In [1]: In [2]:	<pre>import pandas as pd import numpy as np import matplotlib.pyplot as plt data=pd.read_csv('Downloads\GOOGL.csv')</pre>
In [3]: Out[3]:	0 2004-08-19 50.050049 52.082081 48.028027 50.220219 50.220219 44659096 1 2004-08-20 50.555557 54.594597 50.300301 54.209209 54.209209 22834343
	2 2004-08-23 55.430431 56.796799 54.579578 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 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 1317900
In [4]: Out[4]:	data.head() 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 50.300301 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]: Out[5]:	data.tail() 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 1774800
In [6]: Out[6]:	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 1317900 data.shape (4431, 7)
In [7]:	<pre>data.info() <class 'pandas.core.frame.dataframe'=""> RangeIndex: 4431 entries, 0 to 4430 Data columns (total 7 columns): # Column Non-Null Count Dtype</class></pre>
	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(5), int64(1), object(1)
In [8]: Out[8]: In [9]:	memory usage: 242.4+ KB data.index RangeIndex(start=0, stop=4431, step=1) data.columns
Out[9]: In [10]: Out[10]:	Index(['Date', 'Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'], dtype='object') Open
	mean 693.087345 699.735595 686.078751 693.097367 693.097367 6.444992e+06 std 645.118799 651.331215 638.579488 645.187806 7.690351e+06 min 49.644646 50.920921 48.028027 50.055054 50.055054 4.656000e+05 25% 248.558563 250.853355 245.813309 248.415916 248.415916 1.695600e+06 50% 434.924927 437.887878 432.687683 435.330322 435.330322 3.778418e+06
In [11]: Out[11]:	75% 1007.364990 1020.649994 997.274994 1007.790008 1007.790008 8.002390e+06 max 3025.000000 3030.929932 2977.979980 2996.770020 2996.770020 8.215117e+07 data.isna().any() Date False Open False
In [12]:	High False Low False Close False Adj Close False Volume False dtype: bool data.duplicated().sum()
Out[12]: In [13]: Out[13]:	data Date Open High Low Close Adj Close Volume
	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 1774800 4429 2022-03-23 2774.050049 2791.770020 2755.699951 2765.510010 2765.510010 1257700 4430 2022-03-24 2784.000000 2832.379883 2755.010010 2831.439941 1317900
In [14]: Out[14]:	<pre>data['Open'].plot(kind='line') .</pre>
	2500 - 2000 -
	1500 - 1000 - 500 -
In [15]:	data['Close'].plot(kind='line')
Out[15]:	<pre><axes:> 3000 - 2500 -</axes:></pre>
	1500 -
	1000 - 500 - 0 -
In [16]:	<pre>data['High'].plot(kind='line') plt.subplot(3,2,2) data['Low'].plot(kind='line') plt.subplot(3,2,3)</pre>
Out[16]:	
	2000 - 2000 3000 4000 - 1000 2000 3000 4000
In [17]:	data['Volume'].plot(kind='box')
Out[17]:	<pre><axes:> 1e7 8 -</axes:></pre>
	6 - 5 - 4 -
In [18]: Out[18]:	Volume data['Adj Close'].plot(kind='box')
	2500 - 2000 -
	1500 -
In [19]:	plt.subplot(3,2,1) data['Volume'].plot(kind='line')
<pre>In [19]: Out[19]:</pre>	<pre>data['Volume'].plot(kind='line') plt.subplot(3,2,2) data['Adj Close'].plot(kind='line') <axes:> le7</axes:></pre>
	5 - 0 - 1000 2000 3000 4000
In [20]: 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 50.300301 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 [21]:	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 × 7 columns plt.hist(data['Open'], width=100, color='red')
	plt.title('Open') plt.xlabel('Open') plt.ylabel('Count') plt.show() Open
	1750 - 1500 - 1250 -
	1000 - 750 - 500 -
In [22]:	250 - 0 500 1000 1500 2000 2500 3000 Open plt.hist(data['Close'], width=100, color='Green')
111 [22].	<pre>plt.title('Close') plt.xlabel('Close') plt.ylabel('Count') plt.show()</pre> Close
	2000 - 1750 - 1500 -
	750 - 500 -
	250 - 0 500 1000 1500 2000 2500 3000 Close
In [23]: Out[23]:	plt.plot(data['Close']) [<matplotlib.lines.line2d 0x18ae3fd74c0="" at="">] 3000 -</matplotlib.lines.line2d>
	2500 - 2000 - 1500 -
	1000 - 500 -
In [24]:	0 1000 2000 3000 4000 plt.plot(data['Close']) plt.xlim(500,1000) plt.ylim(100,500) (100.0, 500.0)
Out[24]:	500 450 - 400 -
	350 - 300 - 250 -
	200 -
In [25]: Out[25]:	<pre>plt.plot(data['Close']) plt.xlim(3000,4000) plt.ylim(700,3000) (700.0, 3000.0)</pre>
	2500 -
	2000 - 1500 -
	3000 3200 3400 3600 3800 4000
In [26]:	<pre>plt.plot(data['Open'], label='Open', color='red') plt.plot(data['Close'], label= 'Close', color='green') plt.legend(loc='lower right') plt.show()</pre> 3000 -
	2500 - 2000 - 1500 -
	1000 - 500 - Open
In [27]:	Description of the property of
Out[27]:	plt.ylim(700,800) (700.0, 800.0) 800 780 -
	760 -
	720 -
In [28]:	3000 3200 3400 3600 3800 4000 plt.figure(figsize= (10,5)) plt.subplot(2,2,1) plt.plot(data['Open'],color='red') plt.subplot(2,2,3) plt.plot(data['Close'],color='green')
Out[28]:	[<matplotlib.lines.line2d 0x18ae3ddbdf0="" at="">] 2000 -</matplotlib.lines.line2d>
	1000 - 1000 2000 3000 4000 3000 - 1000 2000 3000 4000
	2000 -
In [29]:	0 1000 2000 3000 4000 plt.figure(figsize= (10,10)) plt.subplot(3,1,1) plt.plot(data['Open'], color= 'red') plt.xlabel('Open') plt.ylabel('Count')
Out[29]:	plt.ylabel('Count') plt.subplot(3,1,2) plt.plot(data['Close'],color='green') plt.xlabel('Close') plt.ylabel('Count') Text(0, 0.5, 'Count')
	3000 - 2500 - 2000 - 1500 - 1000 -
	500 - 0 1000 2000 3000 4000 Open
	2500 - 2000 - 1500 - 1000 -
In [30]:	500 - 0 1000 2000 3000 4000 Close import seaborn as sns
In [30]:	<pre>import seaborn as sns sns.jointplot(x='High', y='Low', data=data) plt.show()</pre>
	3000 -
	2000 - 3 1500 -
	1000 - 500 - Experimental and the second sec
In [32]	sns.lmplot(x='High', y='Low', data=data) plt.xlim(2500, 3000)
. _[32]:	<pre>sns.lmplot(x='High', y='Low', data=data) plt.xlim(2500,3000) plt.ylim(2500,3000) plt.show()</pre> 3000
	2900 -
	2700 -
	2500 2500 2700 2800 2900 3000 High
In [33]:	
	1250 -
	1200 - 1150 - 1100 -
	1050
In [34]:	1000 1050 1100 1150 1200 1250 1300 High sns.regplot(x='High', y='Low', data=data, order=1) plt.show()
	3000 - 2500 - 2000 -
	<u>§</u> 1500 - 1000 -
	500 - 0 500 1000 1500 2000 2500 3000 High
In []:	