In [1]:
```
!pip install pandas
!pip install matplotlib
!pip install seaborn
```

```
Requirement already satisfied: pandas in /Users/anthonynguyen/anaconda3/lib/py
thon3.10/site-packages (2.0.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /Users/anthonynguyen/
anaconda3/lib/python3.10/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /Users/anthonynguyen/anaconda3/
lib/python3.10/site-packages (from pandas) (2022.7)
Requirement already satisfied: tzdata>=2022.1 in /Users/anthonynguyen/anaconda
3/lib/python3.10/site-packages (from pandas) (2023.3)
Requirement already satisfied: numpy>=1.21.0 in /Users/anthonynguyen/anaconda
3/lib/python3.10/site-packages (from pandas) (1.24.3)
Requirement already satisfied: six>=1.5 in /Users/anthonynguyen/anaconda3/lib/
python3.10/site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Requirement already satisfied: matplotlib in /Users/anthonynguyen/anaconda3/li
b/python3.10/site-packages (3.7.1)
Requirement already satisfied: fonttools>=4.22.0 in /Users/anthonynguyen/anaco
nda3/lib/python3.10/site-packages (from matplotlib) (4.39.3)
Requirement already satisfied: numpy>=1.20 in /Users/anthonynguyen/anaconda3/l
ib/python3.10/site-packages (from matplotlib) (1.24.3)
Requirement already satisfied: packaging>=20.0 in /Users/anthonynguyen/anacond
a3/lib/python3.10/site-packages (from matplotlib) (23.0)
Requirement already satisfied: python-dateutil>=2.7 in /Users/anthonynguyen/an
aconda3/lib/python3.10/site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in /Users/anthonynguyen/anaco
nda3/lib/python3.10/site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: pillow>=6.2.0 in /Users/anthonynguyen/anaconda
3/lib/python3.10/site-packages (from matplotlib) (9.4.0)
Requirement already satisfied: cycler>=0.10 in /Users/anthonynguyen/anaconda3/
lib/python3.10/site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: contourpy>=1.0.1 in /Users/anthonynguyen/anacon
da3/lib/python3.10/site-packages (from matplotlib) (1.0.7)
Requirement already satisfied: pyparsing>=2.3.1 in /Users/anthonynguyen/anacon
da3/lib/python3.10/site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: six>=1.5 in /Users/anthonynguyen/anaconda3/lib/
python3.10/site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Requirement already satisfied: seaborn in /Users/anthonynguyen/anaconda3/lib/p
ython3.10/site-packages (0.12.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in /Users/anthonynguye
n/anaconda3/lib/python3.10/site-packages (from seaborn) (3.7.1)
Requirement already satisfied: pandas>=0.25 in /Users/anthonynguyen/anaconda3/
lib/python3.10/site-packages (from seaborn) (2.0.1)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in /Users/anthonynguyen/an
aconda3/lib/python3.10/site-packages (from seaborn) (1.24.3)
Requirement already satisfied: python-dateutil>=2.7 in /Users/anthonynguyen/an
aconda3/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
(2.8.2)
Requirement already satisfied: fonttools>=4.22.0 in /Users/anthonynguyen/anaco
nda3/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (4.3
9.3)
Requirement already satisfied: pyparsing>=2.3.1 in /Users/anthonynguyen/anacon
da3/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (3.0.
9)
Requirement already satisfied: cycler>=0.10 in /Users/anthonynguyen/anaconda3/
lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: packaging>=20.0 in /Users/anthonynguyen/anacond
a3/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (23.0)
Requirement already satisfied: pillow>=6.2.0 in /Users/anthonynguyen/anaconda
3/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (9.4.0)
Requirement already satisfied: contourpy>=1.0.1 in /Users/anthonynguyen/anacon
da3/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.0.
7)
```

```
Requirement already satisfied: kiwisolver>=1.0.1 in /Users/anthonynguyen/anaco
nda3/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.
4.4)
Requirement already satisfied: tzdata>=2022.1 in /Users/anthonynguyen/anaconda
3/lib/python3.10/site-packages (from pandas>=0.25->seaborn) (2023.3)
Requirement already satisfied: pytz>=2020.1 in /Users/anthonynguyen/anaconda3/
lib/python3.10/site-packages (from pandas>=0.25->seaborn) (2022.7)
Requirement already satisfied: six>=1.5 in /Users/anthonynguyen/anaconda3/lib/
python3.10/site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.1->
seaborn) (1.16.0)
```

In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

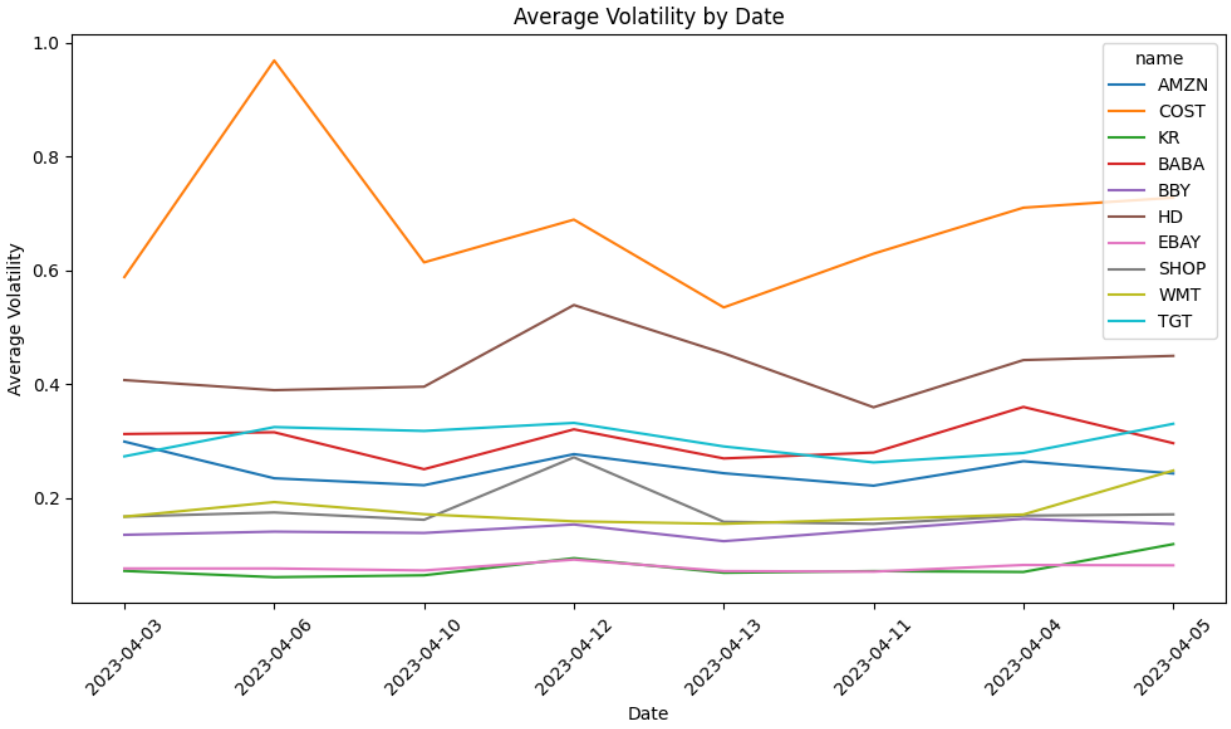
In [3]:
```
project3_df = pd.read_csv('results.csv')
```

In [4]:
```
project3_df.head()
```

Out[4]:

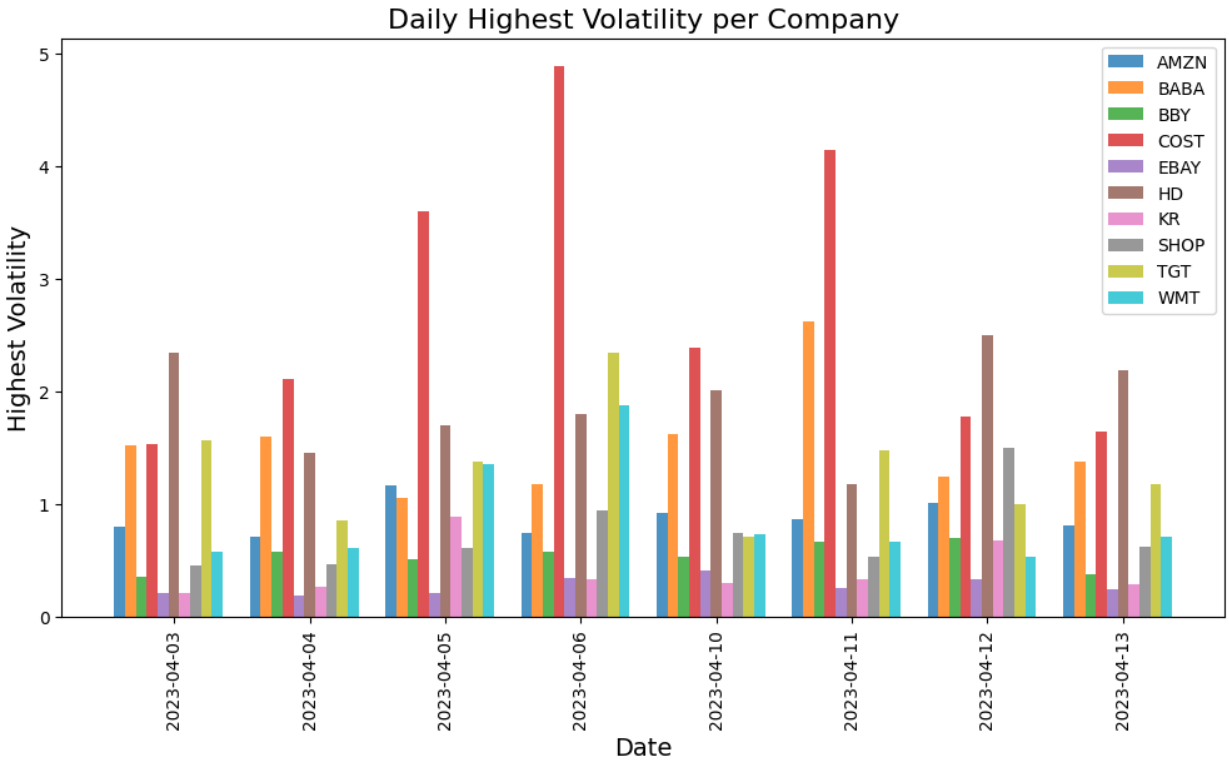
	name	date	average_volatility	highest_volatility	lowest_volatility
0	AMZN	2023-04-03	0.298844	0.800003	0.087097
1	COST	2023-04-06	0.968865	4.890015	0.269989
2	KR	2023-04-10	0.063856	0.299999	0.014999
3	BABA	2023-04-12	0.320448	1.239998	0.095299
4	BABA	2023-04-13	0.269298	1.379898	0.070000

In [5]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Visualization 1: Line plot of average volatility by date
plt.figure(figsize=(10, 6))
sns.lineplot(data=project3_df, x='date', y='average_volatility', hue='name')
plt.title('Average Volatility by Date')
plt.xlabel('Date')
plt.ylabel('Average Volatility')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Costco is the most volatile !!

In [6]:
```
# Group the data by company name and date, and calculate the maximum value of h
grouped_df = project3_df.groupby(['name', 'date'])['highest_volatility'].max().

# Create a list of unique company names
companies = grouped_df['name'].unique()

# Set the size of the plot
plt.figure(figsize=(12, 6))

# Get the unique dates in the data
dates = grouped_df['date'].unique()

# Set the width of each bar
bar_width = 0.8 / len(companies)

# Iterate through each company and plot the daily highest volatility as a bar
for i, company in enumerate(companies):
    company_data = grouped_df[grouped_df['name'] == company]
    # Calculate the x positions for each bar of the company
    x_pos = [j + i * bar_width for j in range(len(dates))]
    plt.bar(x_pos, company_data['highest_volatility'], width=bar_width, label=c

# Set the title, x-axis label, y-axis label, and legend
plt.title('Daily Highest Volatility per Company', fontsize=16)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Highest Volatility', fontsize=14)
plt.legend()

# Set the x-axis ticks and labels to the unique dates
plt.xticks([i + bar_width * len(companies) / 2 for i in range(len(dates))], dat

# Show the plot
plt.show()
```



Yes, the finding of this graph support my previous conclusion from the first graph. Costco is the most volatile.