

Netflix stock analysis project

A Comprehensive Analysis of Netflix's Stock Performance

By - Sweety Srivastava



Agenda

1. Analyze the volume of Netflix stock traded over time.
2. Examine Netflix stock prices (high, open, close)
3. Breakdown Netflix stock prices by day, month, and year.
4. Identify the top 5 dates with the highest stock prices
5. Identify the top 5 dates with the lowest stock prices.

Imported several essential Python libraries

```
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
import numpy as np  
from datetime import datetime
```

1. Pandas: To organize and manage my data easily.
2. Matplotlib: To create clear charts and graphs.
3. Seaborn: To make my visuals more polished and appealing.
4. NumPy: To handle numbers and calculations efficiently.
5. Datetime: To work with dates and times seamlessly.

Loading the Netflix stock data from a CSV file

```
df = pd.read_csv("NFLX.csv")  
  
df
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2015-12-16	119.800003	123.000000	118.089996	122.639999	122.639999	13181000
1	2015-12-17	123.970001	126.349998	122.419998	122.510002	122.510002	17284900
2	2015-12-18	120.849998	122.190002	117.919998	118.019997	118.019997	17948100
3	2015-12-21	119.510002	119.589996	115.660004	116.629997	116.629997	11670000
4	2015-12-22	117.300003	117.430000	114.860001	116.239998	116.239998	9689000
...
1002	2019-12-10	296.119995	298.940002	292.019989	293.119995	293.119995	10476100
1003	2019-12-11	294.489990	299.429993	294.200012	298.929993	298.929993	5589800
1004	2019-12-12	295.670013	299.170013	295.059998	298.440002	298.440002	4766600
1005	2019-12-13	298.500000	301.799988	297.250000	298.500000	298.500000	3879700
1006	2019-12-16	300.850006	305.709991	298.630005	304.209991	304.209991	4658900

1007 rows × 7 columns

In this step - I'm loading the Netflix stock data from a CSV file into my project. The 'pd.read_csv("NFLX.csv")' command reads the data from the "NFLX.csv" file and stores it in a structured format called a DataFrame, which I can easily work with using Pandas. This is like opening a spreadsheet in Excel, but now I can analyze and manipulate the data directly in my code.

Previewing the First Few Rows of Data

```
: df.head()
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2015-12-16	119.800003	123.000000	118.089996	122.639999	122.639999	13181000
1	2015-12-17	123.970001	126.349998	122.419998	122.510002	122.510002	17284900
2	2015-12-18	120.849998	122.190002	117.919998	118.019997	118.019997	17948100
3	2015-12-21	119.510002	119.589996	115.660004	116.629997	116.629997	11670000
4	2015-12-22	117.300003	117.430000	114.860001	116.239998	116.239998	9689000

The `df.head()` command gives me a sneak peek at the first few rows of my data. It's like flipping to the first page of a book to see what it looks like and make sure everything is in order. This helps me quickly check that my data loaded correctly and understand what it looks like at a glance.

Set Date As Index

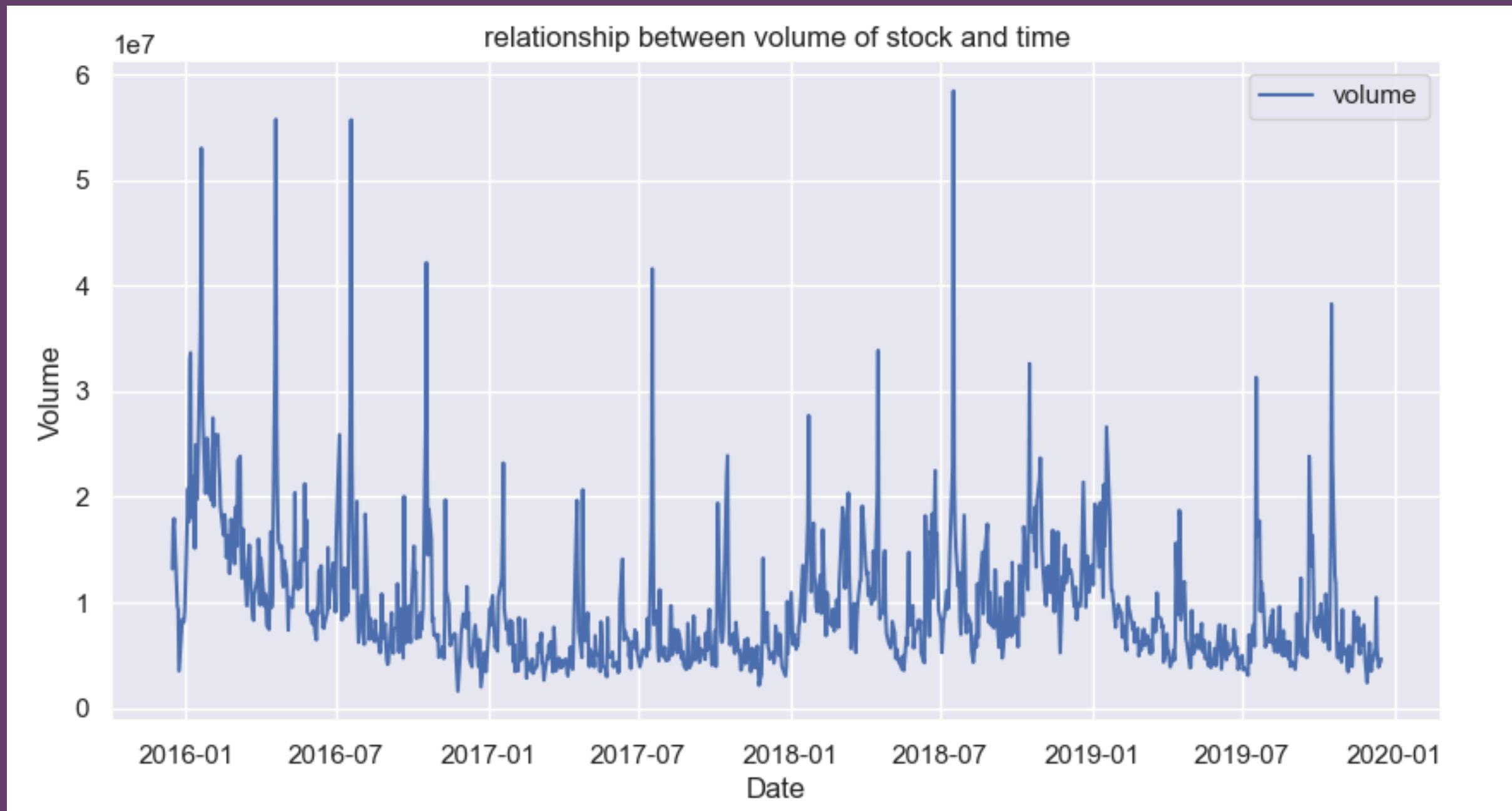
```
sns.set(rc = {'figure.figsize':(10,5)})  
  
df['Date']= pd.to_datetime(df['Date'])  
df = df.set_index('Date')  
df.head()
```

Date	Open	High	Low	Close	Adj Close	Volume
2015-12-16	119.800003	123.000000	118.089996	122.639999	122.639999	13181000
2015-12-17	123.970001	126.349998	122.419998	122.510002	122.510002	17284900
2015-12-18	120.849998	122.190002	117.919998	118.019997	118.019997	17948100
2015-12-21	119.510002	119.589996	115.660004	116.629997	116.629997	11670000
2015-12-22	117.300003	117.430000	114.860001	116.239998	116.239998	9689000

I first made sure my charts would be nice and clear by setting their size to 10 by 5 inches. Then, I changed the 'Date' column to an actual date format to make working with dates easier. I used this 'Date' column as the main index of my data to keep everything organized by date. Finally, I took a quick look at the first few rows to confirm that everything was set up correctly and working as expected.

Analyze the volume of Netflix stock traded over time.

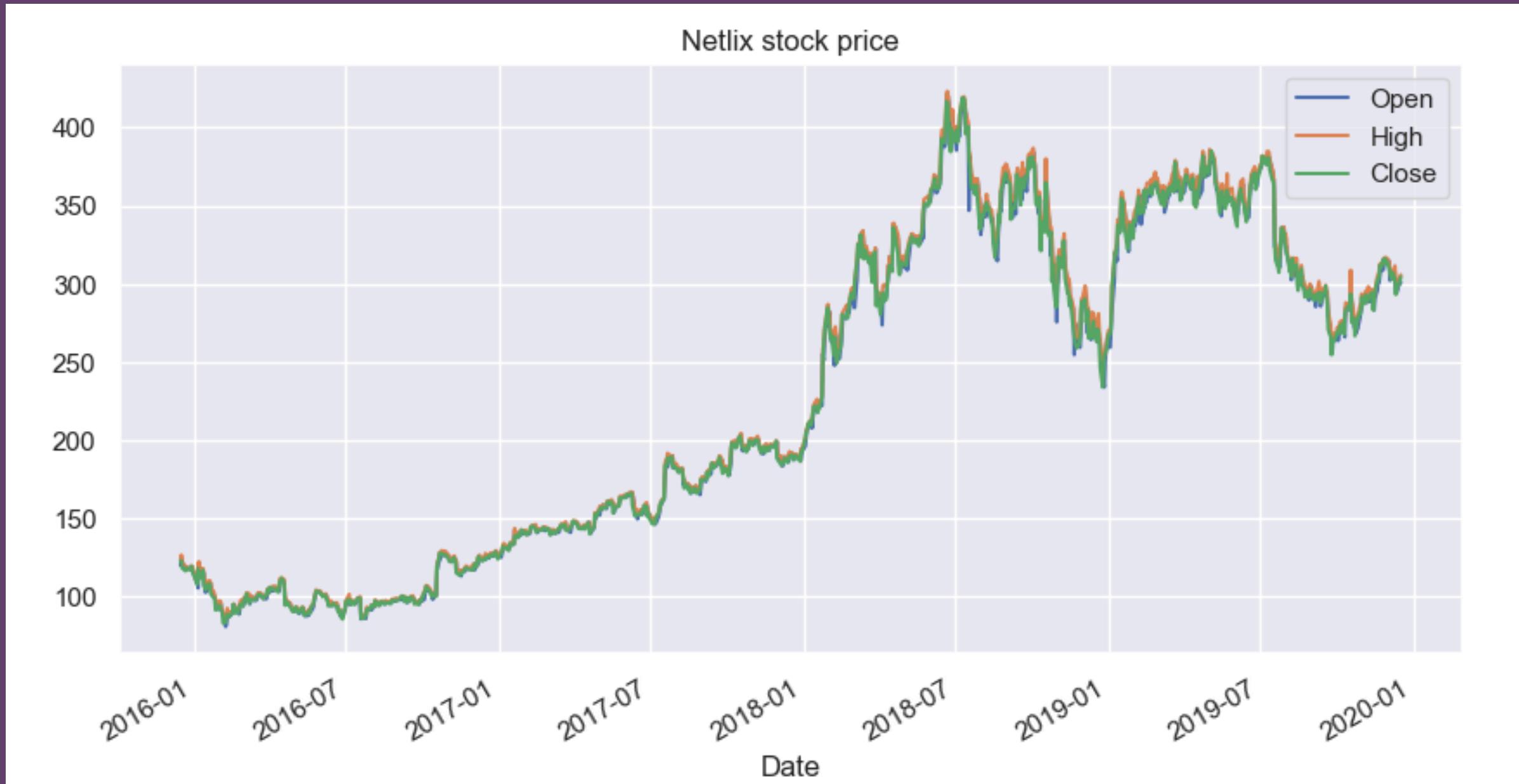
```
sns.lineplot(x = df.index, y = df['Volume'], label = 'volume')  
plt.title("relationship between volume of stock and time")
```



I made a line chart to visualize how Netflix's trading volume changes over time. I used Seaborn to plot the 'Volume' data against the dates, labeled the line as 'volume,' and added a title to show how trading activity relates to time. This helps me easily spot trends and fluctuations in how much the stock is being traded.

Netflix stock price- high , open, close

```
: df.plot (y =['Open', 'High','Close'],title = "Netflix stock price")
```



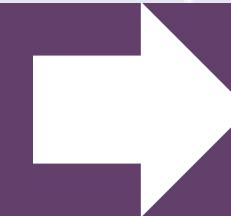
I made a chart to show Netflix's stock prices, including the opening, highest, and closing prices. This lets me see how each of these prices changes over time, and I gave the chart a title to make it clear that it's all about Netflix's stock prices. This way, I can easily compare the different prices and spot any trends.

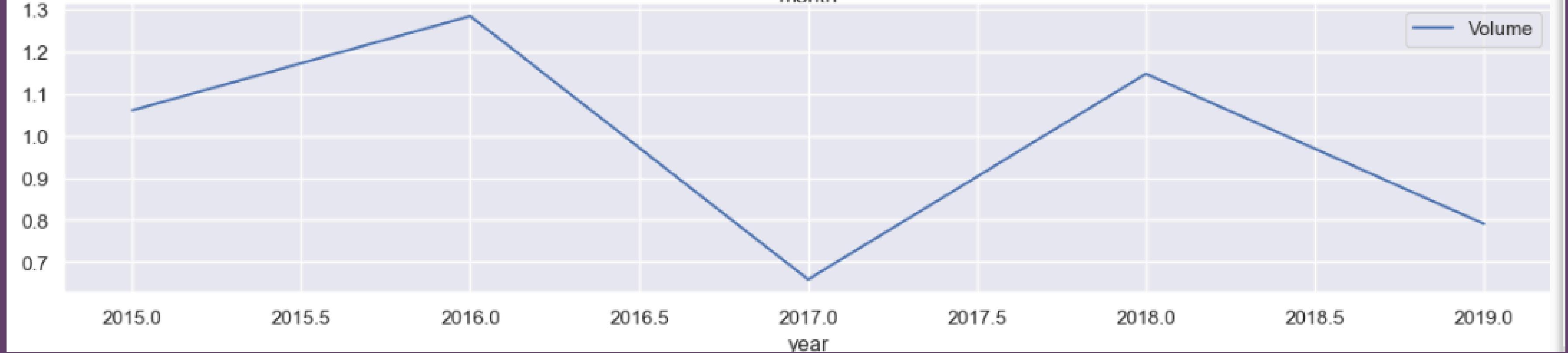
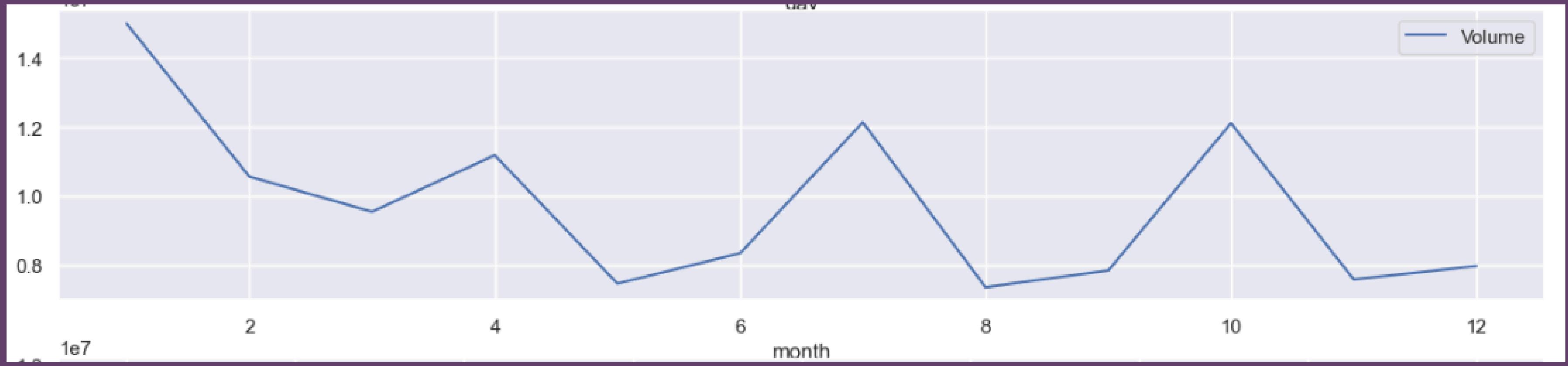
Netflix stock price - day, month , year-wise

```
fig,(ax1,ax2,ax3)= plt.subplots(3, figsize = (15,10))
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')
df.groupby(df.index.year).mean().plot(y = 'Volume', ax = ax3, xlabel = 'year')
```



Next





Next

1. I made three separate charts to look at Netflix's stock prices over different time frames:
2. Daily: The first chart shows the average stock prices for each day of the month.
3. Monthly: The second chart displays the average stock prices for each month.
4. Yearly: The third chart highlights the average stock prices for each year.
5. Each chart helps me see how stock prices change daily, monthly, and yearly, making it easier to spot trends and patterns over different periods.

Top 5 Dates With Highest Stock Price

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

Date	High
2018-06-21	423.209991
2018-06-22	420.500000
2018-07-11	419.769989
2018-06-20	419.470001
2018-07-10	419.440002

Name: High, dtype: float64

I identified the top 5 days when Netflix's stock price was at its highest. I sorted the data to find the days with the highest prices and then picked the top 5. Finally, I displayed these peak prices to see which days had the highest stock values.

Top 5 Dates With Lowest Stock Price

```
a = df.sort_values(by = 'Low', ascending = True).head(5)  
a['Low']
```

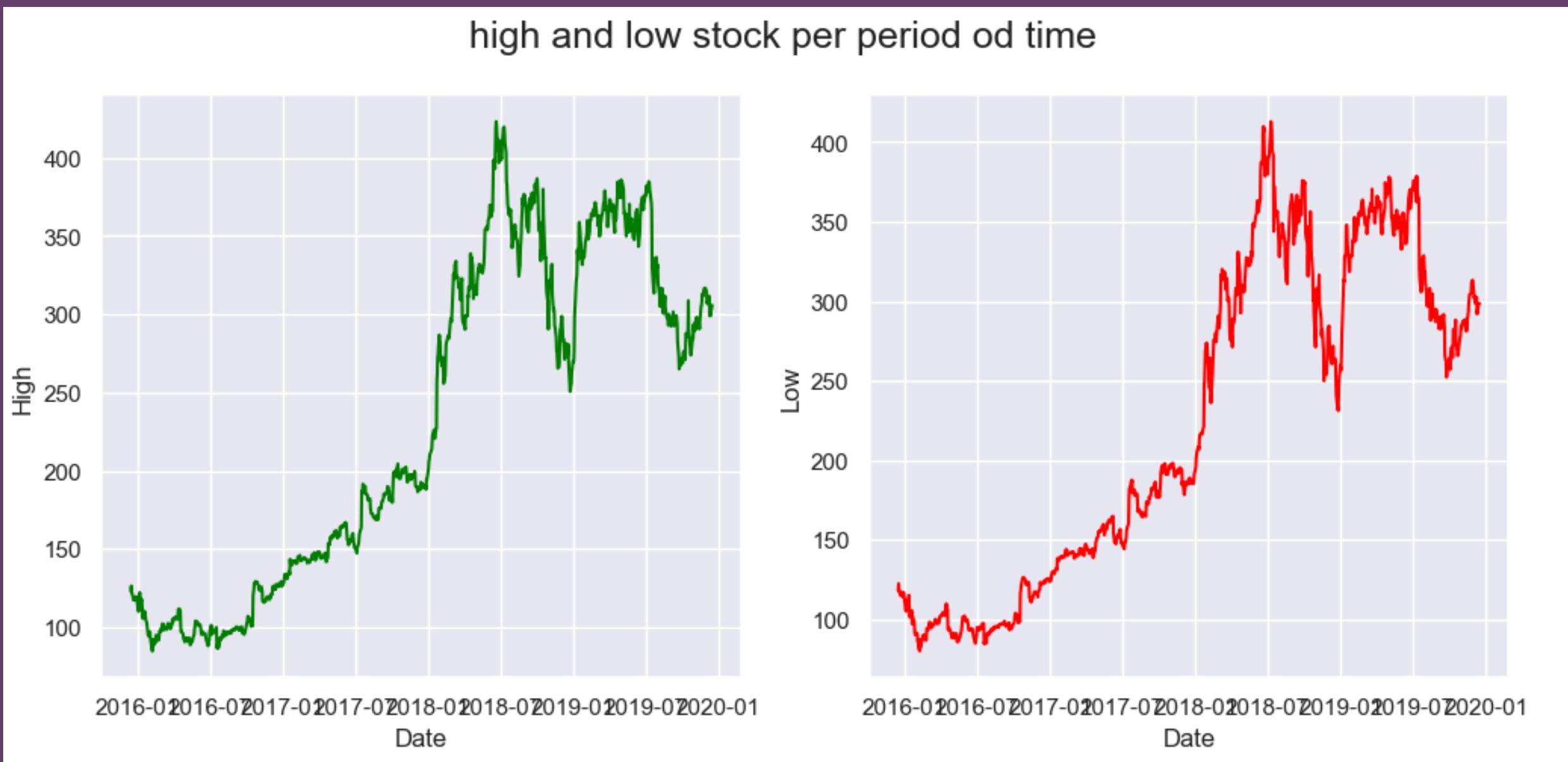
Date	Low
2016-02-08	79.949997
2016-02-09	81.550003
2016-02-05	81.860001
2016-02-11	82.910004
2016-07-19	84.500000

Name: Low, dtype: float64

I identified the top 5 days when Netflix's stock price was at its lowest. I sorted the data to find the days with the lowest prices and picked out the 5 days with the smallest values. Then, I showed these lowest prices to see which days had the lowest stock values.

Graph For High Low Price

```
fig,axes= plt.subplots(nrows = 1, ncols = 2,sharex = True, figsize = (12,5))
fig.suptitle('high and low stock per period od 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')
```



I set up two line charts next to each other to compare Netflix's highest and lowest stock prices:

1. High Prices: The first chart, in green, shows how the highest prices have changed over time.
2. Low Prices: The second chart, in red, tracks the lowest prices over time.

Both charts share the same time axis, so it's easy to see how high and low prices compare across the same periods. The title at the top makes it clear that the charts show stock price highs and lows over time.

Conclusion

In this project, I explored Netflix's stock data to uncover important trends and patterns. By looking at how much stock was traded and how prices changed over time—whether daily, monthly, or yearly—I was able to spot key movements in the market. I also identified the top 5 days when Netflix's stock prices were at their highest and lowest, giving a clear snapshot of the stock's best and worst days.

This analysis helped me better understand how Netflix's stock performs, which could be useful for making smart decisions about buying or selling the stock. The charts and graphs I created made it easy to see these trends and outliers, providing a clear summary of Netflix's stock behavior.

Thankyou