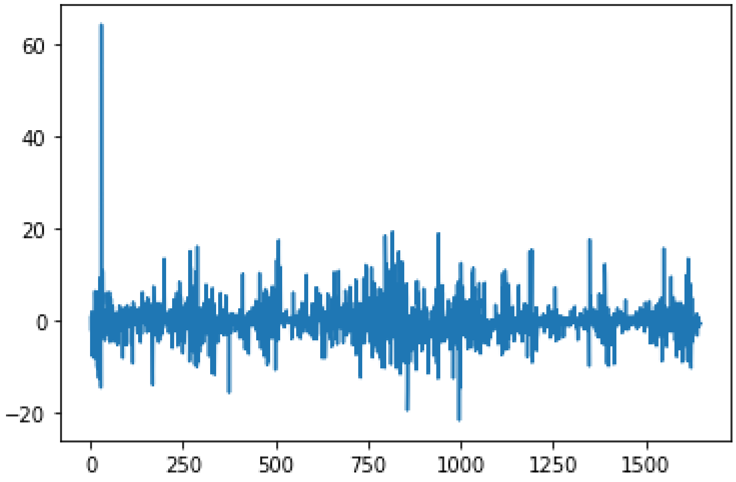


In [ ]:

```

```

In [11]:

```
btc_day_copy.set_index('Date')
ax2 = btc_day_copy['Return'].plot(style = ['-'])
```

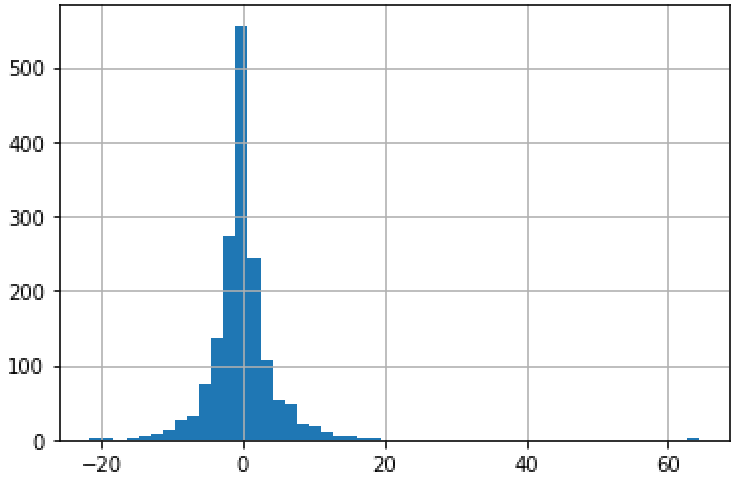


In [11]:

```
btc_day_copy['Return'].hist(bins=50)
```

Out[11]:

```
<AxesSubplot:>
```

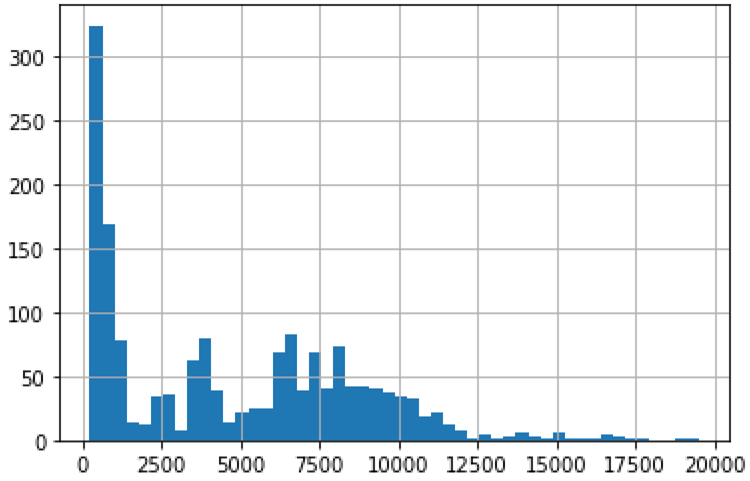


In [12]:

```
btc_day_copy['Close'].hist(bins=50)
```

Out[12]:

```
<AxesSubplot:>
```

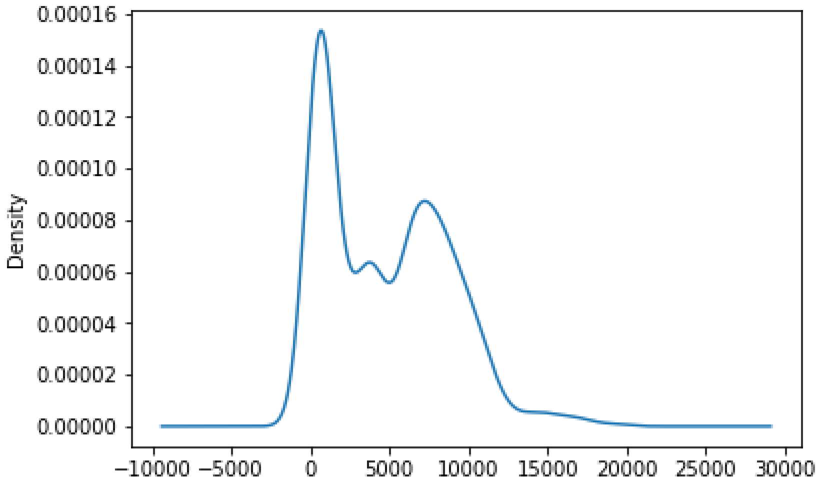


In [13]:

```
#kernel density
btc_day_copy['Close'].plot(kind='kde')
```

Out[13]:

```
<AxesSubplot:ylabel='Density'>
```

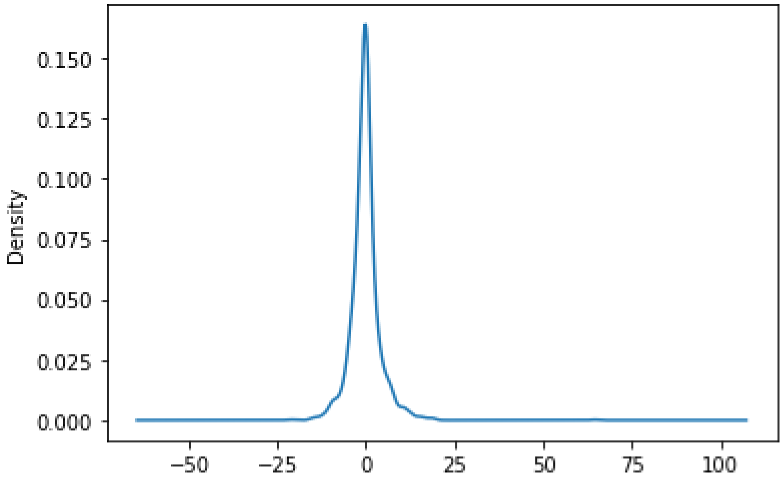


In [14]:

```
btc_day_copy['Return'].plot(kind='kde')
```

Out[14]:

```
<AxesSubplot:ylabel='Density'>
```



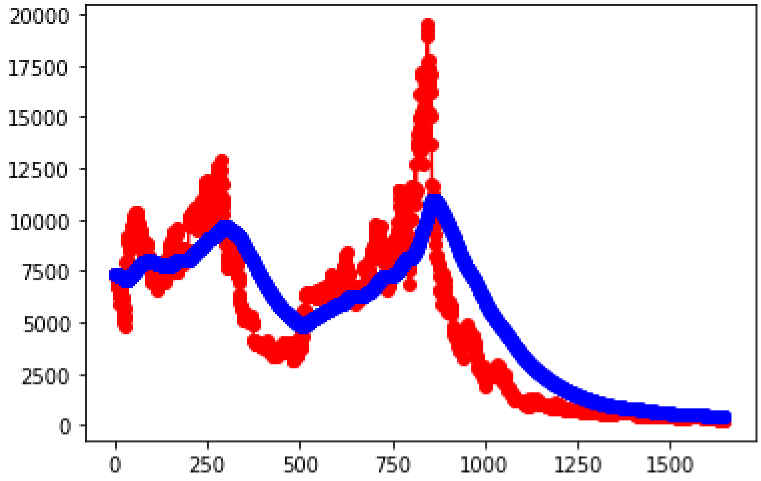
In [16]:

```
from statsmodels.tsa.holtwinters import SimpleExpSmoothing
model = SimpleExpSmoothing(btc_day_copy['Close']).fit(smoothing_level=.01,
optimized=False)
btc_day_copy['Close'].plot(marker = 'o', color='red')
model.fittedvalues.plot(marker = 'o', color = 'blue')
```

```
C:\Users\Sahil\AppData\Roaming\Python\Python310\site-packages\statsmodels\tsa\base\tsa_model.py:
471: ValueWarning: An unsupported index was provided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

Out[16]:

```
<AxesSubplot:>
```

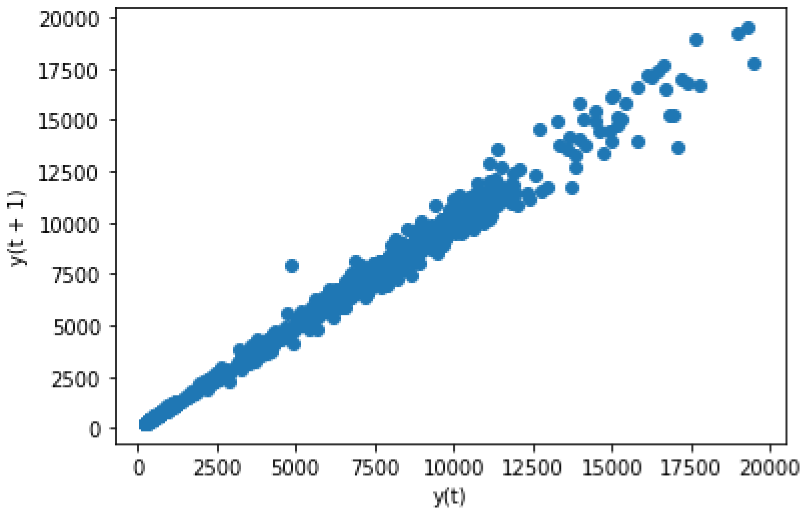


In [17]:

```
from pandas.plotting import lag_plot
lag_plot(btc_day_copy['Close'])
```

Out[17]:

```
<AxesSubplot:xlabel='y(t)', ylabel='y(t + 1)'>
```

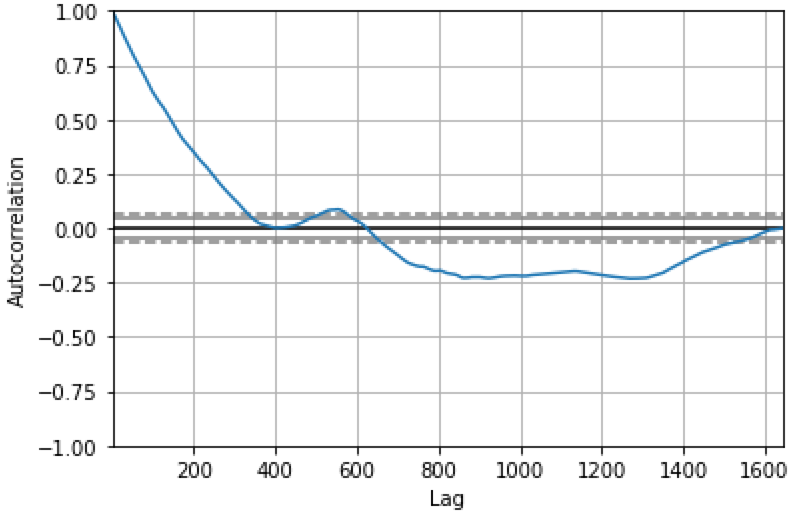


In [18]:

```
#Autocorrelation plot
from pandas.plotting import autocorrelation_plot

autocorrelation_plot(btc_day_copy['Close'])
```

Out[18]:

```
<AxesSubplot:xlabel='Lag', ylabel='Autocorrelation'>
```



In [20]:

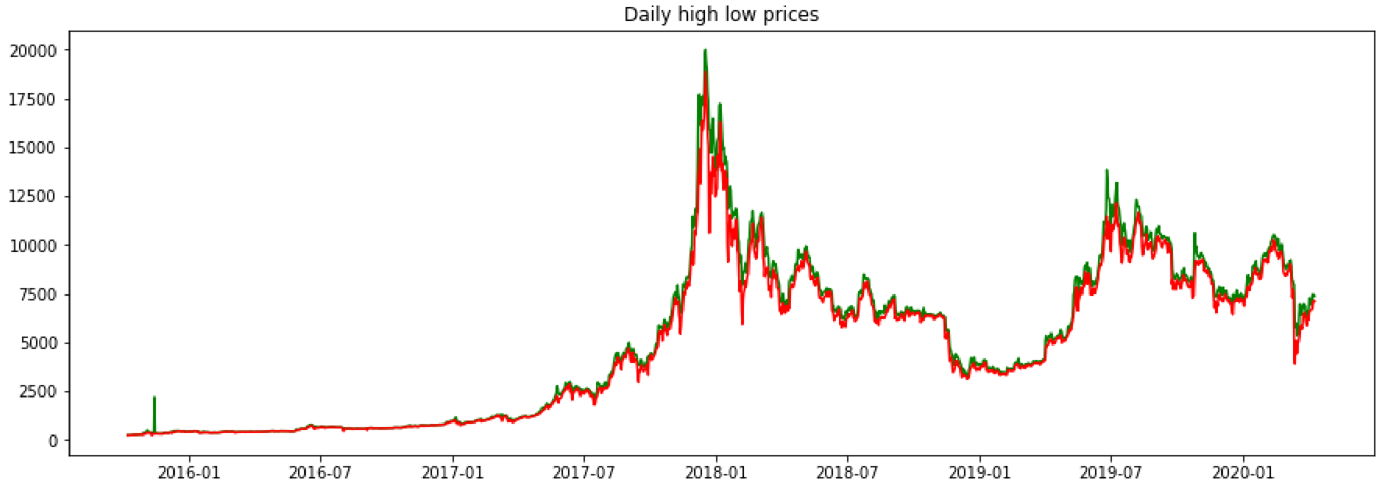
```
btc_day_date_index = btc_day_copy.set_index('Date')
```

In [31]:

```
candlestick = go.Candlestick(
   x = btc_day_date_index.index,
   open = btc_day_date_index['Open'],
   high = btc_day_date_index['High'],
   low = btc_day_date_index['Low'],
   close = btc_day_date_index['Close'])

fig = go.Figure(data = [candlestick])
fig.show()
```

In [30]:

```
plt.figure (figsize=(15,5))
plt.plot(btc_day_date_index['High'], color='green')
# Plot the daily low price
plt.plot(btc_day_date_index['Low'], color='red')
plt.title('Daily high low prices')
plt.show()
```



In [32]:

```
# Resample from daily to weekly
```

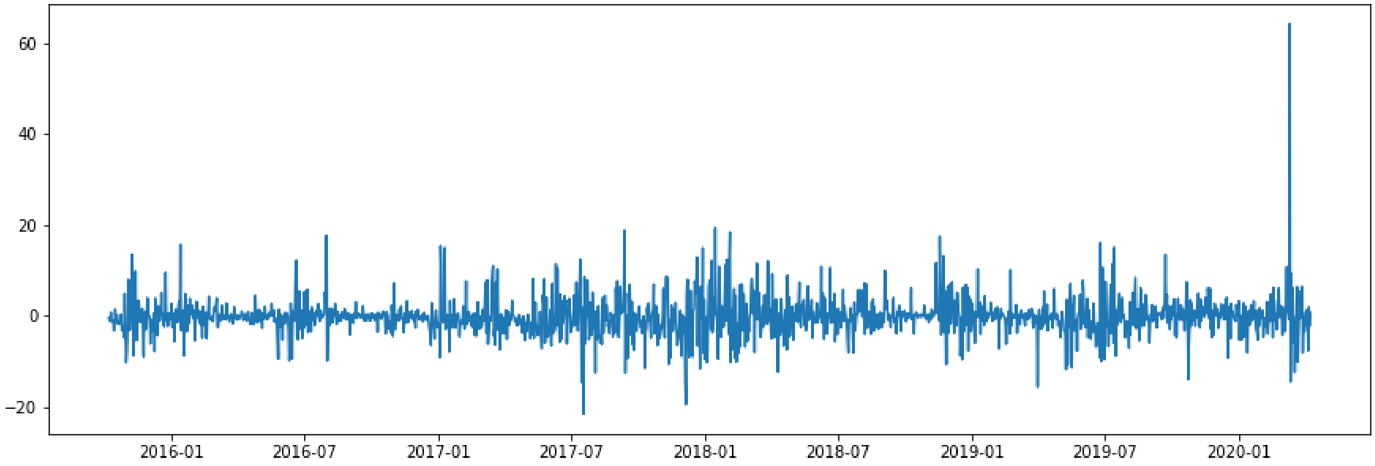
```
btc_weekly = btc_day_date_index.resample('W').mean()
btc_weekly.head()
```

Out[32]:

Date	Open	High	Low	Close	Volume BTC	Volume USD	Return	Typical Price
2015-10-11	244.535000	247.692500	244.250000	246.285000	18.545000	4563.257500	-0.367872	246.075833
2015-10-18	256.031429	263.710000	254.200000	258.648571	175.141429	46461.315714	-1.347957	258.852857
2015-10-25	275.254286	281.348571	273.404286	278.242857	337.531429	93461.344286	-0.810194	277.665238
2015-11-01	306.202857	323.670000	293.718571	312.005714	939.355714	293904.180000	-3.321859	309.798095
2015-11-08	378.771429	417.978571	359.361429	385.485714	1708.240000	670326.267143	-0.407066	387.608571

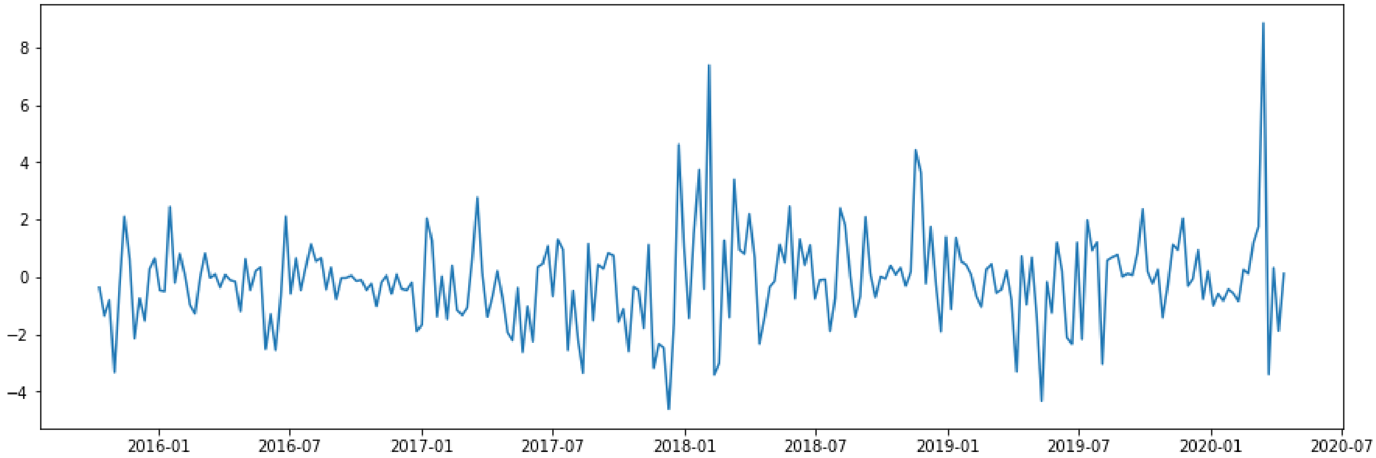
In [37]:

```
#DAILY RETURNS

plt.figure (figsize=(15,5))
plt.plot(btc_day_date_index['Return'])
plt.show()
```



In [34]:

```
#WEEKLY RETURN

plt.figure (figsize=(15,5))
plt.plot(btc_weekly['Return'])
plt.show()
```



DATA TRANSFORMATION

SMA- Simple Moving Average

It is simply the arithmetic mean of the price over a specified period. The average is called "moving" because it is always calculated using the most recent n periods, and therefore moves along with the price on the chart.

In [35]:

```
fig1 = go.Figure(
        data = [
            go.Candlestick(
                x = btc_day_date_index.index,
                open = btc_day_date_index.Open,
                high = btc_day_date_index.High,
                low = btc_day_date_index.Low,
                close = btc_day_date_index.Close
            ),
            go.Scatter(
                x = btc_day_date_index.index,
                y = btc_day_date_index.Close.rolling(window=20).mean(),
                mode = 'lines',
                name = '20SMA',
                line = {'color': '#ff006a'}
            ),
            go.Scatter(
                x = btc_day_date_index.index,
                y = btc_day_date_index.Close.rolling(window=50).mean(),
                mode = 'lines',
                name = '50SMA',
                line = {'color': '#1900ff'}
            )
        ]
    )

fig1.update_layout(
    title = 'The Candlestick graph for BITCOIN',
    xaxis_title = 'Date',
    yaxis_title = 'Price CURRENCY',
    xaxis_rangeslider_visible = False
)
#fig1.update_yaxes(tickprefix='£')

fig1.show()
```

https://itnext.io/create-beautiful-cryptocurrency-graphs-in-python-bec7b9cbc21a

In [ ]:

```
fig2 = go.Figure(
        data = [
            go.Scatter(
                x = btc_day_date_index.index,
                y = btc_day_date_index.Close,
                mode = 'lines',
                name = 'Close',
                line = {'color': '#006400'}
            ),
            go.Scatter(
                x = btc_day_date_index.index,
                y = btc_day_date_index.Close.rolling(window=20).mean(),
                mode = 'lines',
                name = '20SMA',
                line = {'color': '#ff006a'}
            ),
            go.Scatter(
                x = btc_day_date_index.index,
                y = btc_day_date_index.Close.rolling(window=50).mean(),
                mode = 'lines',
                name = '50SMA',
                line = {'color': '#1900ff'}
            )
        ]
    )

fig2.update_layout(
    #title = 'The Candlestick graph for BITCOIN',
    xaxis_title = 'Date',
    yaxis_title = 'Close',
    xaxis_rangeslider_visible = False
)
#fig1.update_yaxes(tickprefix='£')

fig2.show()
```