

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

In [2]: data=pd.read_csv('downloads\GOOG.L.csv')

In [3]: data

Out[3]:
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2004-08-19	50.050049	52.082081	48.028027	50.220219	50.220219	44659096
1	2004-08-20	50.555557	54.594597	50.303001	54.209209	54.209209	22834343
2	2004-08-23	55.430431	56.796799	54.579578	54.754753	54.754753	18256126
3	2004-08-24	55.675674	55.855858	51.836838	52.487488	52.487488	15247337
4	2004-08-25	52.532532	54.054054	51.991993	53.053055	53.053055	9188602
...
4426	2022-03-18	2668.489990	2724.879883	2645.169922	2722.510010	2722.510010	2223100
4427	2022-03-21	2723.270020	2741.000000	2681.850098	2722.030029	2722.030029	1341600
4428	2022-03-22	2722.030029	2821.000000	2722.030029	2797.360107	2797.360107	1774800
4429	2022-03-23	2774.050049	2791.770020	2756.699951	2765.510010	2765.510010	1257700
4430	2022-03-24	2784.000000	2832.379883	2755.010010	2831.439941	2831.439941	1317900

4431 rows x 7 columns

```
In [4]: data.head()

Out[4]:
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2004-08-19	50.050049	52.082081	48.028027	50.220219	50.220219	44659096
1	2004-08-20	50.555557	54.594597	50.303001	54.209209	54.209209	22834343
2	2004-08-23	55.430431	56.796799	54.579578	54.754753	54.754753	18256126
3	2004-08-24	55.675674	55.855858	51.836838	52.487488	52.487488	15247337
4	2004-08-25	52.532532	54.054054	51.991993	53.053055	53.053055	9188602

```
In [5]: data.tail()

Out[5]:
```

	Date	Open	High	Low	Close	Adj Close	Volume
4426	2022-03-18	2668.489990	2724.879883	2645.169922	2722.510010	2722.510010	2223100
4427	2022-03-21	2723.270020	2741.000000	2681.850098	2722.030029	2722.030029	1341600
4428	2022-03-22	2722.030029	2821.000000	2722.030029	2797.360107	2797.360107	1774800
4429	2022-03-23	2774.050049	2791.770020	2756.699951	2765.510010	2765.510010	1257700
4430	2022-03-24	2784.000000	2832.379883	2755.010010	2831.439941	2831.439941	1317900

```
In [6]: data.shape

Out[6]: (4431, 7)

In [7]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4431 entries, 0 to 4430
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
--  --
 0   Date        4431 non-null    object
 1   Open        4431 non-null    float64
 2   High        4431 non-null    float64
 3   Low         4431 non-null    float64
 4   Close       4431 non-null    float64
 5   Adj Close   4431 non-null    float64
 6   Volume      4431 non-null    int64
dtypes: float64(6), int64(1), object(1)
memory usage: 242.4+ KB

In [8]: data.index

Out[8]: RangeIndex(start=0, stop=4431, step=1)

In [9]: data.columns

Out[9]: Index(['Date', 'Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'], dtype=object)

In [10]: data.describe()

Out[10]:
```

	Open	High	Low	Close	Adj Close	Volume
count	4431.000000	4431.000000	4431.000000	4431.000000	4431.000000	4.431000e+03
mean	603.087345	699.735595	686.078751	693.097367	693.097367	6.444992e+06
std	645.118799	651.331215	638.579488	645.107806	645.107806	7.692038e+06
min	49.644646	50.9320921	48.028027	50.059054	50.059054	4.656000e+05
25%	248.558563	250.853355	245.813309	248.415916	248.415916	1.695600e+06
50%	434.924827	437.887878	432.687893	435.330322	435.330322	3.778418e+06
75%	1007.364890	1020.649994	997.274994	1007.790008	1007.790008	8.002300e+06
max	3025.000000	3030.929932	2977.979980	2996.770020	2996.770020	8.215117e+07

```
In [11]: data.isna().any()

Out[11]:
```

Date	Open	High	Low	Close	Adj Close	Volume
False	False	False	False	False	False	False
False	False	False	False	False	False	False
False	False	False	False	False	False	False
False	False	False	False	False	False	False
False	False	False	False	False	False	False
False	False	False	False	False	False	False
dtype: bool						

```
In [12]: data.duplicated().sum()

Out[12]: 0

In [13]: data

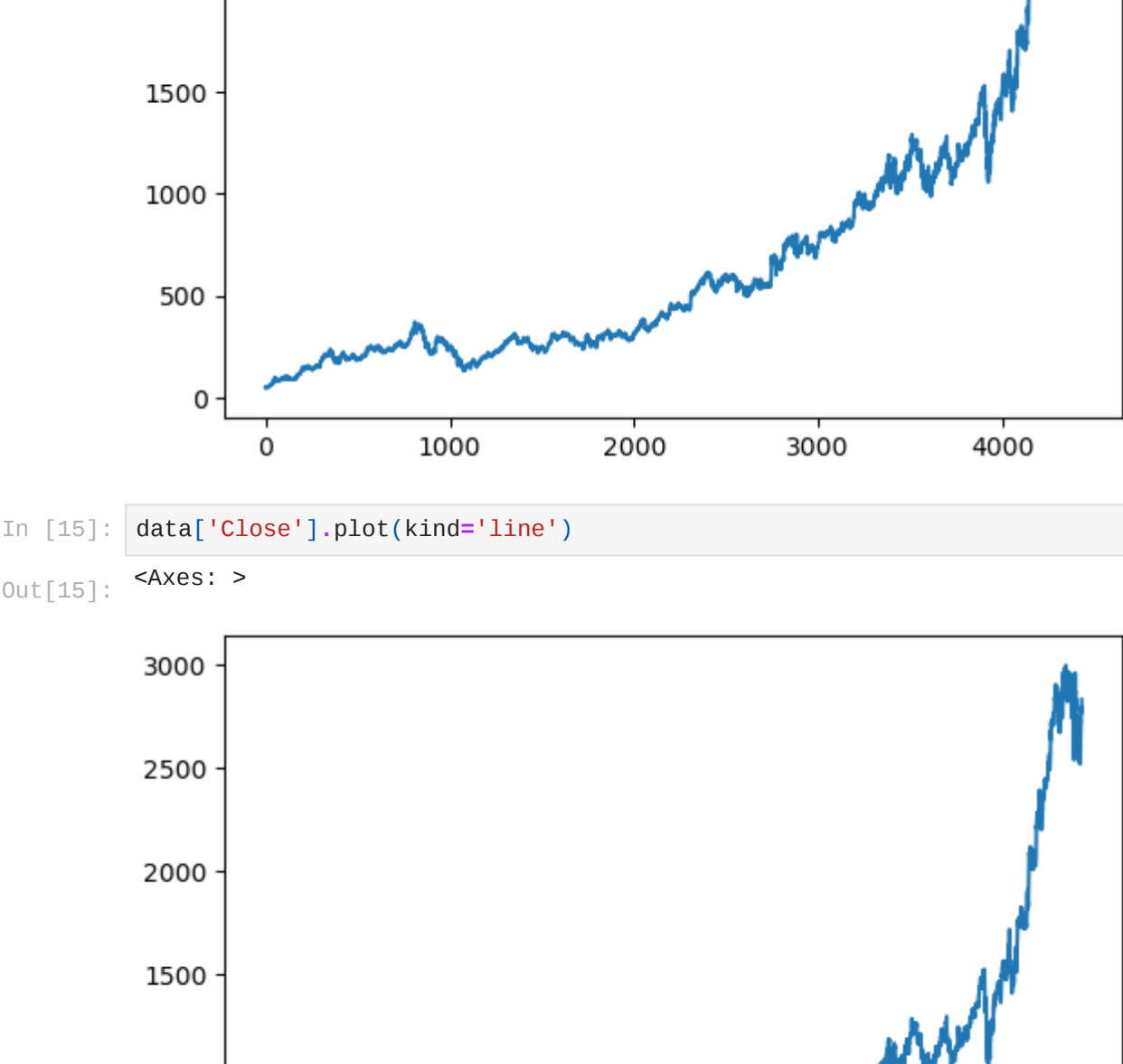
Out[13]:
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2004-08-19	50.050049	52.082081	48.028027	50.220219	50.220219	44659096
1	2004-08-20	50.555557	54.594597	50.303001	54.209209	54.209209	22834343
2	2004-08-23	55.430431	56.796799	54.579578	54.754753	54.754753	18256126
3	2004-08-24	55.675674	55.855858	51.836838	52.487488	52.487488	15247337
4	2004-08-25	52.532532	54.054054	51.991993	53.053055	53.053055	9188602
...
4426	2022-03-18	2668.489990	2724.879883	2645.169922	2722.510010	2722.510010	2223100
4427	2022-03-21	2723.270020	2741.000000	2681.850098	2722.030029	2722.030029	1341600
4428	2022-03-22	2722.030029	2821.000000	2722.030029	2797.360107	2797.360107	1774800
4429	2022-03-23	2774.050049	2791.770020	2756.699951	2765.510010	2765.510010	1257700
4430	2022-03-24	2784.000000	2832.379883	2755.010010	2831.439941	2831.439941	1317900

4431 rows x 7 columns

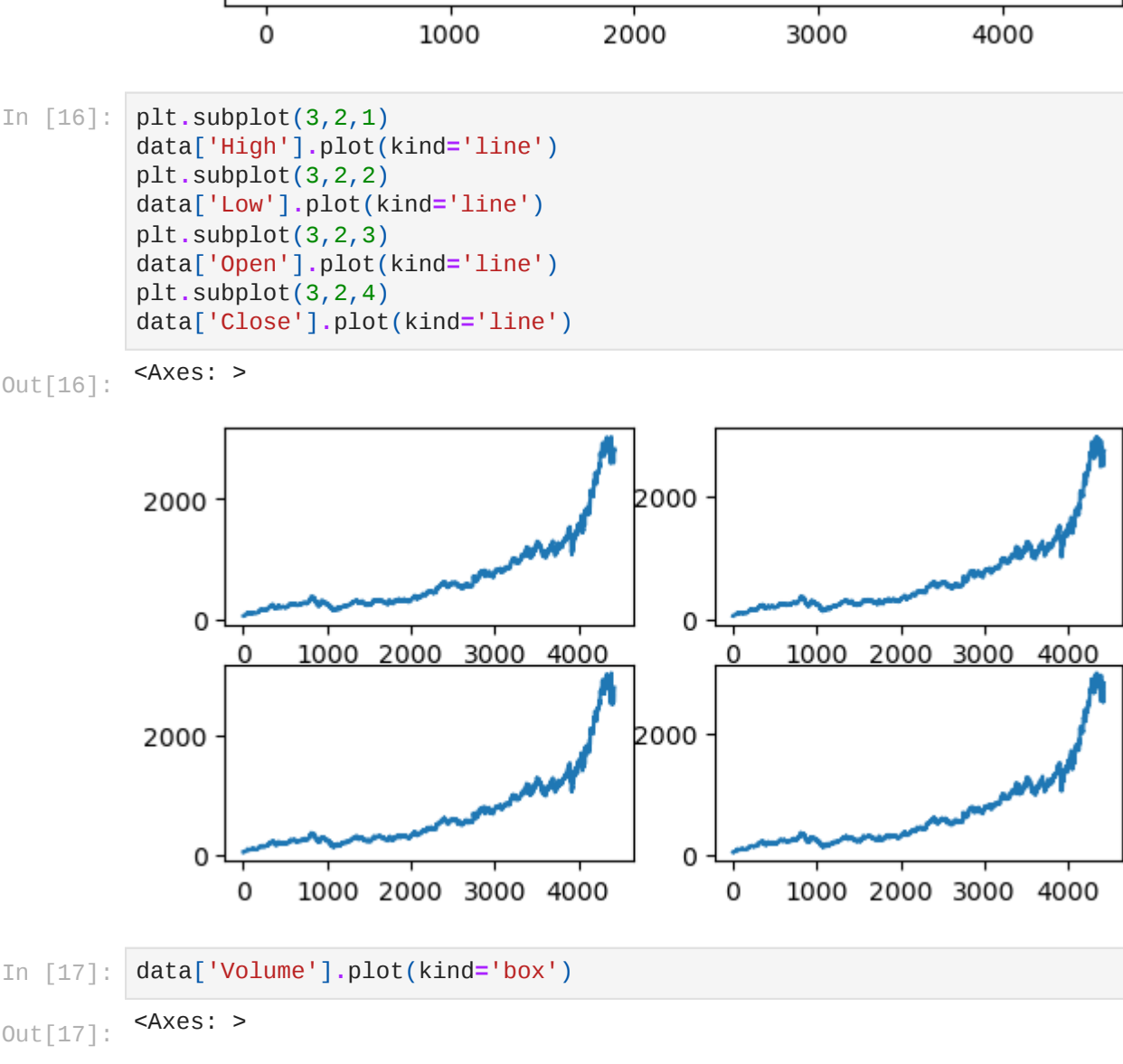
```
In [14]: data["Open"].plot(kind='line')

Out[14]: <Axes: >
```



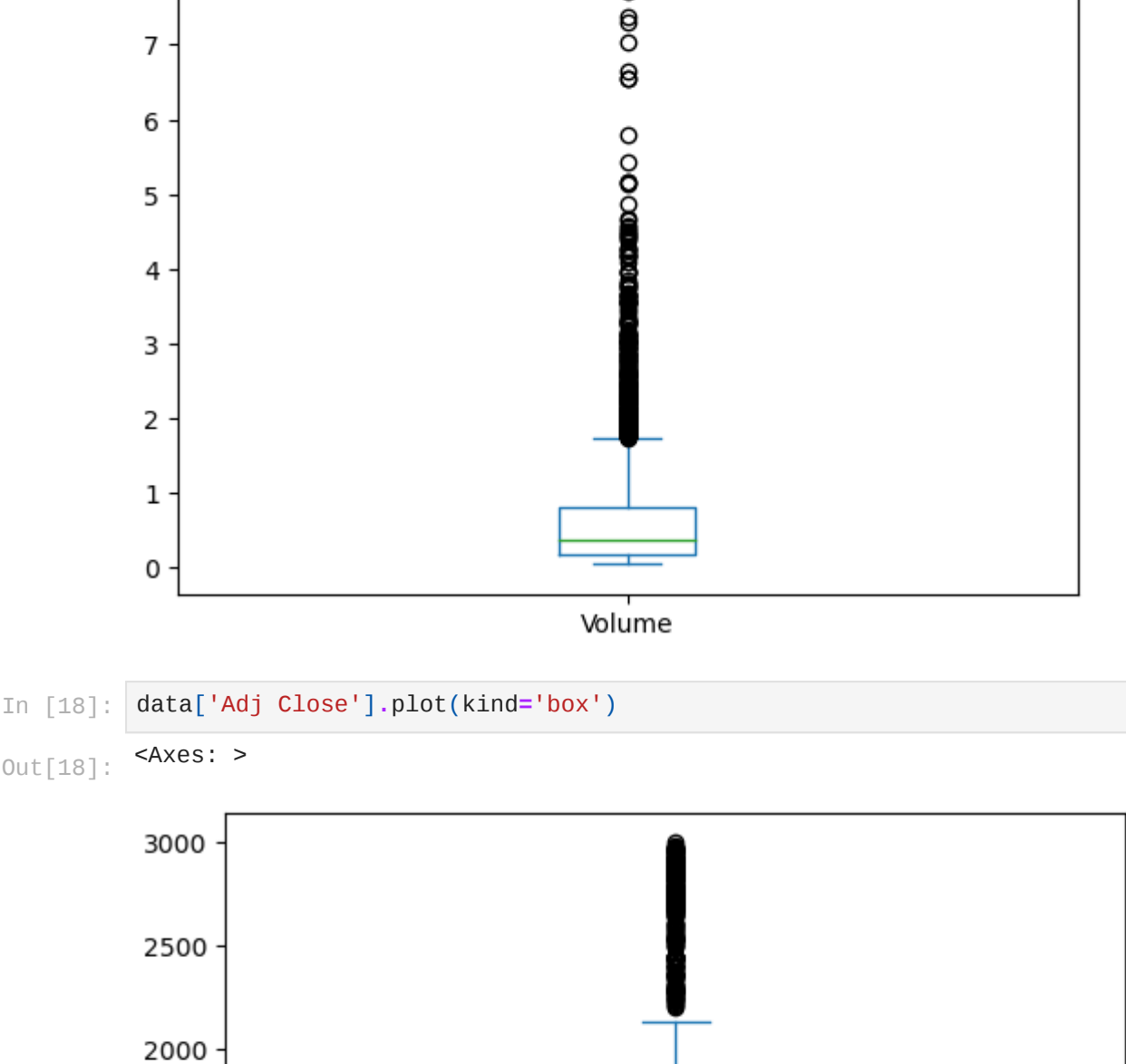
```
In [15]: data["Close"].plot(kind='line')

Out[15]: <Axes: >
```



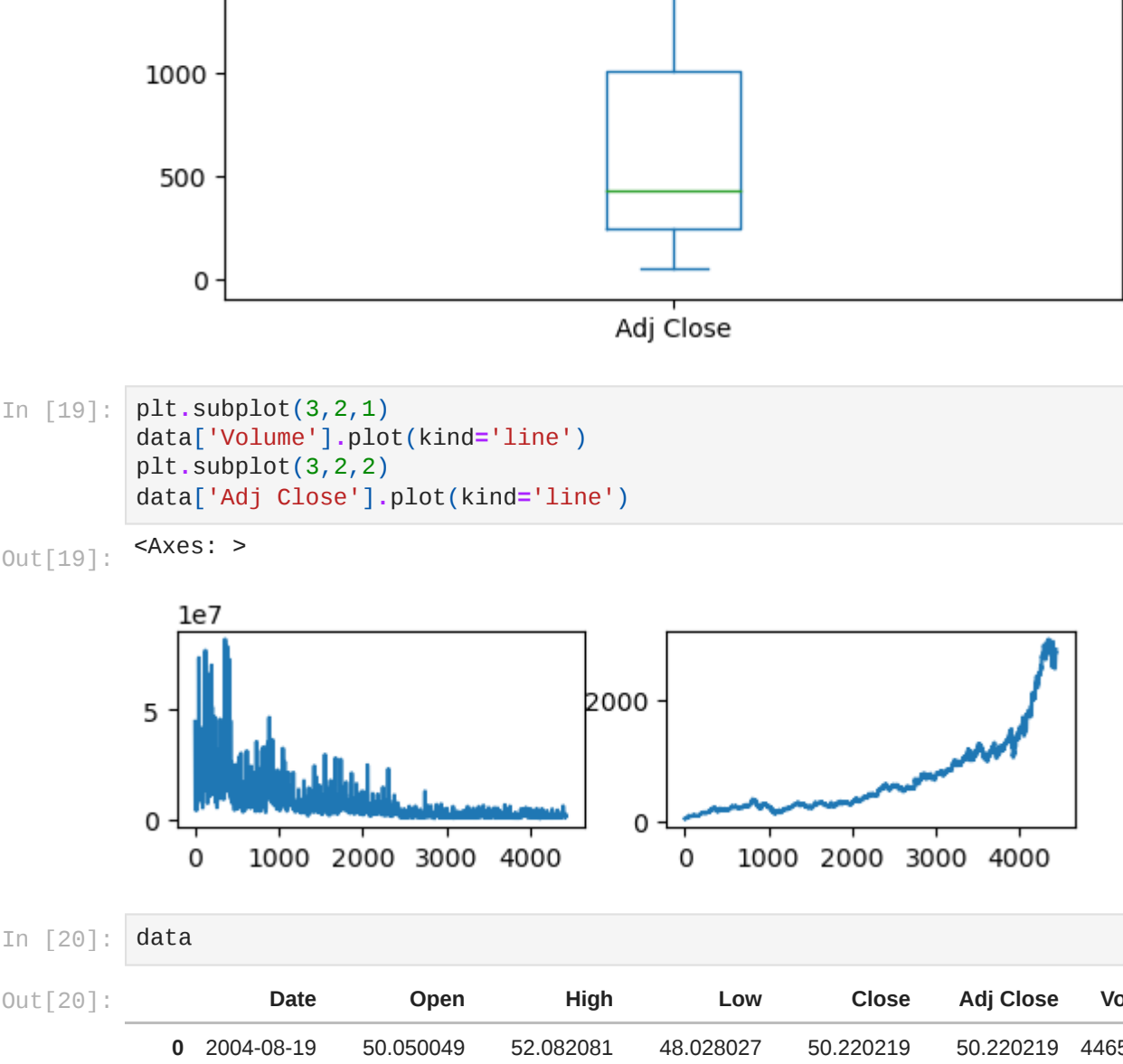
```
In [16]: plt.subplot(3,2,1)
data["High"].plot(kind='line')
plt.subplot(3,2,2)
data["Low"].plot(kind='line')
plt.subplot(3,2,3)
data["Volume"].plot(kind='line')
plt.subplot(3,2,4)
data["Adj Close"].plot(kind='line')

Out[16]: <Axes: >
```



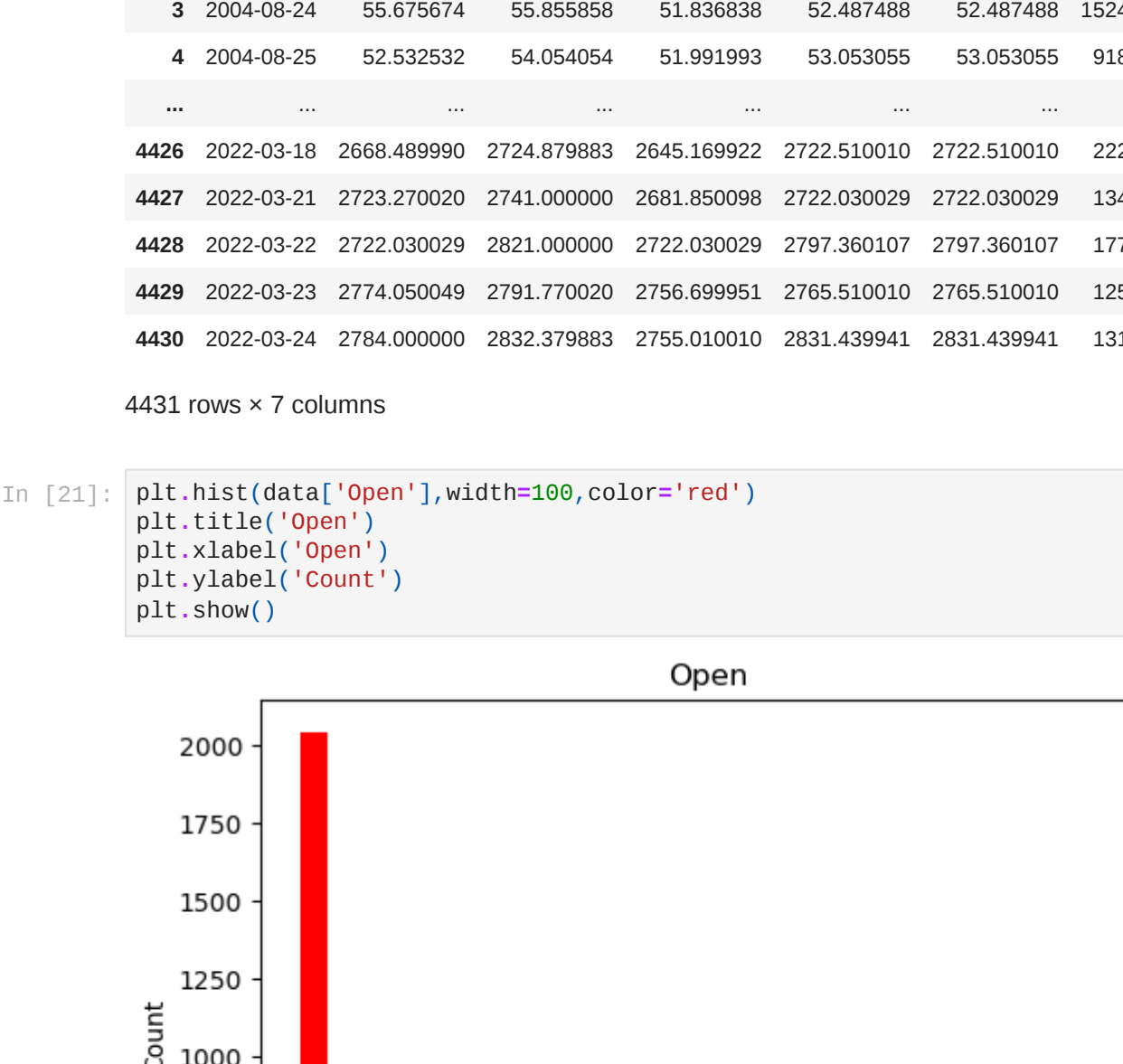
```
In [17]: data["Volume"].plot(kind='box')

Out[17]: <Axes: >
```




```
In [18]: data["Adj Close"].plot(kind='box')

Out[18]: <Axes: >
```



```
In [19]: plt.subplot(3,2,3)
data["Volume"].plot(kind='line')
plt.subplot(3,2,2)
data["Adj Close"].plot(kind='line')

Out[19]: <Axes: >
```



```
In [20]: data

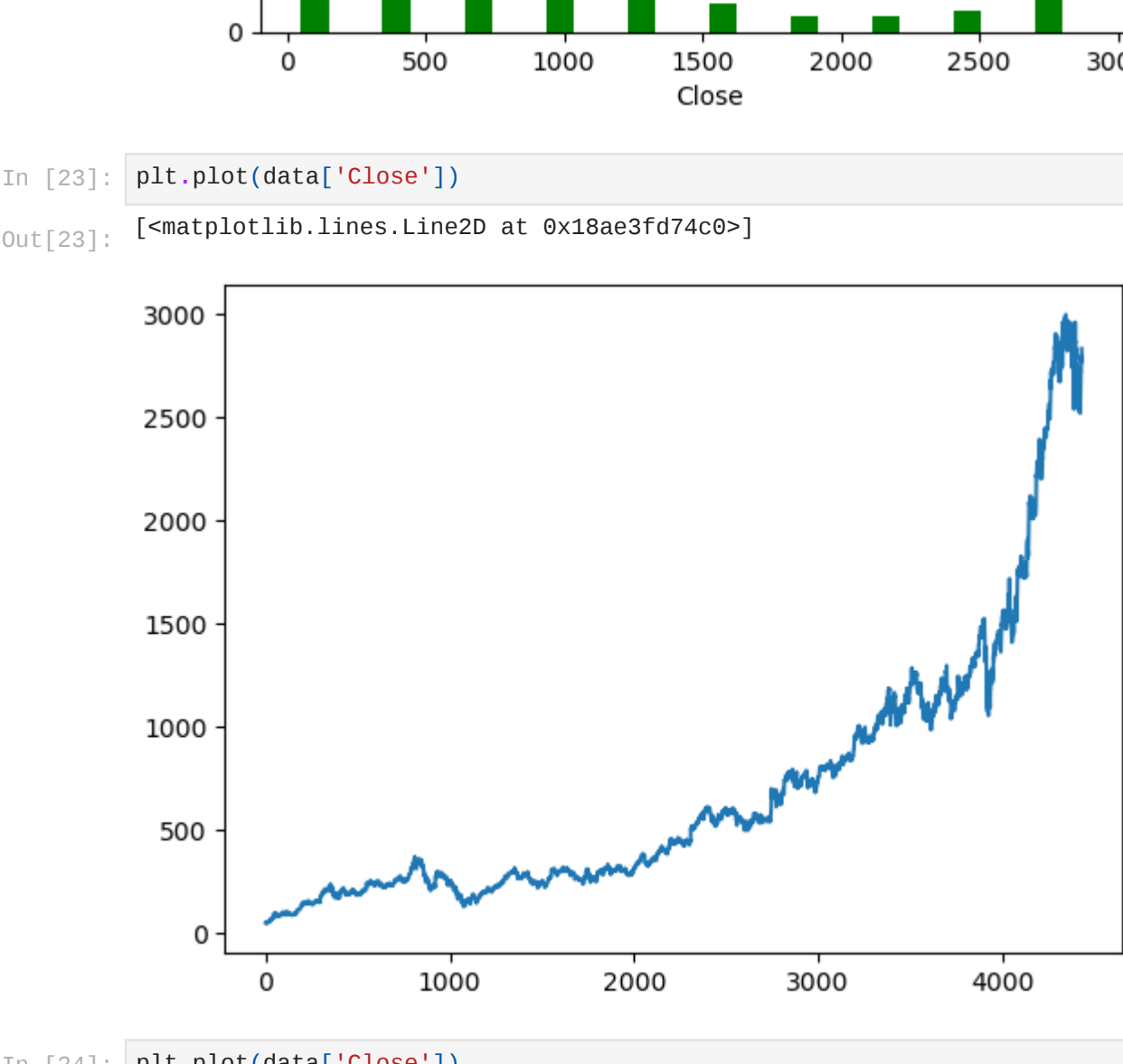
Out[20]:
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2004-08-19	50.050049	52.082081	48.028027	50.220219	50.220219	44659096
1	2004-08-20	50.555557	54.594597	50.303001	54.209209	54.209209	22834343
2	2004-08-23	55.430431	56.796799	54.579578	54.754753	54.754753	18256126
3	2004-08-24	55.675674	55.855858	51.836838	52.487488	52.487488	15247337
4	2004-08-25	52.532532	54.054054	51.991993	53.053055	53.053055	9188602
...
4426	2022-03-18	2668.489990	2724.879883	2645.169922	2722.510010	2722.510010	2223100
4427	2022-03-21	2723.270020	2741.000000	2681.850098	2722.030029	2722.030029	1341600
4428	2022-03-22	2722.030029	2821.000000	2722.030029	2797.360107	2797.360107	1774800
4429	2022-03-23	2774.050049	2791.770020	2756.699951	2765.510010	2765.510010	1257700
4430	2022-03-24	2784.000000	2832.379883	2755.010010	2831.439941	2831.439941	1317900

4431 rows x 7 columns

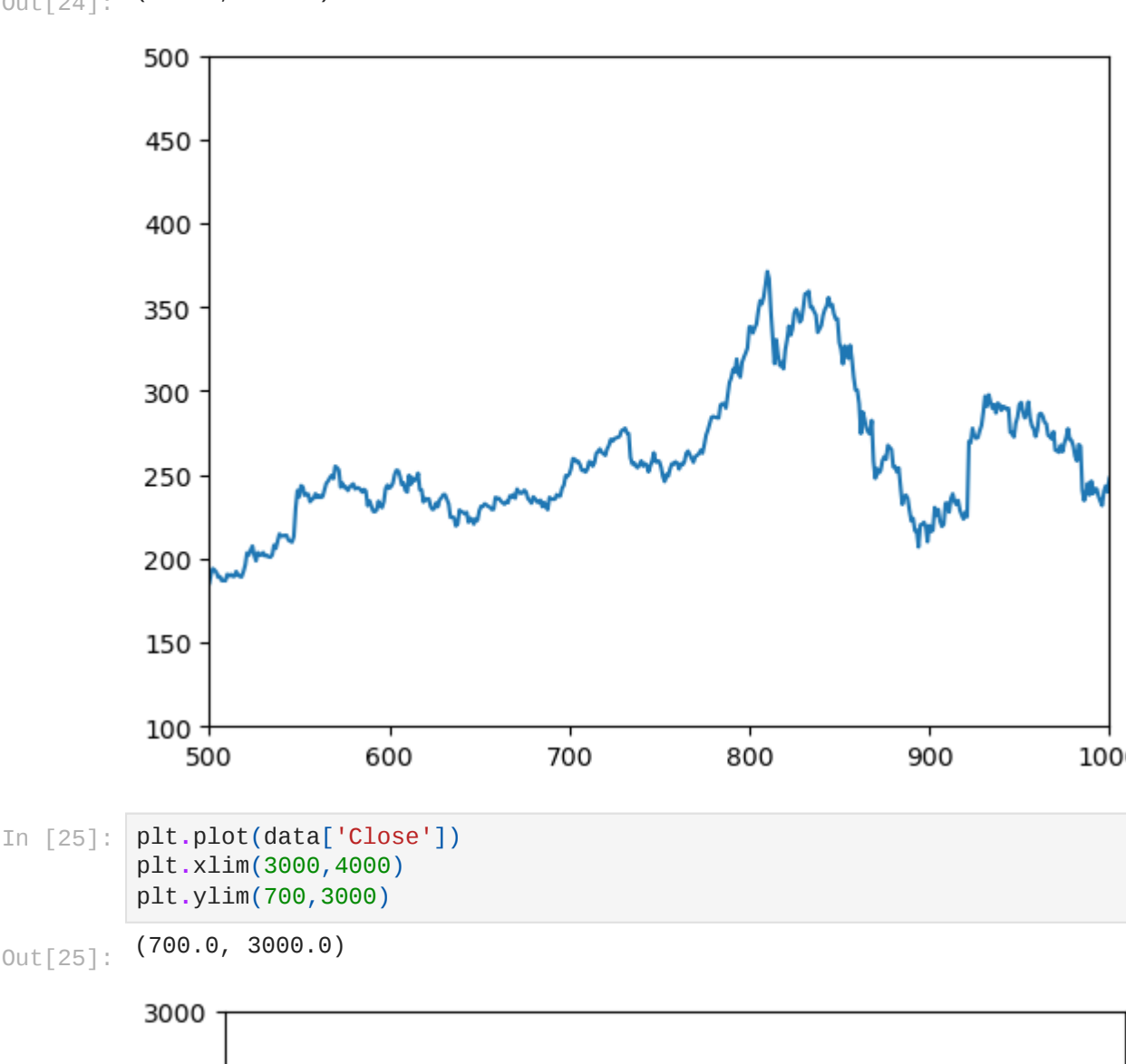
```
In [21]: plt.hist(data["Open"],width=100,color='red')
plt.title("Open")
plt.xlabel("Count")
plt.ylabel("Count")
plt.show()

Out[21]:
```



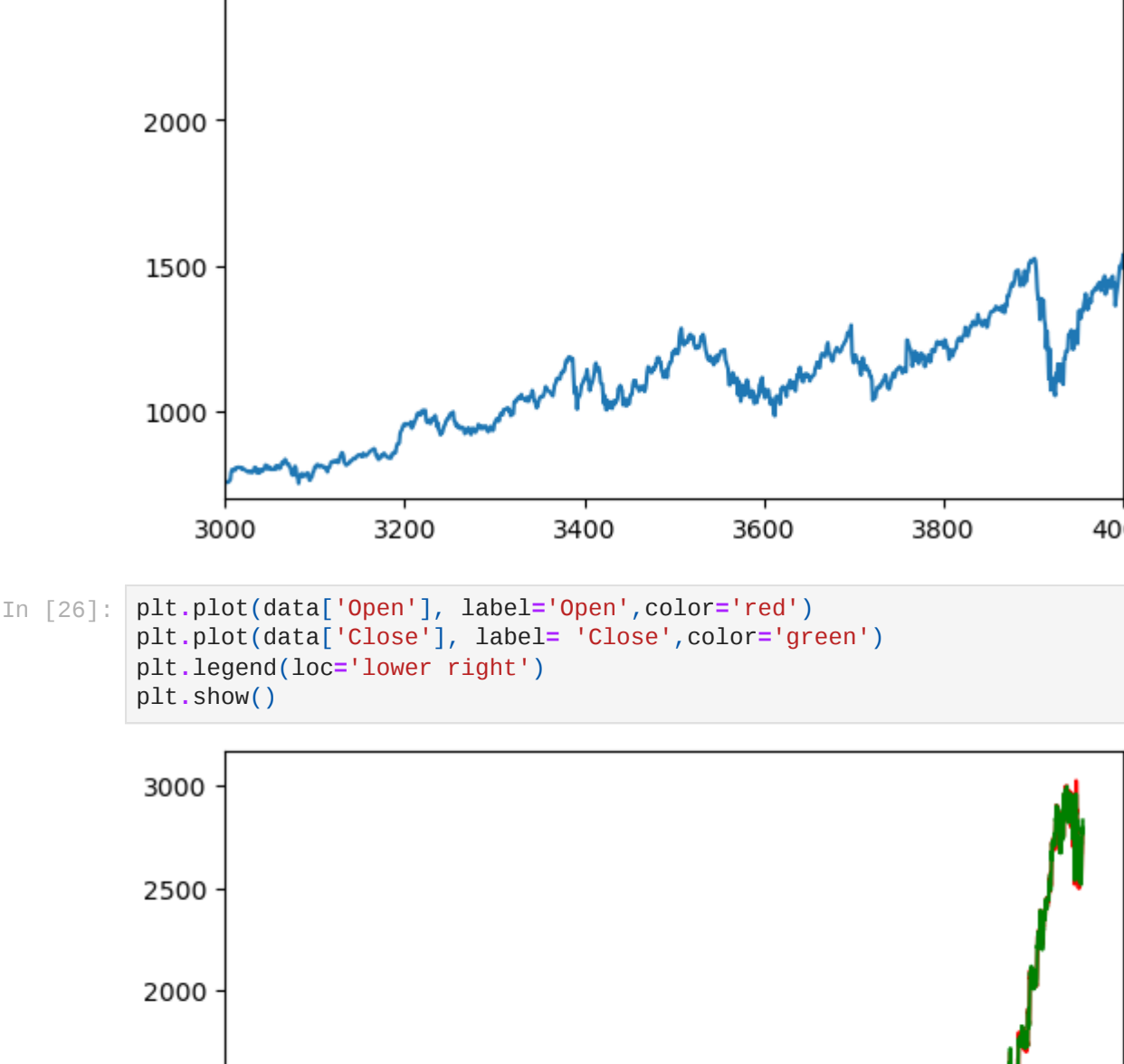
```
In [22]: plt.hist(data["Close"],width=100,color='green')
plt.title("Close")
plt.xlabel("Count")
plt.ylabel("Count")
plt.show()

Out[22]:
```



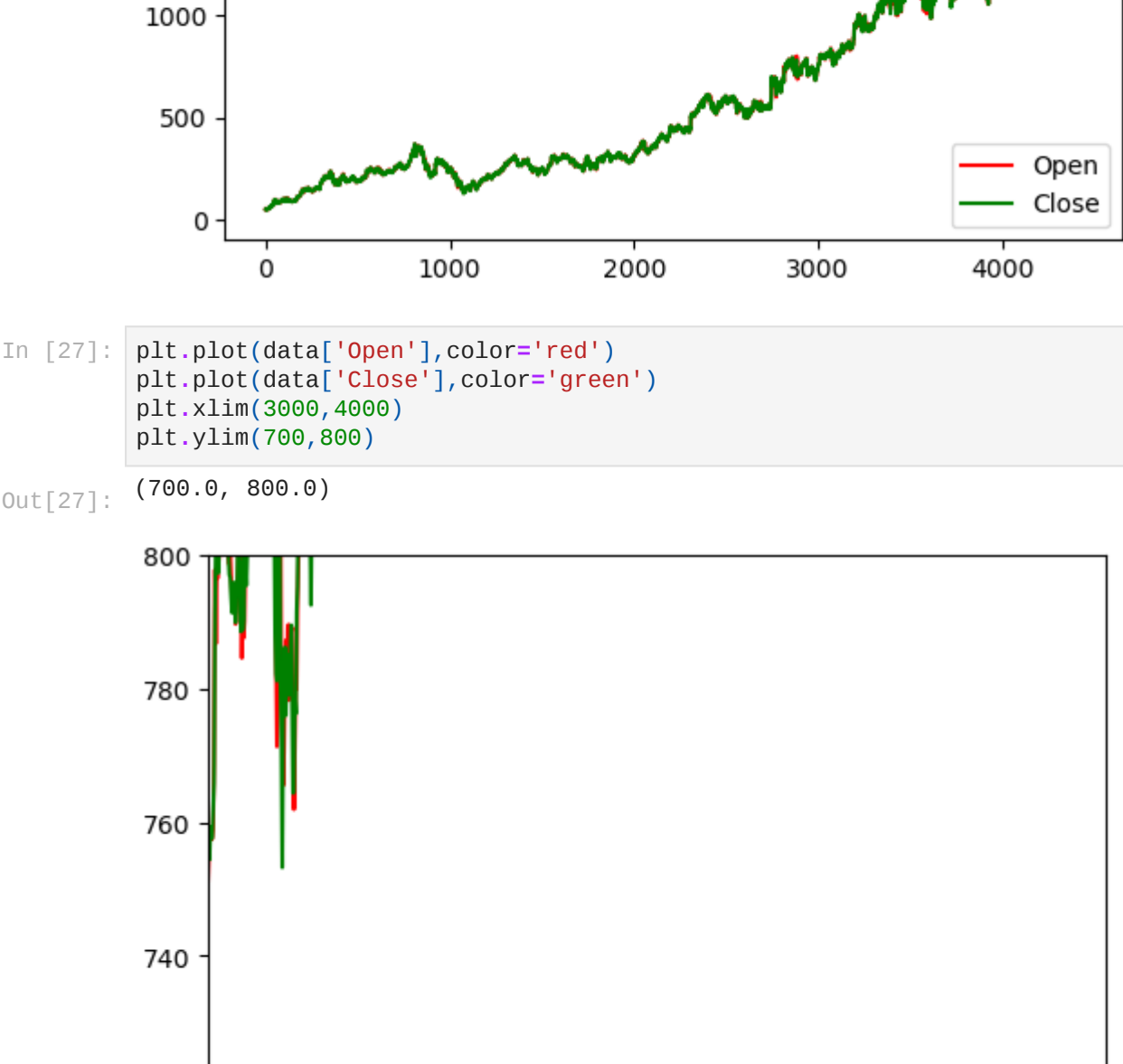
```
In [23]: plt.plot(data["Close"])

Out[23]: [matplotlib.lines.Line2D at 0x18ae3f674c8]
```



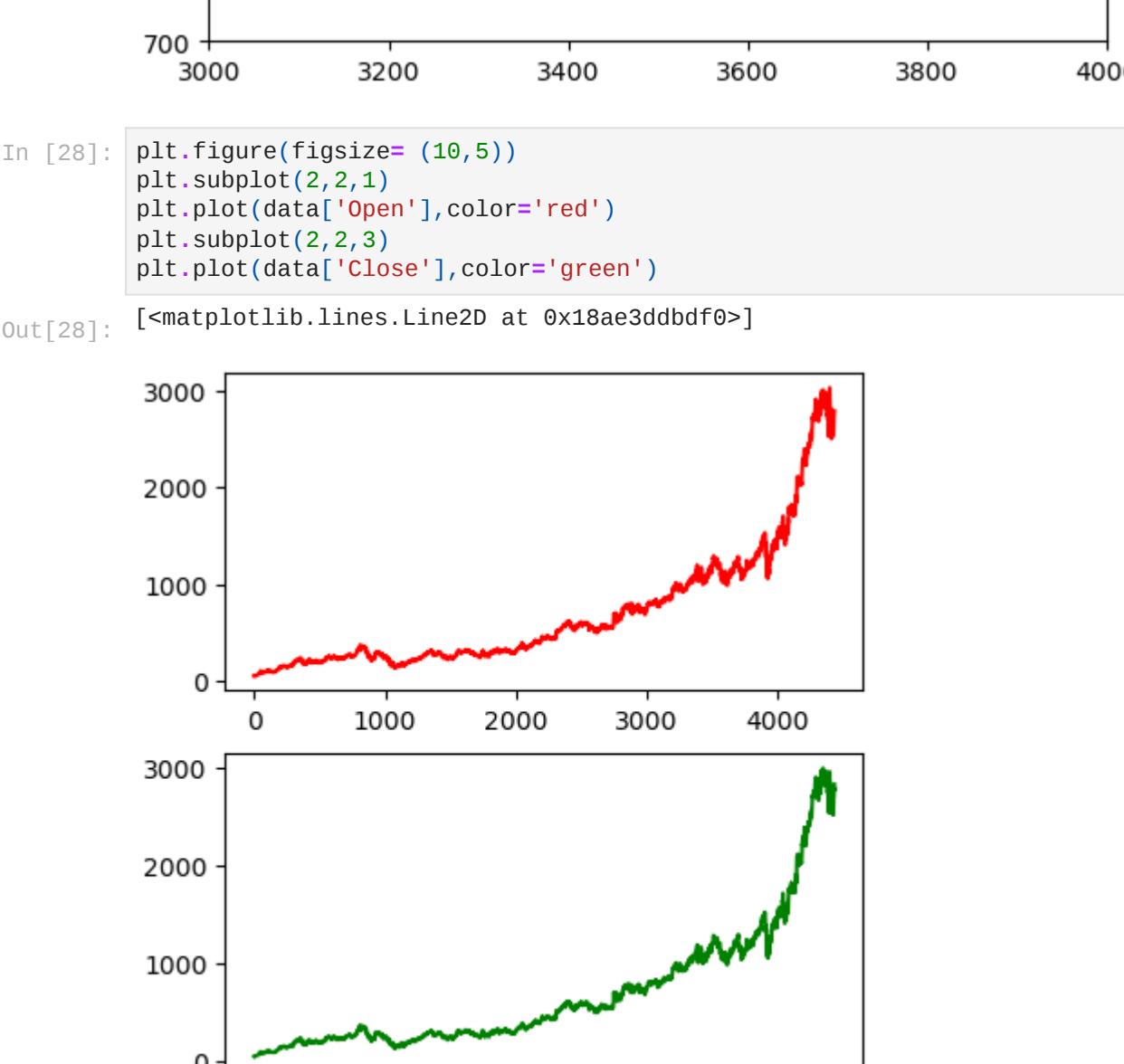
```
In [24]: plt.plot(data["Close"])
plt.xlabel(1000,1000)
plt.ylabel(100,500)

Out[24]: (100.0, 500.0)
```



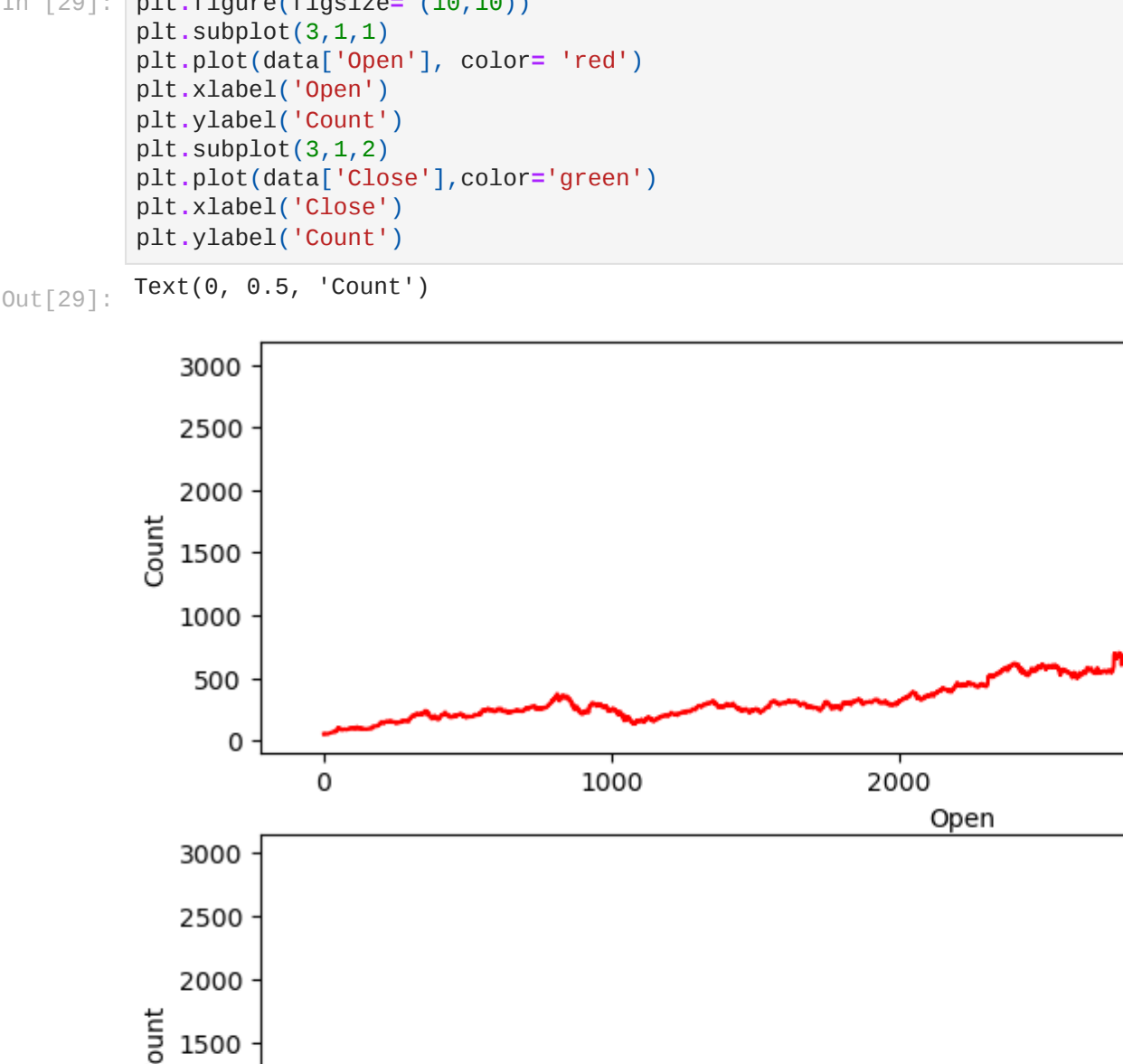
```
In [25]: plt.plot(data["Close"])
plt.xlabel(1000,1000)
plt.ylabel(100,500)

Out[25]: (100.0, 500.0)
```



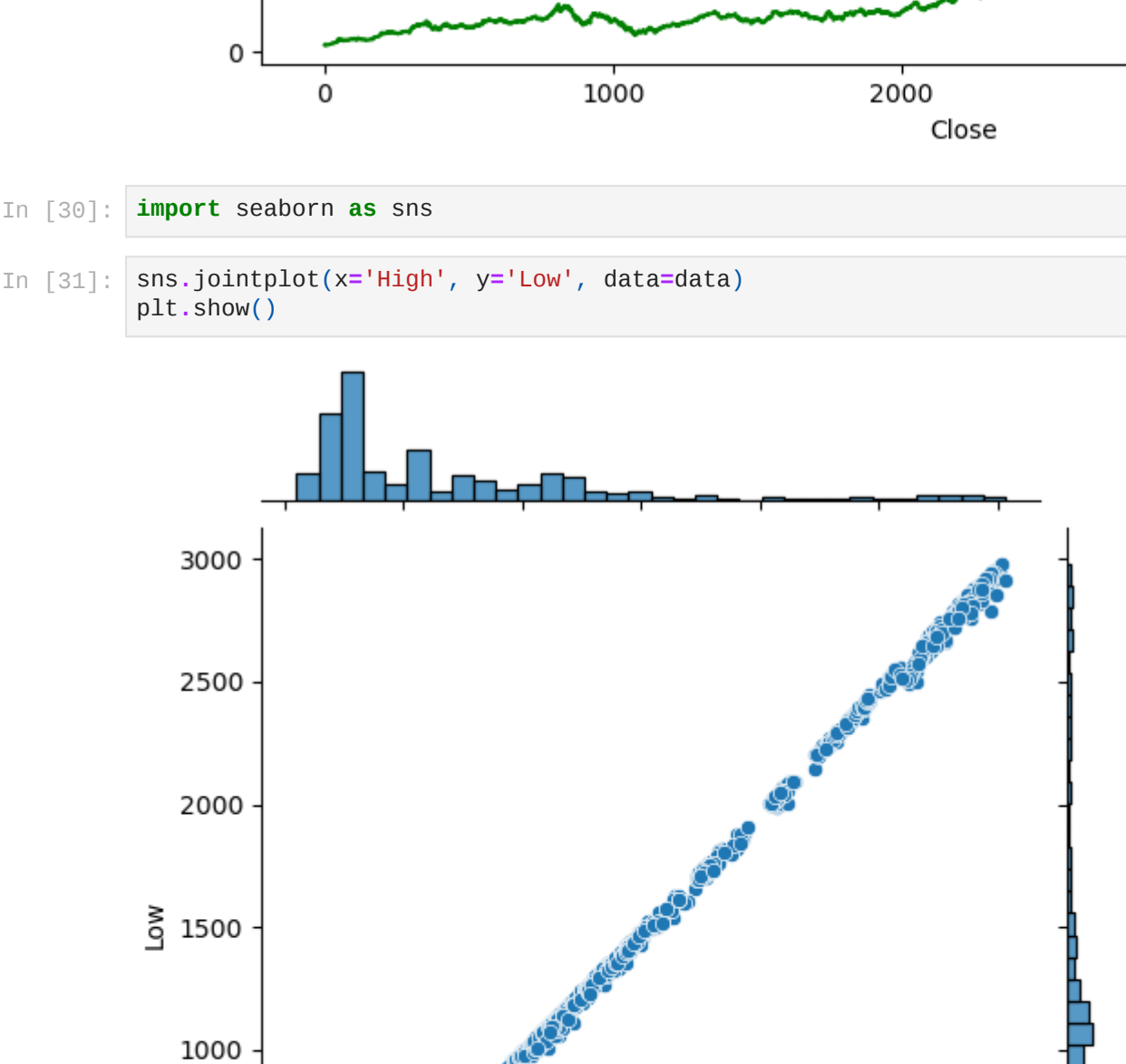
```
In [26]: plt.plot(data["Open"], label='Open',color='red')
plt.plot(data["Close"], label='Close',color='green')
plt.legend(loc='lower right')
plt.show()

Out[26]:
```



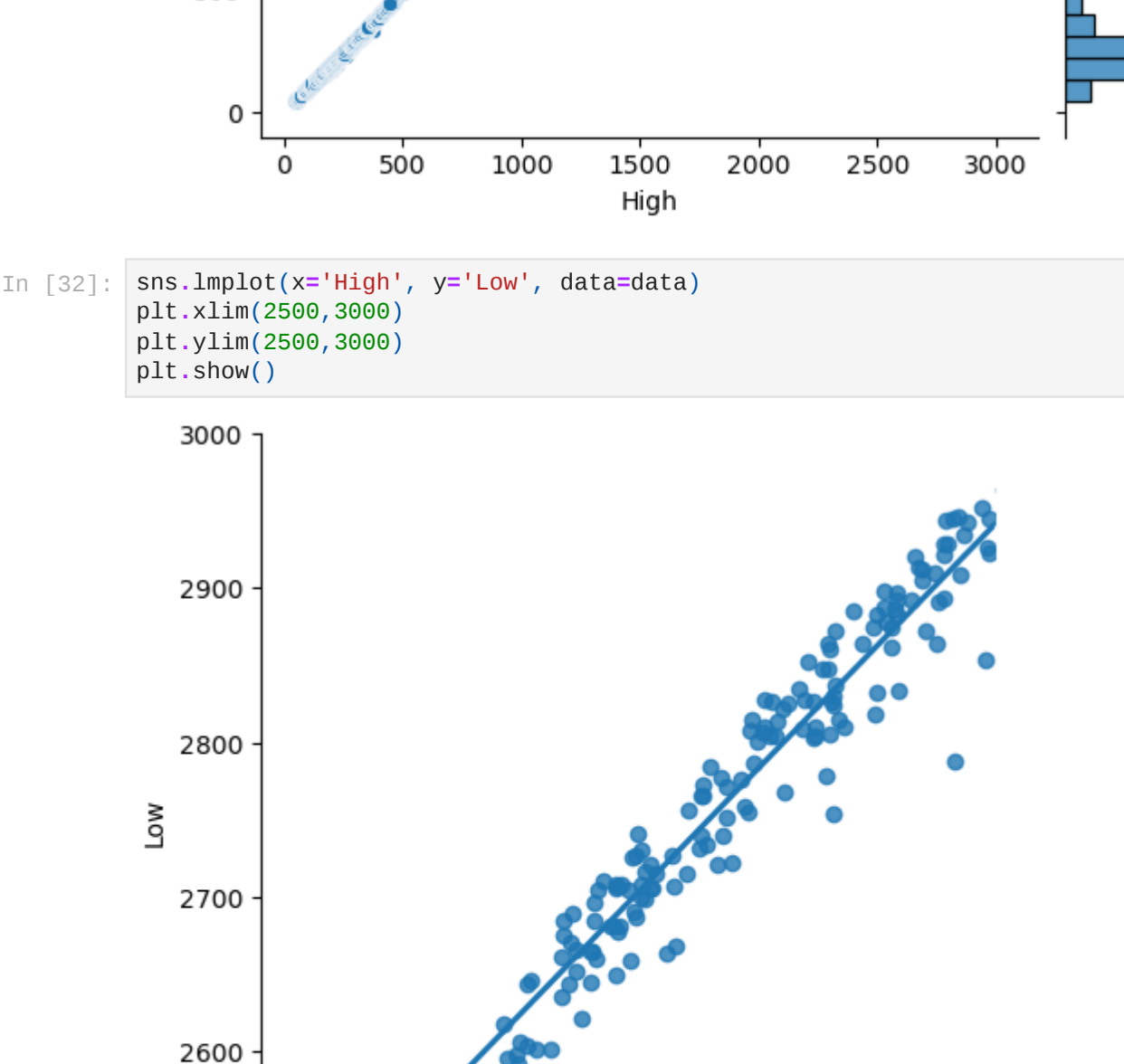
```
In [27]: plt.plot(data["Open"], label='Open',color='red')
plt.plot(data["Close"], label='Close',color='green')
plt.xlabel(1000,1000)
plt.ylabel(100,500)

Out[27]: (100.0, 500.0)
```



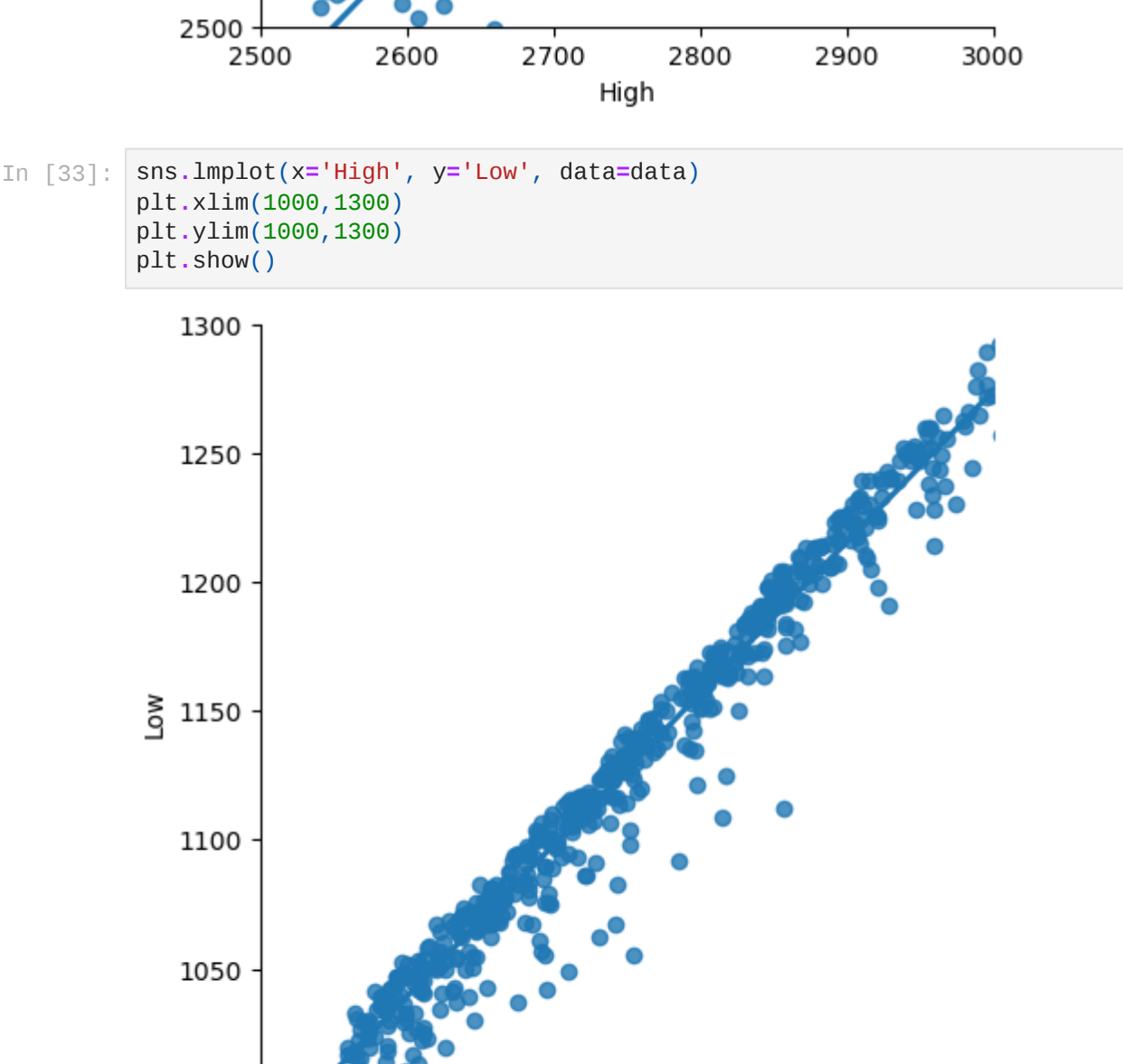
```
In [28]: plt.figure(figsize=(10,5))
plt.subplot(2,1,1)
plt.plot(data["Open"],color='red')
plt.subplot(2,1,2)
plt.plot(data["Close"],color='green')

Out[28]: [matplotlib.lines.Line2D at 0x18ae3ddbf9a]
```



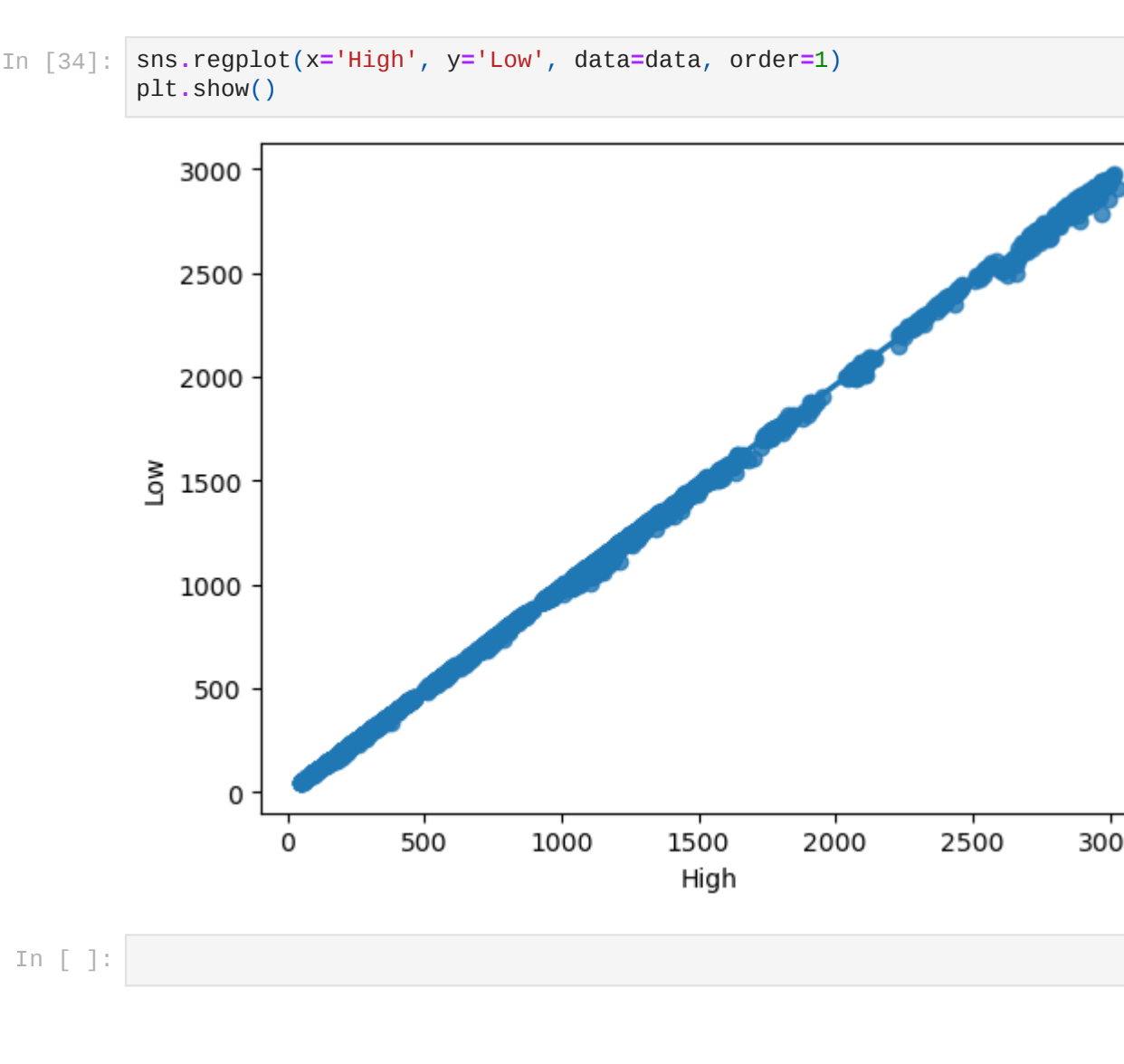
```
In [29]: plt.figure(figsize=(10,10))
plt.subplot(3,2,1)
plt.plot(data["Open"], color='red')
plt.xlabel("Count")
plt.subplot(3,2,2)
plt.plot(data["Close"],color='green')
plt.xlabel("Count")
plt.show()

Out[29]: Text(0.5, 'Count')
```



```
In [30]: import seaborn as sns

In [31]: sns.jointplot(x='High', y='Low', data=data)
plt.show()
```



```
In [32]: sns.lmplot(x='High', y='Low', data=data)
plt.xlabel(1000,1000)
plt.ylabel(1000,1000)
plt.show()
```



```
In [33]: sns.lmplot(x='High', y='Low', data=data)
plt.xlabel(1000,1000)
plt.ylabel(1000,1000)
plt.show()
```



```
In [34]: sns.regplot(x='High', y='Low', data=data, order=1)
plt.show()
```

