

In [11]: `!pip install mplfinance`

```
Collecting mplfinance
  Downloading mplfinance-0.12.9b7-py3-none-any.whl (73 kB)
Requirement already satisfied: pandas in c:\users\dell\anaconda3\lib\site-packages (from mplfinance) (1.4.2)
Requirement already satisfied: matplotlib in c:\users\dell\anaconda3\lib\site-packages (from mplfinance) (3.5.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib->mplfinance) (2.8.2)
Requirement already satisfied: packaging>=20.0 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib->mplfinance) (21.3)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib->mplfinance) (1.3.2)
Requirement already satisfied: numpy>=1.17 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib->mplfinance) (1.21.5)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib->mplfinance) (3.0.4)
Requirement already satisfied: cyclor>=0.10 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib->mplfinance) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib->mplfinance) (4.25.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib->mplfinance) (9.0.1)
Requirement already satisfied: six>=1.5 in c:\users\dell\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib->mplfinance) (1.16.0)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell\anaconda3\lib\site-packages (from pandas->mplfinance) (2021.3)
Installing collected packages: mplfinance
Successfully installed mplfinance-0.12.9b7
```

In [12]: `import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from mplfinance.original_flavor import candlestick_ohlc
import matplotlib.dates as mpl_dates`

In [14]: `df = pd.read_csv("ETH-USD.csv", parse_dates=True)`

In [15]: `df = df.drop(columns=['Adj Close'])
df.head()`

Out[15]:

	Date	Open	High	Low	Close	Volume
0	2015-08-07	2.831620	3.536610	2.521120	2.772120	164329.0
1	2015-08-08	2.793760	2.798810	0.714725	0.753325	674188.0
2	2015-08-09	0.706136	0.879810	0.629191	0.701897	532170.0
3	2015-08-10	0.713989	0.729854	0.636546	0.708448	405283.0
4	2015-08-11	0.708087	1.131410	0.663235	1.067860	1463100.0

```
In [16]: df['100ma'] = df['Close'].rolling(window = 100, min_periods = 0).mean()
df
```

```
Out[16]:
```

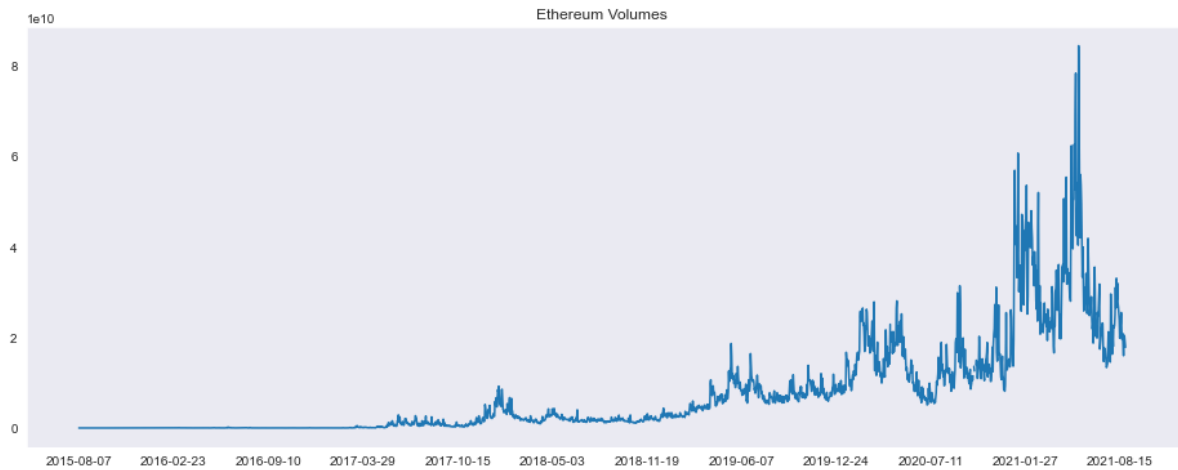
	Date	Open	High	Low	Close	Volume	100ma
0	2015-08-07	2.831620	3.536610	2.521120	2.772120	1.643290e+05	2.772120
1	2015-08-08	2.793760	2.798810	0.714725	0.753325	6.741880e+05	1.762723
2	2015-08-09	0.706136	0.879810	0.629191	0.701897	5.321700e+05	1.409114
3	2015-08-10	0.713989	0.729854	0.636546	0.708448	4.052830e+05	1.233947
4	2015-08-11	0.708087	1.131410	0.663235	1.067860	1.463100e+06	1.200730
...
2207	2021-08-22	3226.227295	3272.733154	3142.007080	3242.115479	1.598328e+10	2497.191824
2208	2021-08-23	3241.357422	3373.384277	3235.851318	3319.257324	2.051111e+10	2494.003176
2209	2021-08-24	3324.855469	3358.688232	3154.121338	3172.456299	2.013103e+10	2489.852678
2210	2021-08-25	3174.269775	3248.727295	3086.114990	3224.915283	1.890273e+10	2489.277854
2211	2021-08-26	3229.452148	3247.775391	3062.338867	3101.602051	1.783986e+10	2486.493174

2212 rows × 7 columns

```
In [17]: fig, ax = plt.subplots(figsize=(16,6))
ax.plot(df.Date, df.Close)
ax.plot(df.Date, df['100ma'])
ax.xaxis.set_major_locator(plt.MaxNLocator(15)) # reduce number of x-labels
plt.title('Ethereum Prices')
plt.grid()
plt.show()
```

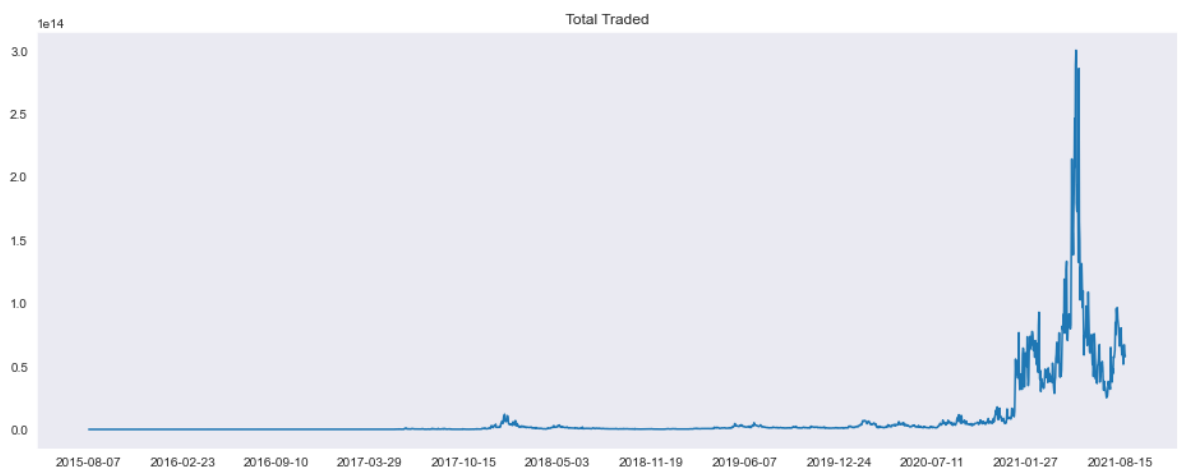


```
In [18]: fig, ax = plt.subplots(figsize=(16,6))
ax.plot(df.Date, df.Volume)
ax.xaxis.set_major_locator(plt.MaxNLocator(15)) # reduce number of x-labels
plt.title('Ethereum Volumes')
plt.grid()
plt.show()
```



```
In [19]: # Market Cap

df['Total Traded'] = df['Open']*df['Volume']
fig, ax = plt.subplots(figsize=(16,6))
ax.plot(df.Date, df['Total Traded'])
ax.xaxis.set_major_locator(plt.MaxNLocator(15)) # reduce number of x-labels
plt.title('Total Traded')
plt.grid()
plt.show()
```



```
In [20]: df.iloc[df['Total Traded'].argmax()]
```

```
Out[20]: Date                2021-05-13
Open                3828.918457
High                4032.563477
Low                 3549.407227
Close               3715.148438
Volume              78398214539.0
100ma               2140.178688
Total Traded       300180370644222.875
Name: 2106, dtype: object
```

```
In [21]: ohlc = df[(df['Date'] > '2021-04-01') & (df['Date'] <= '2021-07-26')]
ohlc = ohlc.loc[:, ['Date', 'Open', 'High', 'Low', 'Close']]
ohlc['Date'] = pd.to_datetime(ohlc['Date'])
ohlc['Date'] = ohlc['Date'].apply(mpl_dates.date2num)
ohlc = ohlc.astype(float)
fig, ax = plt.subplots(figsize = (16,6))
candlestick_ohlc(ax, ohlc.values, width=0.6, colorup='green', colordown='red',
ax.set_xlabel('Date')
ax.set_ylabel('Price')
date_format = mpl_dates.DateFormatter('%d-%m-%Y')
ax.xaxis.set_major_formatter(date_format)
fig.autofmt_xdate()

fig.tight_layout()

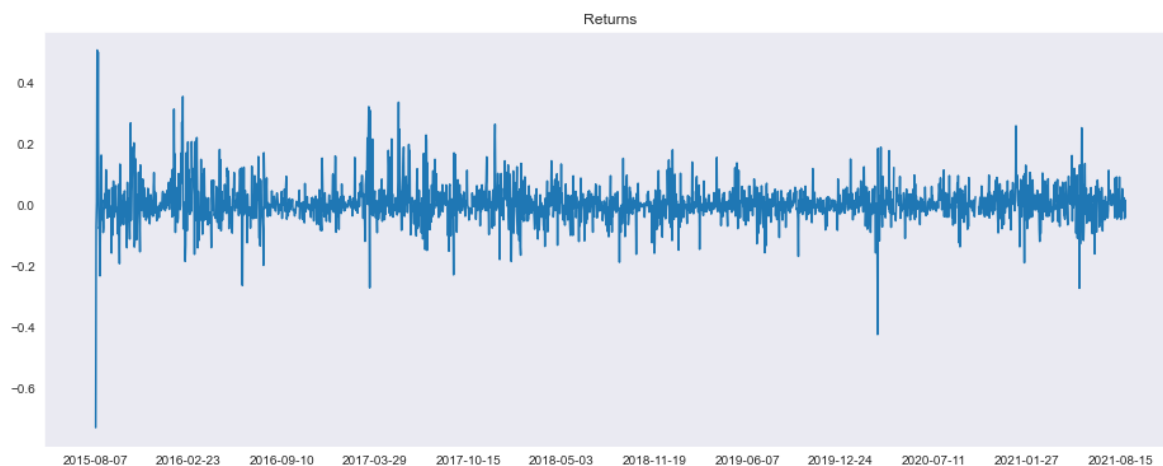
plt.show()
```



In [22]: *# Daily percentage Change*

```
df["returns"] = (df["Close"]/df["Close"].shift(1)) - 1

fig, ax = plt.subplots(figsize=(16,6))
ax.plot(df.Date, df['returns'])
ax.xaxis.set_major_locator(plt.MaxNLocator(15)) # reduce number of x-labels
plt.title('Returns')
plt.grid()
plt.show()
```



In []: