

```
In [27]: import pandas as pd
import os
import numpy as np
import matplotlib.pyplot as plt
```

```
In [28]: df = pd.read_csv(r"C:\Users\khush\Documents\BigData\Projects\Project3\Athena\results.csv")
df
```

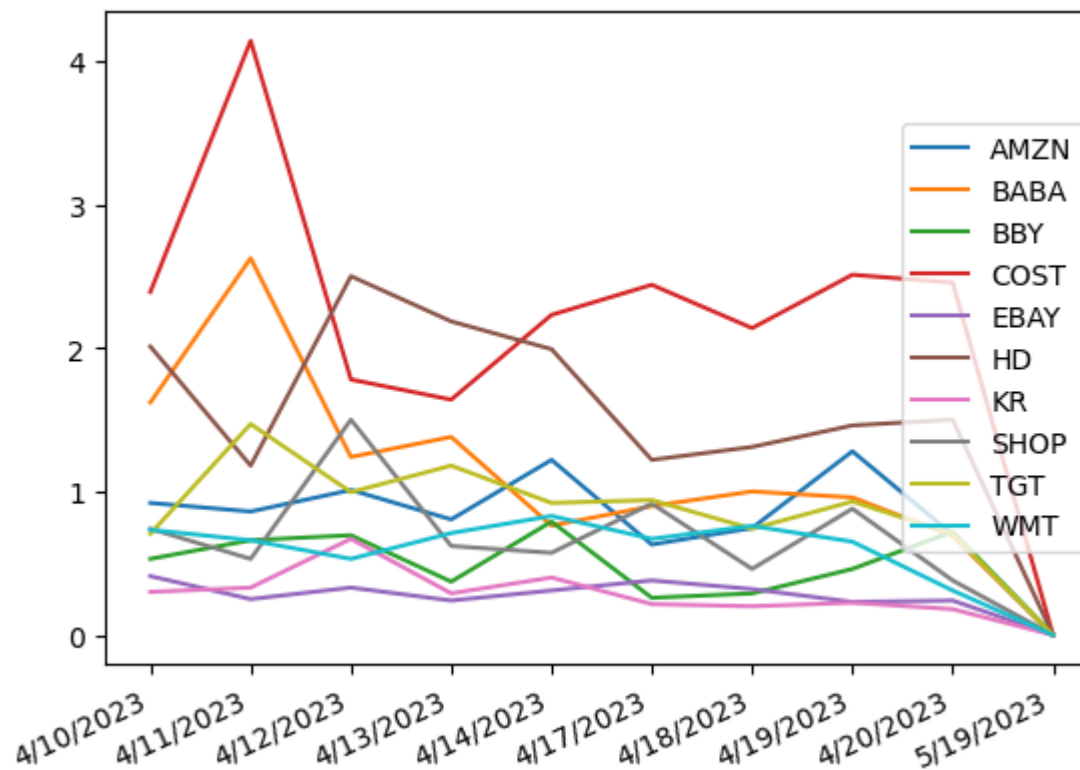
```
Out[28]:
```

	Company_Name	Date	Highest_Volatility	Average_Volatility	Lowest_Volatility
0	AMZN	4/10/2023	0.919998	0.222208	0.095001
1	AMZN	4/11/2023	0.860001	0.221386	0.079407
2	AMZN	4/12/2023	1.010002	0.276850	0.074997
3	AMZN	4/13/2023	0.805000	0.243291	0.080002
4	AMZN	4/14/2023	1.220001	0.282877	0.119896
...
95	WMT	4/17/2023	0.669998	0.144602	0.039993
96	WMT	4/18/2023	0.759995	0.158071	0.058899
97	WMT	4/19/2023	0.650009	0.140120	0.040009
98	WMT	4/20/2023	0.309998	0.137618	0.039993
99	WMT	5/19/2023	0.000000	0.000000	0.000000

100 rows × 5 columns

```
In [29]: #Graph the maximum volatility trend per company (A single Line Chart: Each line refers to a company)
high_vol = df.drop(columns=["Average_Volatility","Lowest_Volatility"]).pivot(index="Date",columns="Company_Na
high_vol
tickers = high_vol.columns

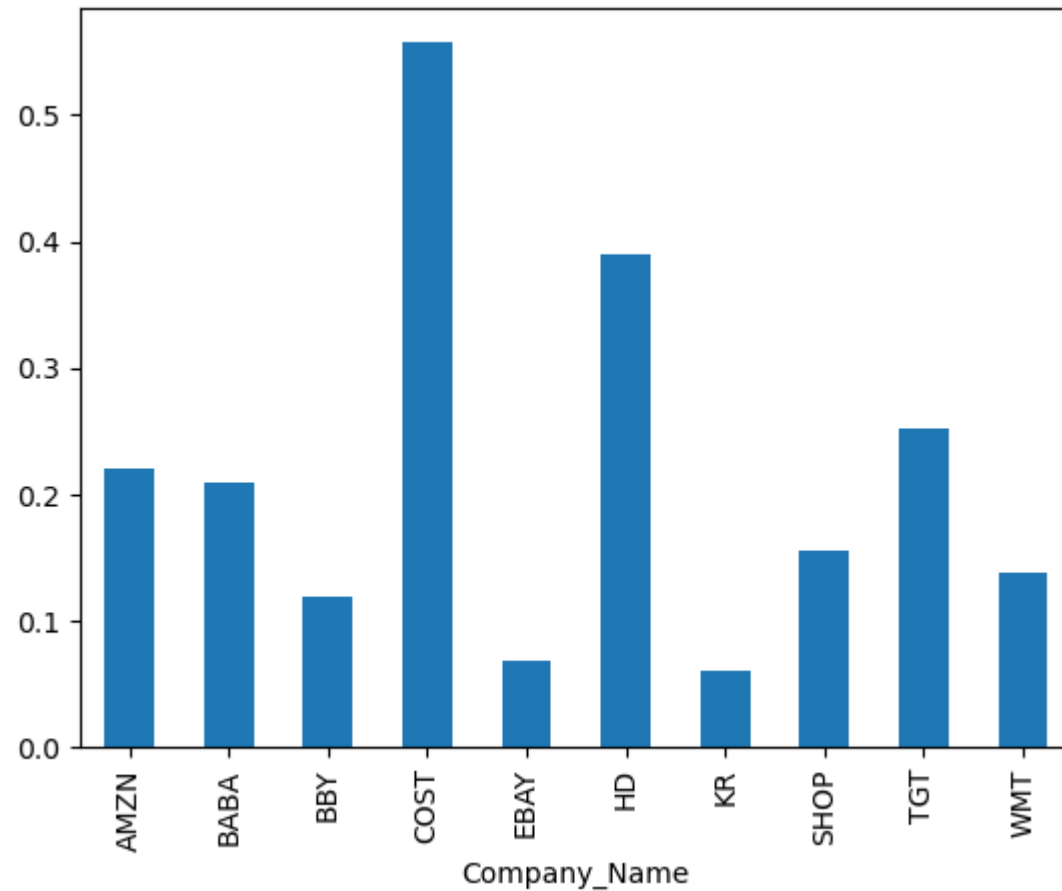
for ticker in tickers:
    plt.plot(high_vol.index,high_vol[ticker],label=ticker)
plt.gcf().autofmt_xdate(rotation=25)
plt.legend(loc="right")
plt.show()
```



```
In [ ]: #According to the plot, costco looks to be the most volatile.
```

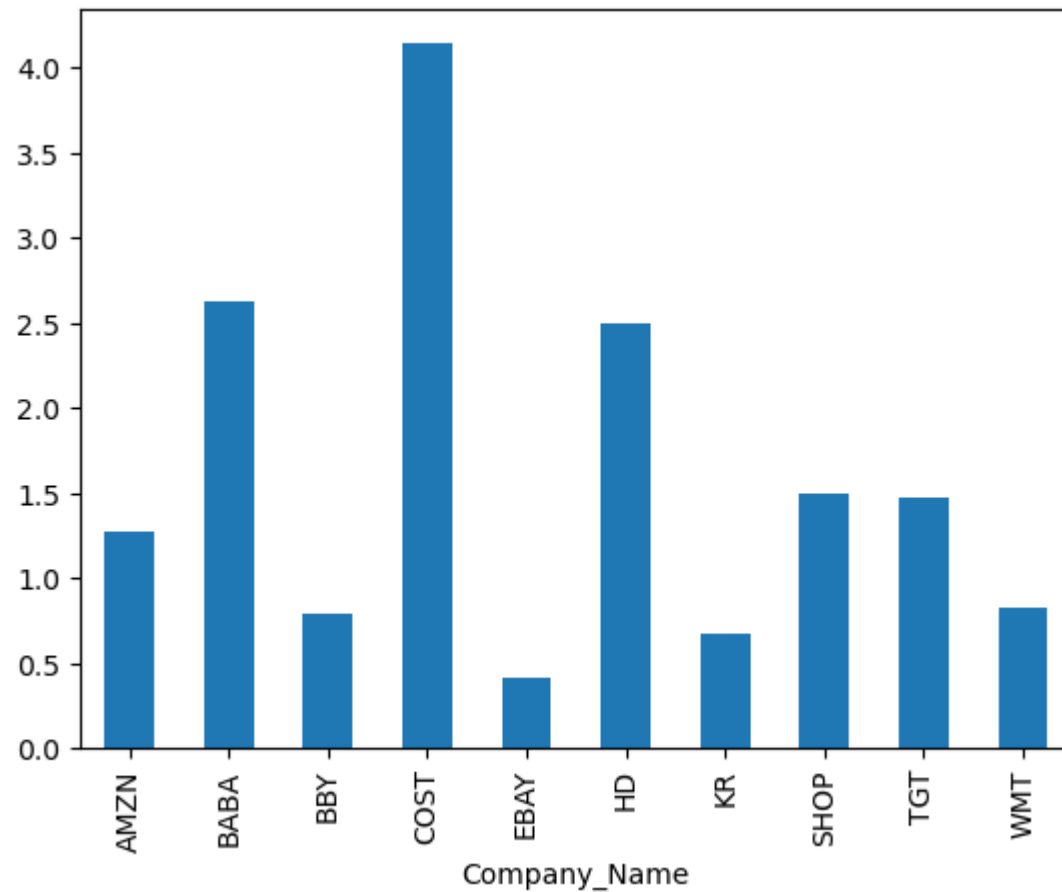
```
In [30]: #Graph the daily average volatility per company(A Grouped Bar Chart: Each group refers to a company and the k  
avg_vol = df.groupby('Company_Name')['Average_Volatility'].mean()  
avg_vol.plot.bar()
```

```
Out[30]: <AxesSubplot:xlabel='Company_Name'>
```



```
In [35]: # Lowest Daily Maximum Volatility oer company  
highest_vol = df.groupby('Company_Name')['Highest_Volatility'].agg('max')  
highest_vol.plot.bar()
```

```
Out[35]: <AxesSubplot:xlabel='Company_Name'>
```



In [43]: *#Graph the maximum volatility trend of Amazon*

```
high_vol = df.drop(columns=["Average_Volatility","Lowest_Volatility"]).pivot(index="Date",columns="Company_Na  
high_vol  
tickers = high_vol.columns  
  
for AMZN in tickers:  
    plt.plot(high_vol.index,high_vol[ticker],label=ticker)  
plt.gcf().autofmt_xdate(rotation=25)  
plt.show()
```

