NETFLIX STOCK ANALYSIS

Import Liberaries: Pandas, Numpy, Matplotlib, Seaborn, Plotly

In [71]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from datetime import datetime
```

Import Netflix Stocks Dataset

In [72]:
```
df = pd.read_csv("NFLXdaily.csv")
```

In [73]:
```
df.head(5)
```

Out[73]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2019-01-02	259.279999	269.750000	256.579987	267.660004	267.660004	11679500
1	2019-01-03	270.200012	275.790009	264.429993	271.200012	271.200012	14969600
2	2019-01-04	281.880005	297.799988	278.540009	297.570007	297.570007	19330100
3	2019-01-07	302.100006	316.799988	301.649994	315.339996	315.339996	18620100
4	2019-01-08	319.980011	320.589996	308.010010	320.269989	320.269989	15359200

Set figure in seaborn liberary and Index the date coloumn

In [74]:
```
sns.set(rc={'figure.figsize':(10,5)})
```

In [75]:
```
df['Date'] = pd.to_datetime(df['Date'])
df = df.set_index('Date')
df.head()
```

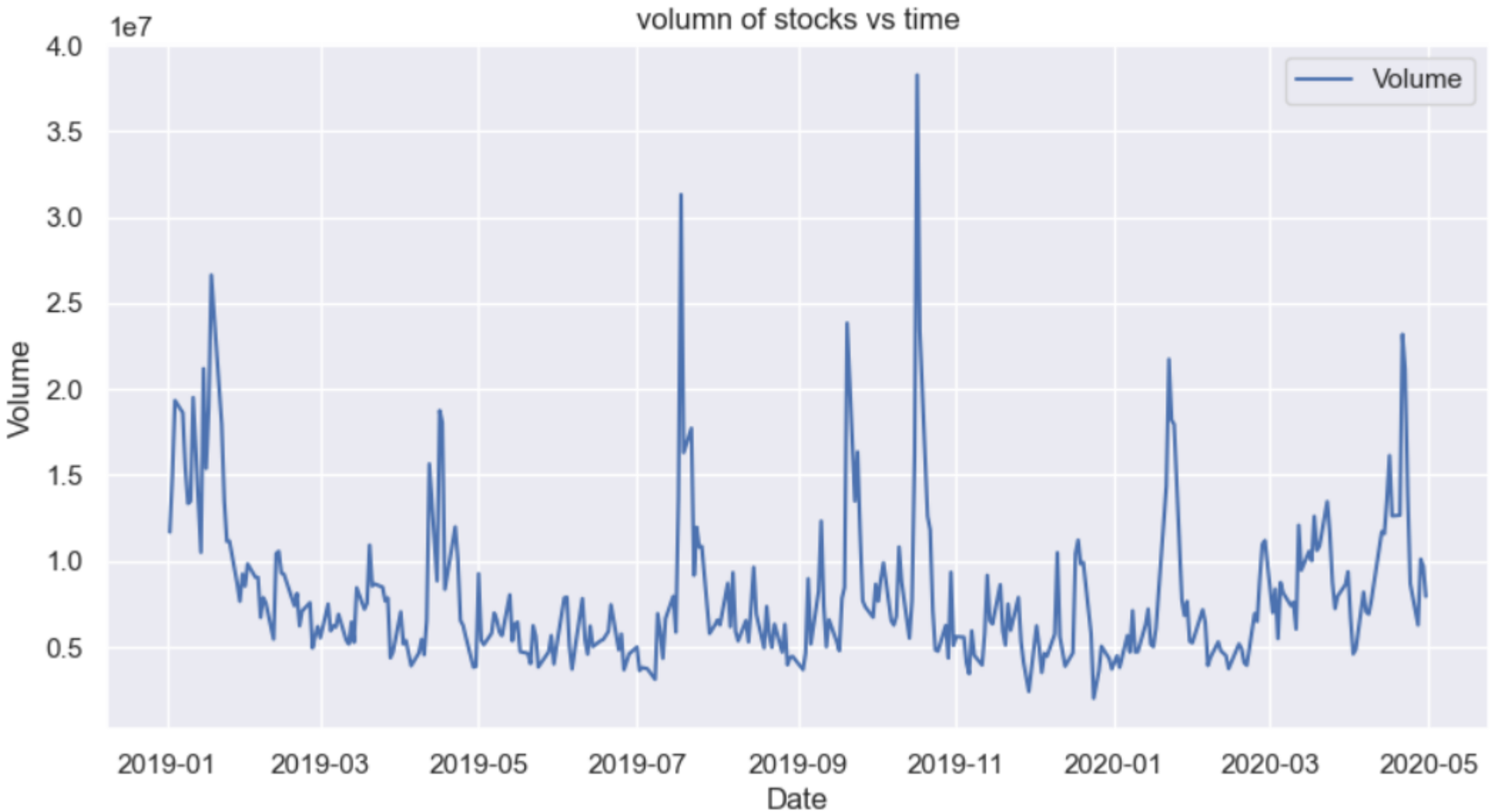
Out[75]:

Date	Open	High	Low	Close	Adj Close	Volume
2019-01-02	259.279999	269.750000	256.579987	267.660004	267.660004	11679500
2019-01-03	270.200012	275.790009	264.429993	271.200012	271.200012	14969600
2019-01-04	281.880005	297.799988	278.540009	297.570007	297.570007	19330100
2019-01-07	302.100006	316.799988	301.649994	315.339996	315.339996	18620100
2019-01-08	319.980011	320.589996	308.010010	320.269989	320.269989	15359200

Volumn of Stock Trade Analysis by Line Plot

In [76]:
```
sns.lineplot(x=df.index, y = df['Volume'], label = 'Volume')
plt.title('volumn of stocks vs time')
```

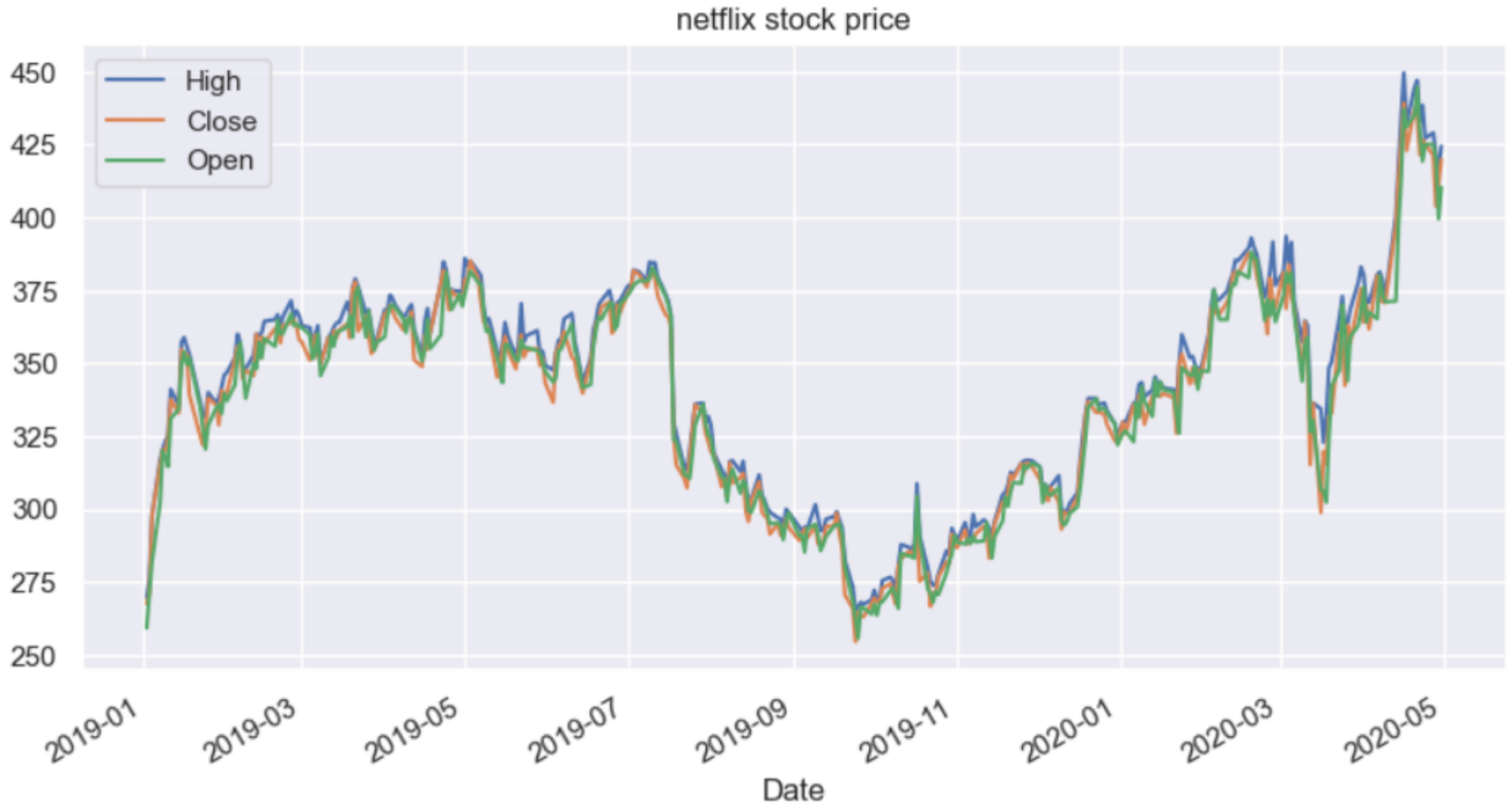
Out[76]:
```
Text(0.5, 1.0, 'volumn of stocks vs time')
```



Netflix Stock Price Analysis - High, Open, Close

In [77]:
```
df.plot(y =['High', 'Close', 'Open'], title = 'netflix stock price')
```

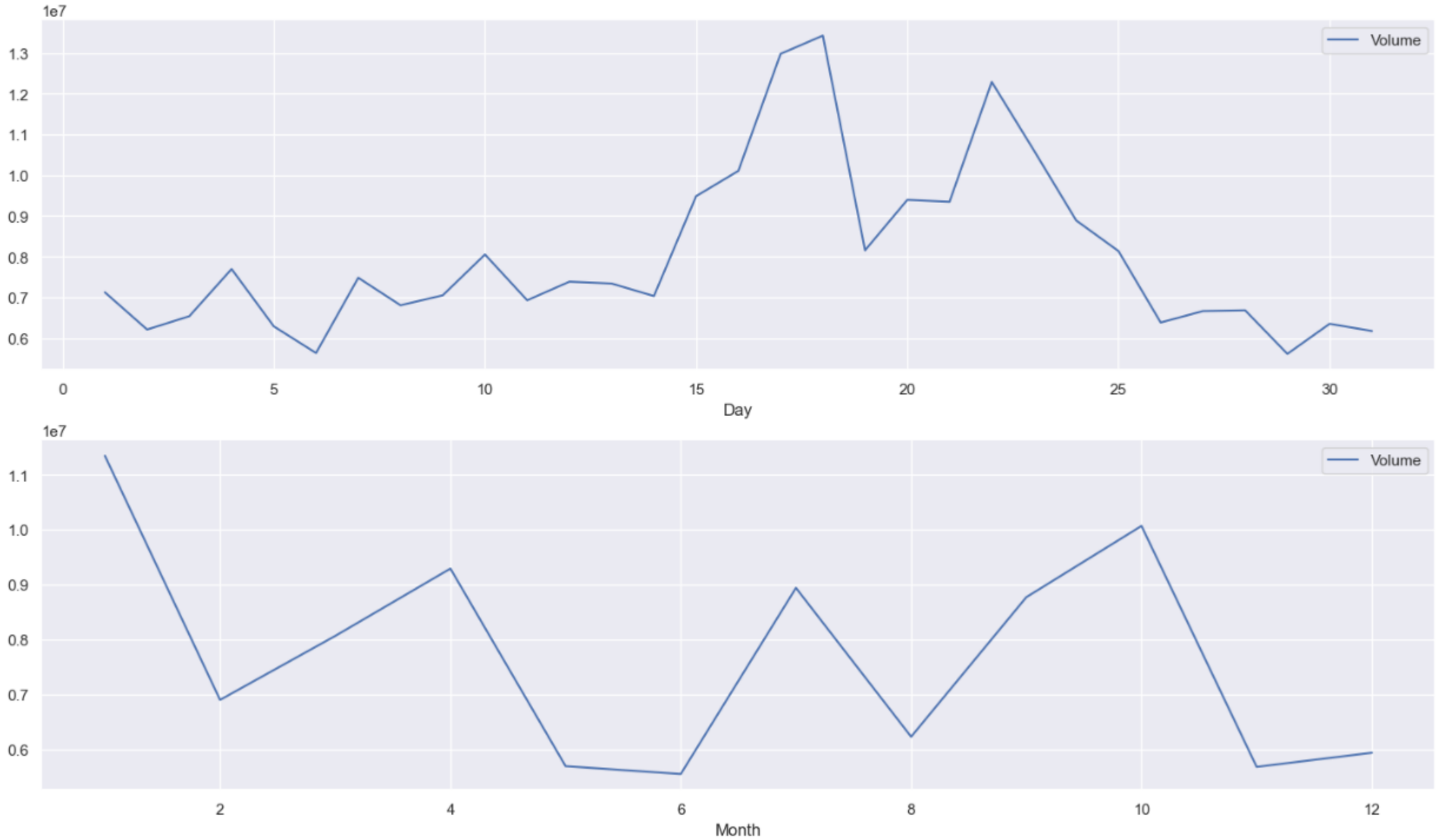
Out[77]:
```
<Axes: title={'center': 'netflix stock price'}, xlabel='Date'>
```



Netflix Stock Price Analysis - Day and Month Wise by subplots

In [84]:
```
fig, (ax1,ax2) = plt.subplots(2, figsize = (18,10))
df.groupby(df.index.day).mean().plot(y= 'Volume', ax= ax1, xlabel = 'Day')
df.groupby(df.index.month).mean().plot(y = 'Volume', ax = ax2, xlabel = 'Month')
```

Out[84]: <Axes: xlabel='Month'>



Top 5 Dates with highest stock price and lowest stock price

In [79]:
```
a = df.sort_values(by = 'High', ascending = False).head(5)
```

In [80]:
```
a['High']
```

Out[80]:
```
Date
2020-04-16    449.519989
2020-04-21    447.000000
2020-04-20    444.489990
2020-04-23    438.410004
2020-04-15    434.980011
Name: High, dtype: float64
```

In [81]:
```
b = df.sort_values(by = 'Low', ascending = True).head(5)
b['Low']
```

Out[81]:
```
Date
2019-09-24    252.279999
2019-09-25    253.699997
2019-01-02    256.579987
2019-10-03    257.010010
2019-09-26    260.200012
Name: Low, dtype: float64
```

Analysis of high and low stock per period of time by subplots

In [82]:
```
fig,axes = plt.subplots(nrows = 1, ncols= 2, sharex = True, figsize = (16,5))
fig.suptitle("High and Low values stock per period of time", fontsize = 18)
sns.lineplot(ax = axes[0], y= df['High'], x = df.index, color = 'green')
sns.lineplot(ax = axes[1],y = df['Low'], x = df.index, color = 'red')
```

Out[82]: <Axes: xlabel='Date', ylabel='Low'>