

Time Series Data Visualization

November 12, 2022

1 Write a Pandas program to create a line plot of the historical stock prices of Alphabet Inc. between two specific dates

```
[2]: import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("alphabet_stock_data.csv")
df
```

```
[2]:
```

	Date	Open	High	Low	Close \
0	2020-04-01	1122.000000	1129.689941	1097.449951	1105.619995
1	2020-04-02	1098.260010	1126.859985	1096.400024	1120.839966
2	2020-04-03	1119.015015	1123.540039	1079.810059	1097.880005
3	2020-04-06	1138.000000	1194.660034	1130.939941	1186.920044
4	2020-04-07	1221.000000	1225.000000	1182.229980	1186.510010
..
122	2020-09-24	1411.030029	1443.708984	1409.849976	1428.290039
123	2020-09-25	1432.630005	1450.000000	1413.339966	1444.959961
124	2020-09-28	1474.209961	1476.800049	1449.301025	1464.520020
125	2020-09-29	1470.390015	1476.662964	1458.805054	1469.329956
126	2020-09-30	1466.800049	1489.750000	1459.880005	1469.599976

	Adj Close	Volume
0	1105.619995	2343100
1	1120.839966	1964900
2	1097.880005	2313400
3	1186.920044	2664700
4	1186.510010	2387300
..
122	1428.290039	1450200
123	1444.959961	1323000
124	1464.520020	2007900
125	1469.329956	978200
126	1469.599976	1700600

[127 rows x 7 columns]

```
[3]: start_date = pd.to_datetime('2020-6-1')
end_date = pd.to_datetime('2020-09-30')
df['Date'] = pd.to_datetime(df['Date'])
new_df = (df['Date']>= start_date) & (df['Date']<= end_date) #new_df is a
↳boolean df
new_df
```

```
[3]: 0      False
1      False
2      False
3      False
4      False
...
122     True
123     True
124     True
125     True
126     True
Name: Date, Length: 127, dtype: bool
```

```
[4]: df1 = df.loc[new_df]
df1
```

```
[4]:
```

	Date	Open	High	Low	Close \
41	2020-06-01	1418.390015	1437.959961	1418.000000	1431.819946
42	2020-06-02	1430.550049	1439.609985	1418.829956	1439.219971
43	2020-06-03	1438.300049	1446.552002	1429.776978	1436.380005
44	2020-06-04	1430.400024	1438.959961	1404.729980	1412.180054
45	2020-06-05	1413.170044	1445.050049	1406.000000	1438.390015
..
122	2020-09-24	1411.030029	1443.708984	1409.849976	1428.290039
123	2020-09-25	1432.630005	1450.000000	1413.339966	1444.959961
124	2020-09-28	1474.209961	1476.800049	1449.301025	1464.520020
125	2020-09-29	1470.390015	1476.662964	1458.805054	1469.329956
126	2020-09-30	1466.800049	1489.750000	1459.880005	1469.599976

	Adj Close	Volume
41	1431.819946	1217100
42	1439.219971	1278100
43	1436.380005	1256200
44	1412.180054	1484300
45	1438.390015	1734900
..
122	1428.290039	1450200
123	1444.959961	1323000
124	1464.520020	2007900
125	1469.329956	978200

126 1469.599976 1700600

[86 rows x 7 columns]

```
[5]: df2 = df1.set_index('Date') #for time series, date should be the index
df2
```

```
[5]:
```

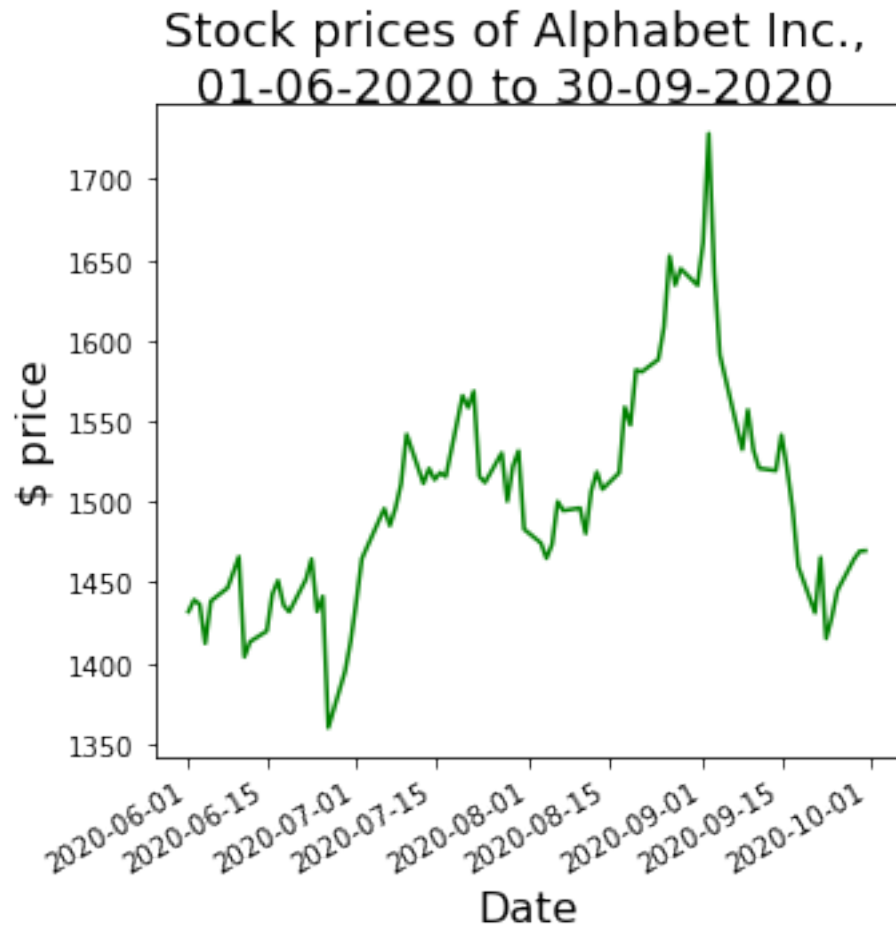
	Open	High	Low	Close	Adj Close \
Date					
2020-06-01	1418.390015	1437.959961	1418.000000	1431.819946	1431.819946
2020-06-02	1430.550049	1439.609985	1418.829956	1439.219971	1439.219971
2020-06-03	1438.300049	1446.552002	1429.776978	1436.380005	1436.380005
2020-06-04	1430.400024	1438.959961	1404.729980	1412.180054	1412.180054
2020-06-05	1413.170044	1445.050049	1406.000000	1438.390015	1438.390015
...
2020-09-24	1411.030029	1443.708984	1409.849976	1428.290039	1428.290039
2020-09-25	1432.630005	1450.000000	1413.339966	1444.959961	1444.959961
2020-09-28	1474.209961	1476.800049	1449.301025	1464.520020	1464.520020
2020-09-29	1470.390015	1476.662964	1458.805054	1469.329956	1469.329956
2020-09-30	1466.800049	1489.750000	1459.880005	1469.599976	1469.599976

	Volume
Date	
2020-06-01	1217100
2020-06-02	1278100
2020-06-03	1256200
2020-06-04	1484300
2020-06-05	1734900
...	...
2020-09-24	1450200
2020-09-25	1323000
2020-09-28	2007900
2020-09-29	978200
2020-09-30	1700600

[86 rows x 6 columns]

```
[6]: plt.figure(figsize=(5,5))
plt.suptitle('Stock prices of Alphabet Inc.,\n01-06-2020 to 30-09-2020', \
            fontsize=18, color='black')
plt.xlabel("Date",fontsize=16, color='black')
plt.ylabel("$ price", fontsize=16, color='black')

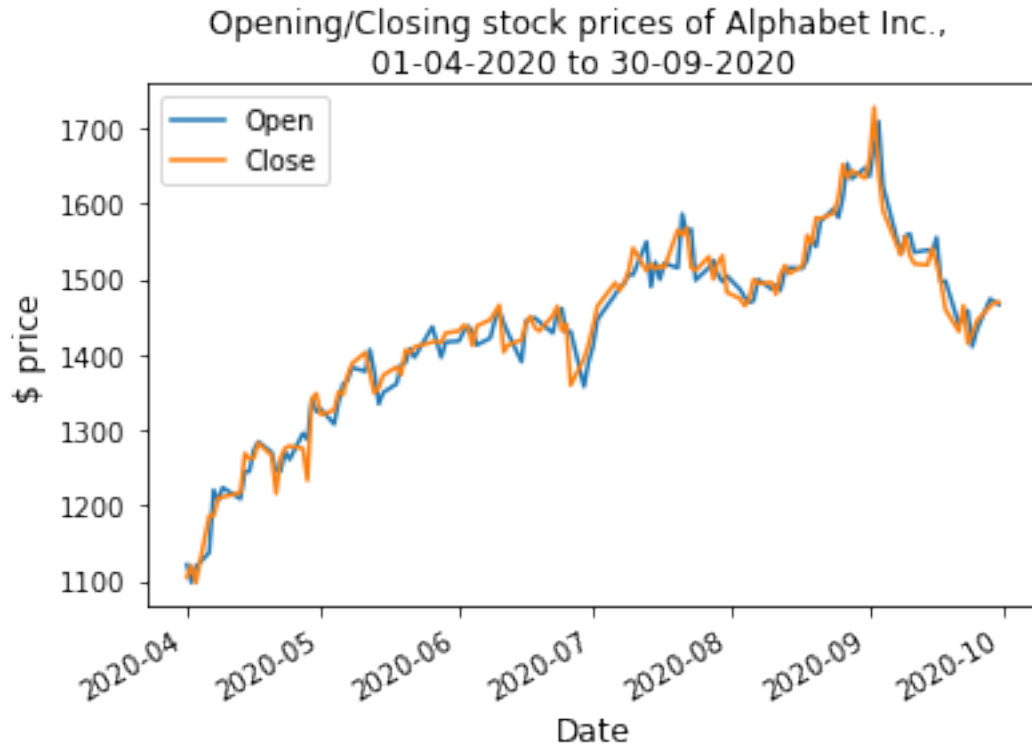
df2['Close'].plot(color='green');
plt.show()
```



2 Write a Pandas program to create a line plot of the opening, closing stock prices of Alphabet Inc.

```
[7]: df['Date'] = pd.to_datetime(df['Date'])
plt.figure(figsize=(10,10))
df.plot(x='Date', y=['Open', 'Close']);
plt.suptitle('Opening/Closing stock prices of Alphabet Inc.,\n 01-04-2020 to 30-09-2020', fontsize=12, color='black')
plt.xlabel("Date",fontsize=12, color='black')
plt.ylabel("$ price", fontsize=12, color='black')
plt.show()
```

<Figure size 720x720 with 0 Axes>



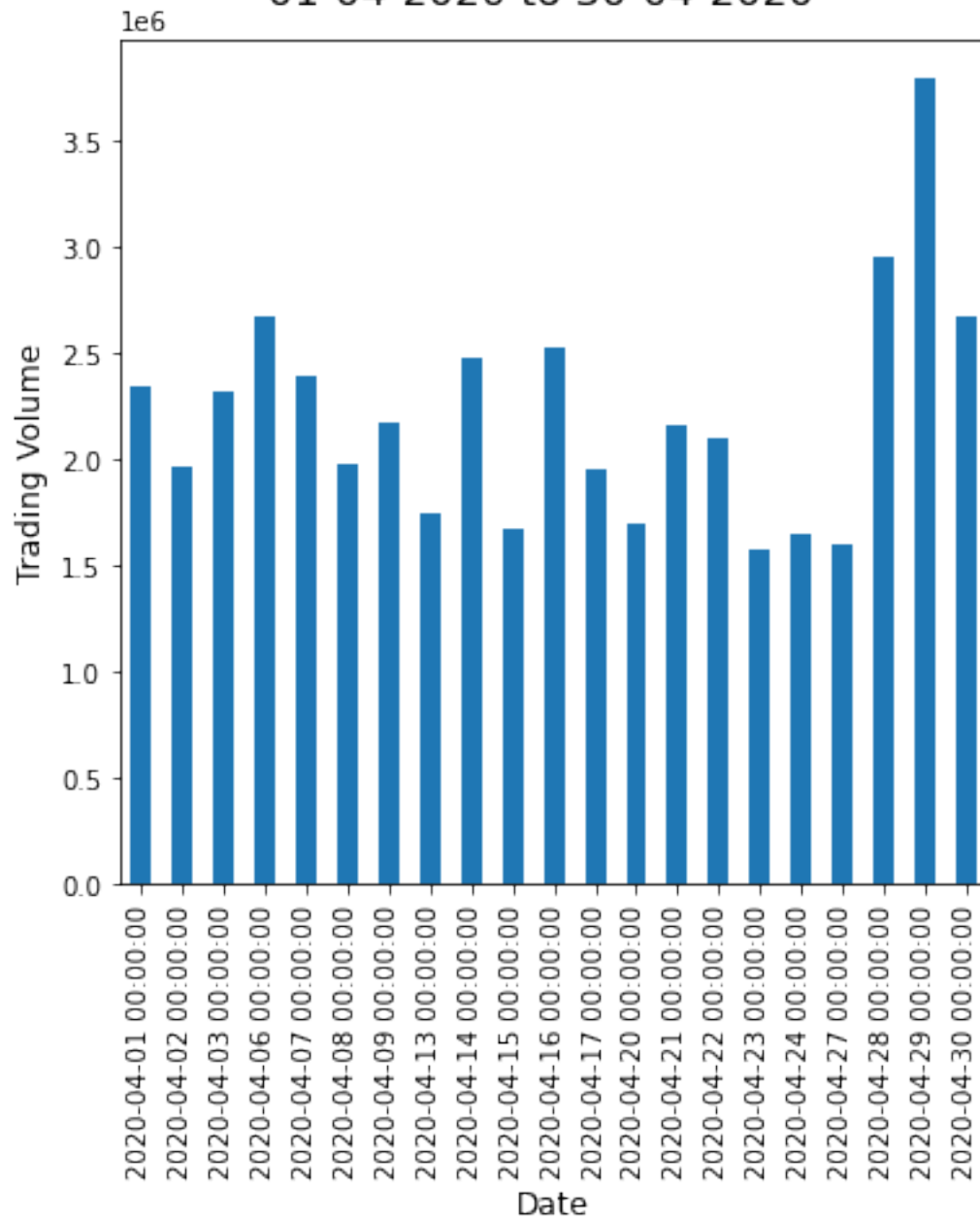
3 Write a Pandas program to create a bar plot of the trading volume of Alphabet Inc. stock between two specific dates.

```
[8]: df = pd.read_csv("alphabet_stock_data.csv")
start_date = pd.to_datetime('2020-4-1')
end_date = pd.to_datetime('2020-4-30')
df['Date'] = pd.to_datetime(df['Date'])
new_df = (df['Date'] >= start_date) & (df['Date'] <= end_date)

df1 = df.loc[new_df]
df2 = df1.set_index('Date')

plt.figure(figsize=(6,6))
plt.suptitle('Trading Volume of Alphabet Inc. stock,\n01-04-2020 to 30-04-2020',
            ↪fontsize=16, color='black')
plt.xlabel("Date",fontsize=12, color='black')
plt.ylabel("Trading Volume", fontsize=12, color='black')
df2['Volume'].plot(kind='bar');
plt.show()
```

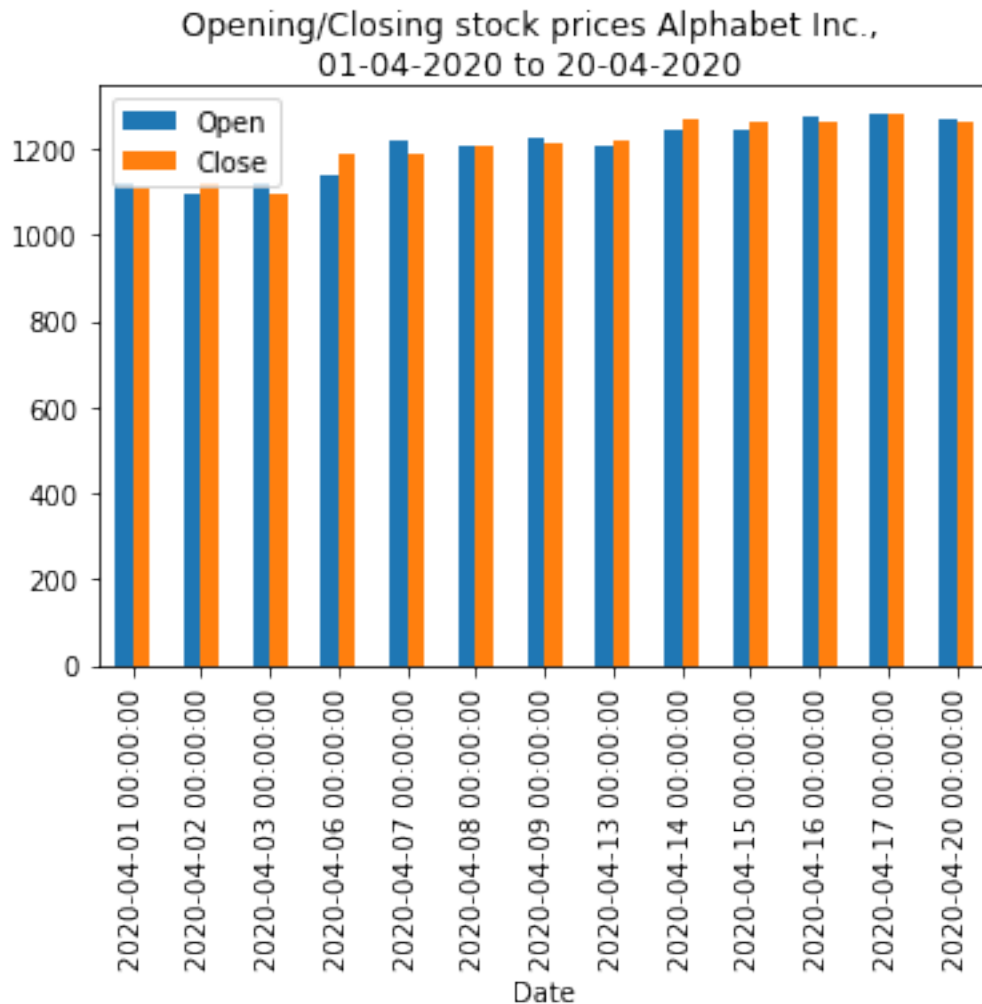
Trading Volume of Alphabet Inc. stock,
01-04-2020 to 30-04-2020



4 Write a Pandas program to create a bar plot of opening, closing stock prices of Alphabet Inc. between two specific dates.

```
[9]: import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("alphabet_stock_data.csv")
start_date = pd.to_datetime('2020-4-1')
end_date = pd.to_datetime('2020-4-20')
df['Date'] = pd.to_datetime(df['Date'])
new_df = (df['Date'] >= start_date) & (df['Date'] <= end_date)
df1 = df.loc[new_df]
df2 = df1[['Date', 'Open', 'Close']]
df3 = df2.set_index('Date')
plt.figure(figsize=(20,20))
df3.plot.bar();
plt.suptitle('Opening/Closing stock prices Alphabet Inc.,\n01-04-2020 to 20-04-2020', fontsize=12, color='black')
plt.show()
```

<Figure size 1440x1440 with 0 Axes>



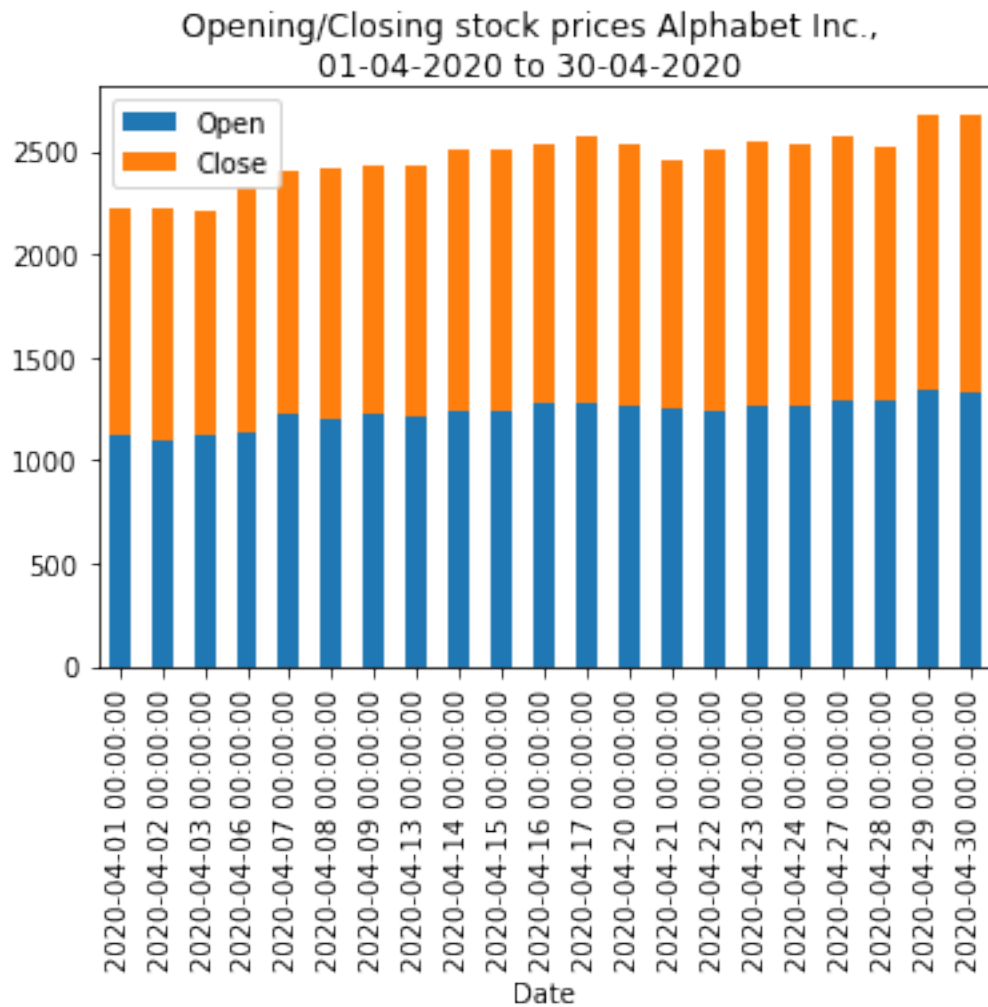
- 5 Write a Pandas program to create a stacked bar plot of opening, closing stock prices of Alphabet Inc. between two specific dates.

```
[10]: import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("alphabet_stock_data.csv")
start_date = pd.to_datetime('2020-4-1')
end_date = pd.to_datetime('2020-4-30')
df['Date'] = pd.to_datetime(df['Date'])
new_df = (df['Date'] >= start_date) & (df['Date'] <= end_date)
df1 = df.loc[new_df]
df2 = df1[['Date', 'Open', 'Close']]
df3 = df2.set_index('Date')
plt.figure(figsize=(20,20))
```



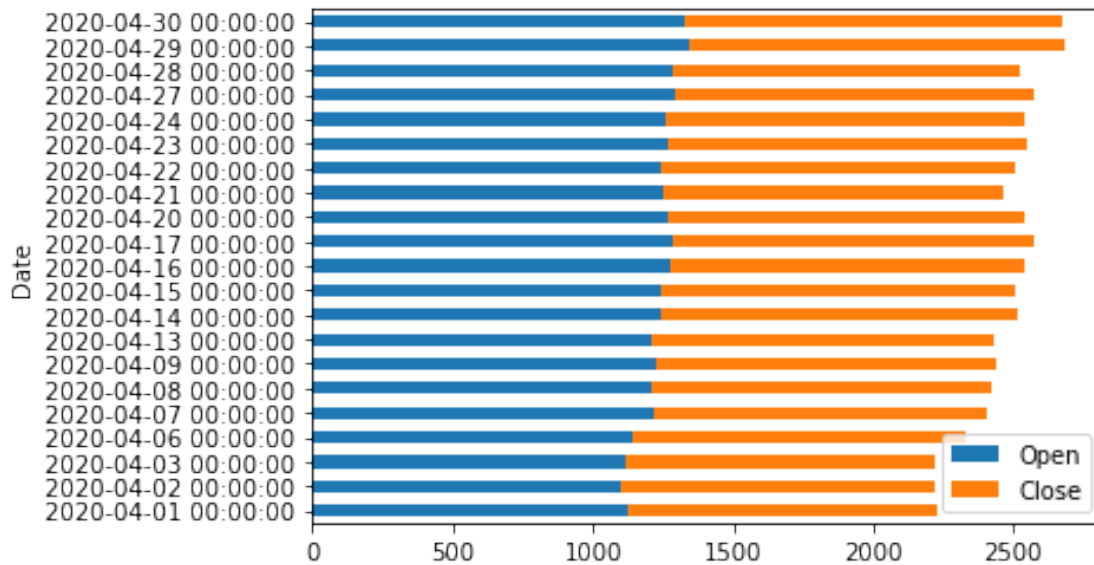
```
df3.plot.bar(stacked=True);
plt.suptitle('Opening/Closing stock prices Alphabet Inc.,\n01-04-2020 to 30-04-2020', fontsize=12, color='black')
plt.show()
```

<Figure size 1440x1440 with 0 Axes>



6 Create a horizontal stacked bar plot of opening, closing stock prices of Alphabet Inc.

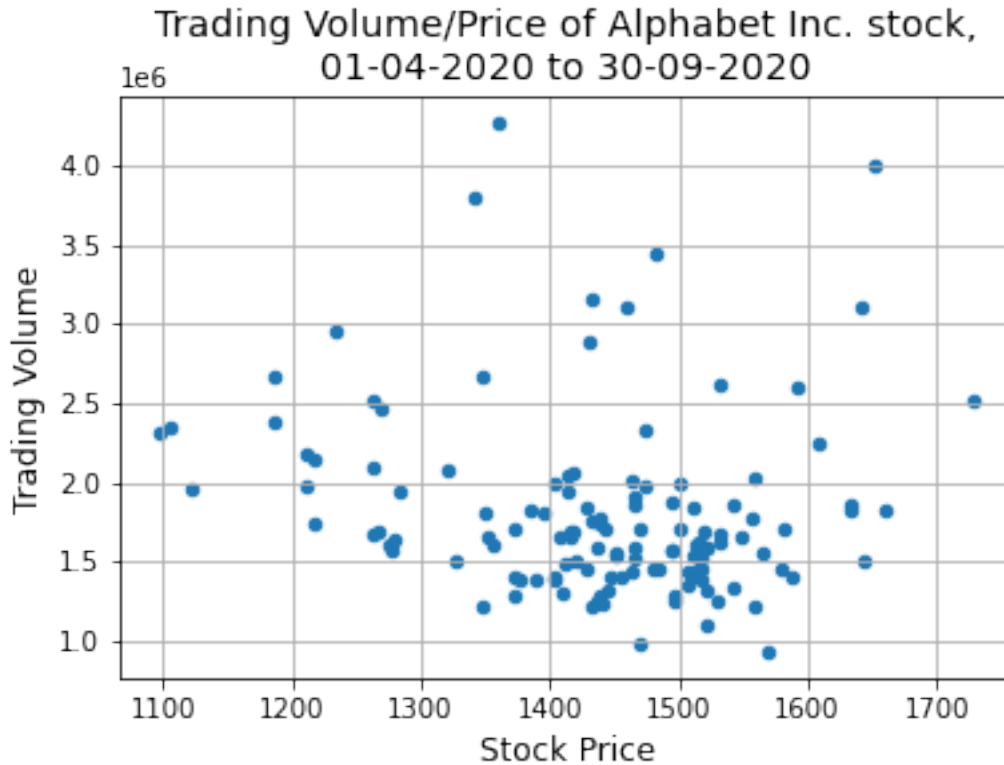
```
[11]: df3.plot.barh(stacked=True);
```



- 7 Write a Pandas program to create a scatter plot of the trading volume/stock prices of Alphabet Inc. stock between two specific dates.

```
[12]: df = pd.read_csv("alphabet_stock_data.csv")
start_date = pd.to_datetime('2020-4-1')
end_date = pd.to_datetime('2020-9-30')
df['Date'] = pd.to_datetime(df['Date'])
new_df = (df['Date'] >= start_date) & (df['Date'] <= end_date)
df1 = df.loc[new_df]
df2 = df1.set_index('Date')
#x= ['Close']; y = ['Volume']
plt.figure(figsize=[15,10])
df2.plot.scatter('Close', 'Volume', s=20);
plt.grid(True)
plt.title('Trading Volume/Price of Alphabet Inc. stock,\n01-04-2020 to 30-09-2020', fontsize=14, color='black')
plt.xlabel("Stock Price", fontsize=12, color='black')
plt.ylabel("Trading Volume", fontsize=12, color='black')
plt.show()
```

<Figure size 1080x720 with 0 Axes>



8 Write a Pandas program to create a plot to visualize daily percentage returns of Alphabet Inc. stock price between two specific dates

```
[13]: df = pd.read_csv("alphabet_stock_data.csv")
start_date = pd.to_datetime('2020-4-1')
end_date = pd.to_datetime('2020-9-30')
df['Date'] = pd.to_datetime(df['Date'])
new_df = (df['Date'] >= start_date) & (df['Date'] <= end_date)
df1 = df.loc[new_df]
df2 = df1[['Date', 'Adj Close']]
df3 = df2.set_index('Date')
print(df3)
daily_changes = df3.pct_change(periods=1)
print(daily_changes)
daily_changes['Adj Close'].
    →plot(figsize=(10,7),legend=True,linestyle='--',marker='o')
plt.suptitle('Daily % return of Alphabet Inc. stock price,\n01-04-2020 to
    →30-09-2020', fontsize=12, color='black')
plt.grid(True)
```

```
plt.show()
```

Date	Adj Close
2020-04-01	1105.619995
2020-04-02	1120.839966
2020-04-03	1097.880005
2020-04-06	1186.920044
2020-04-07	1186.510010
...	...
2020-09-24	1428.290039
2020-09-25	1444.959961
2020-09-28	1464.520020
2020-09-29	1469.329956
2020-09-30	1469.599976

[127 rows x 1 columns]

Date	Adj Close
2020-04-01	NaN
2020-04-02	0.013766
2020-04-03	-0.020485
2020-04-06	0.081102
2020-04-07	-0.000345
...	...
2020-09-24	0.009242
2020-09-25	0.011671
2020-09-28	0.013537
2020-09-29	0.003284
2020-09-30	0.000184

[127 rows x 1 columns]

Daily % return of Alphabet Inc. stock price,
01-04-2020 to 30-09-2020

