```
import pandas as pd
import matplotlib.pyplot as plt # Importing what needed to imports

!pip install openpyxl # For reading an excel file the command openpyxl
is used
```

```
Collecting openpyxl
  Downloading openpyxl-3.1.5-py2.py3-none-any.whl.metadata (2.5 kB)
Collecting et-xmlfile (from openpyxl)
  Downloading et_xmlfile-2.0.0-py3-none-any.whl.metadata (2.7 kB)
Downloading openpyxl-3.1.5-py2.py3-none-any.whl (250 kB)
Downloading et_xmlfile-2.0.0-py3-none-any.whl (18 kB)
Installing collected packages: et-xmlfile, openpyxl
Successfully installed et-xmlfile-2.0.0 openpyxl-3.1.5


[notice] A new release of pip is available: 24.2 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
df =
pd.read_excel(r"C:/Users/ASUS/Downloads/Financial_Summary_Microsoft_Te
sla_Apple_2021_2023.xlsx")
df.head() # To show the file i have made using excel
```

```
   Year  Microsoft_Revenue  Microsoft_Net_Income  Microsoft_Assets  \
0  2021             168088                 61271            333779
1  2022             198270                 72738            364840
2  2023             211915                 72361            411995

   Microsoft_Liabilities  Microsoft_Cash_Flow  Tesla_Revenue  \
0                 191791                76737          53823
1                 198298                89004          81462
2                 198298                89004          96773

   Tesla_Net_Income  Tesla_Assets  Tesla_Liabilities  Tesla_Cash_Flow
\
0              5519         62131              30548            11497

1             12556         82338              35466            14724

2             12556         93659              40537            13256


   Apple_Revenue  Apple_Net_Income  Apple_Assets  Apple_Liabilities  \
0         274515             57411        351002             287912
1         365817             94680        351002             287912
2         394328             99803        352755             302083

   Apple_Cash_Flow
0           104038
```

```
1           122151
2           122151
```

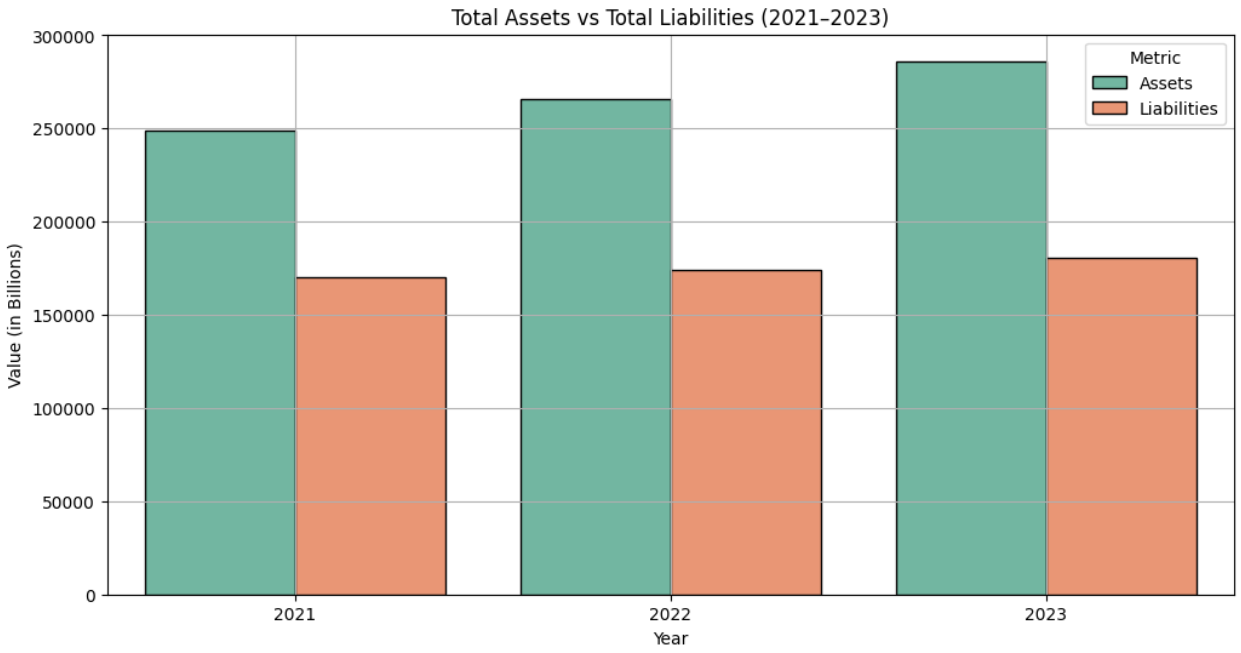
```
print(df.columns.tolist())
```

```
['Year', 'Microsoft_Revenue', 'Microsoft_Net_Income',
'Microsoft_Assets', 'Microsoft_Liabilities', 'Microsoft_Cash_Flow',
'Tesla_Revenue', 'Tesla_Net_Income', 'Tesla_Assets',
'Tesla_Liabilities', 'Tesla_Cash_Flow', 'Apple_Revenue',
'Apple_Net_Income', 'Apple_Assets', 'Apple_Liabilities',
'Apple_Cash_Flow', 'Microsoft_Revenue_Growth_%',
'Microsoft_Net_Income_Growth_%', 'Microsoft_Assets_Growth_%',
'Microsoft_Liabilities_Growth_%', 'Microsoft_Cash_Flow_Growth_%',
'Tesla_Revenue_Growth_%', 'Tesla_Net_Income_Growth_%',
'Tesla_Assets_Growth_%', 'Tesla_Liabilities_Growth_%',
'Tesla_Cash_Flow_Growth_%', 'Apple_Revenue_Growth_%',
'Apple_Net_Income_Growth_%', 'Apple_Assets_Growth_%',
'Apple_Liabilities_Growth_%', 'Apple_Cash_Flow_Growth_%']
```

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Create long format dataframe for Assets & Liabilities
assets_liab = pd.DataFrame()

companies = ['Microsoft', 'Tesla', 'Apple']
for company in companies:
    temp = df[['Year', f'{company}_Assets',
f'{company}_Liabilities']].copy()
    temp['Company'] = company
    temp = temp.rename(columns={f'{company}_Assets': 'Assets',
f'{company}_Liabilities': 'Liabilities'})
    assets_liab = pd.concat([assets_liab, temp], ignore_index=True)

# Melt the dataframe
assets_liab_melted = assets_liab.melt(id_vars=['Year', 'Company'],
                                      value_vars=['Assets',
'Liabilities'],
                                      var_name='Metric',
value_name='Value')

plt.figure(figsize=(12, 6))
sns.barplot(data=assets_liab_melted, x='Year', y='Value',
hue='Metric', ci=None,
            palette='Set2', edgecolor='black')

plt.title("Total Assets vs Total Liabilities (2021–2023)")
plt.xlabel("Year")
plt.ylabel("Value (in Billions)")
plt.legend(title='Metric')
```

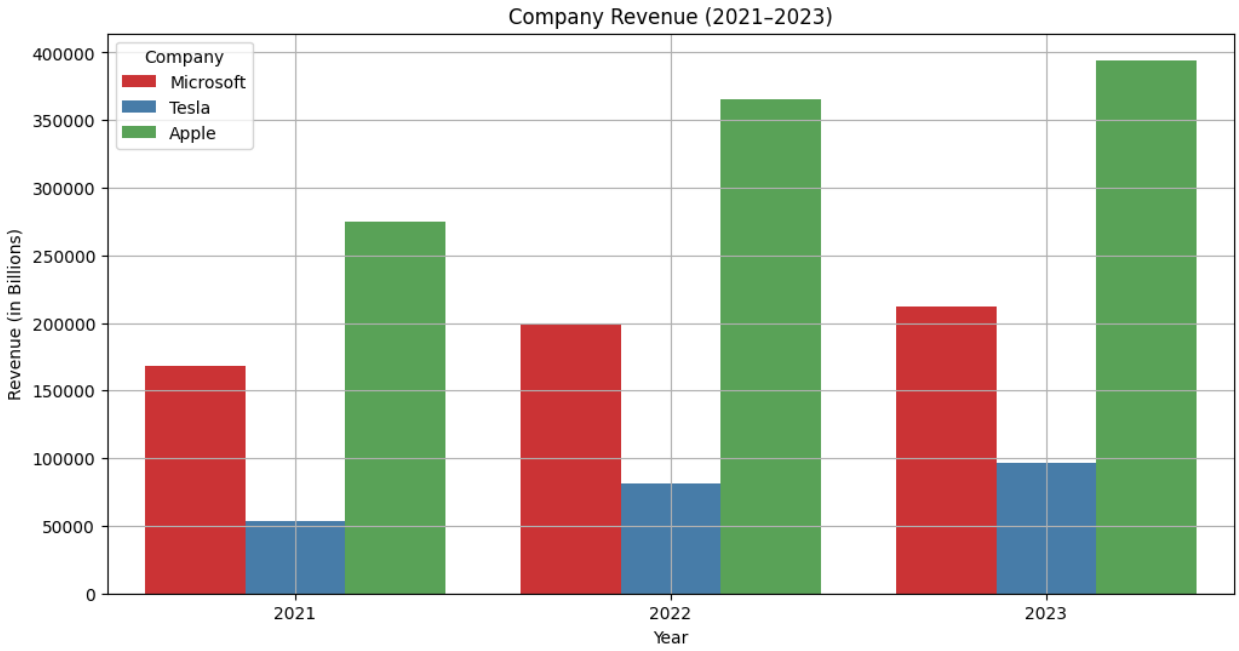
```
plt.grid(True)
plt.show()
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_10188\3549175415.py:2:
FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same
effect.

  sns.barplot(data=assets_liab_melted, x='Year', y='Value',
hue='Metric', ci=None,
```



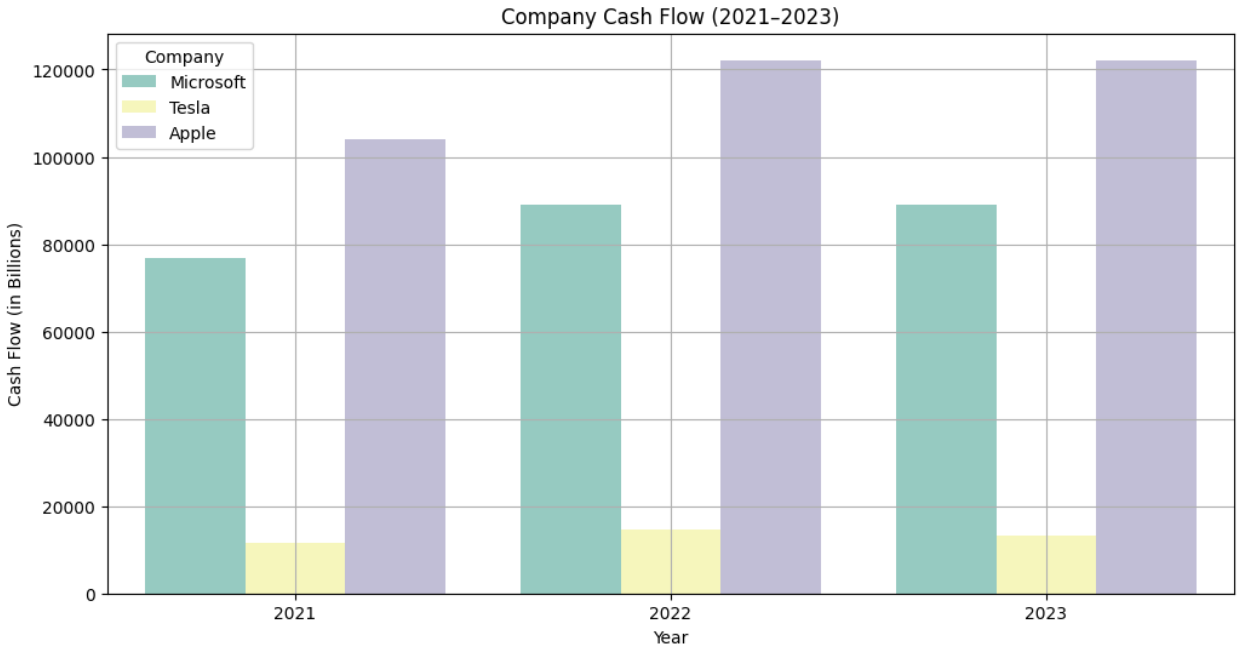

```
import pandas as pd
import seaborn as sns # Seaborn for barplots
import matplotlib.pyplot as plt

companies = ['Microsoft', 'Tesla', 'Apple']

# Prepare Revenue data
revenue_df = pd.DataFrame()
for company in companies:  #For loops for better plotting
    temp = df[['Year', f'{company}_Revenue']].copy()
    temp['Company'] = company
    temp = temp.rename(columns={f'{company}_Revenue': 'Revenue'})
    revenue_df = pd.concat([revenue_df, temp], ignore_index=True)

plt.figure(figsize=(12, 6))
sns.barplot(data=revenue_df, x='Year', y='Revenue', hue='Company',
palette='Set1')
```

```
plt.title("Company Revenue (2021–2023)")
plt.ylabel("Revenue (in Billions)")
plt.grid(True)
plt.show()
```




```
# Prepare Net Income data
net_income_df = pd.DataFrame()
for company in companies:
    temp = df[['Year', f'{company}_Net_Income']].copy()
    temp['Company'] = company
    temp = temp.rename(columns={f'{company}_Net_Income':
'Net_Income'})
    net_income_df = pd.concat([net_income_df, temp],
ignore_index=True)

plt.figure(figsize=(12, 6))
sns.barplot(data=net_income_df, x='Year', y='Net_Income',
hue='Company', palette='Set2')
plt.title("Company Net Income (2021–2023)")
plt.ylabel("Net Income (in Billions)")
plt.grid(True)
plt.show()
```


```
# Prepare Cash Flow data
cash_flow_df = pd.DataFrame()
for company in companies:
    temp = df[['Year', f'{company}_Cash_Flow']].copy()
    temp['Company'] = company
    temp = temp.rename(columns={f'{company}_Cash_Flow': 'Cash_Flow'})
    cash_flow_df = pd.concat([cash_flow_df, temp], ignore_index=True)

plt.figure(figsize=(12, 6))
sns.barplot(data=cash_flow_df, x='Year', y='Cash_Flow', hue='Company',
palette='Set3')
plt.title("Company Cash Flow (2021–2023)")
plt.ylabel("Cash Flow (in Billions)")
plt.grid(True)
plt.show()
```



#Financial Document Analysis: Microsoft, Tesla, Apple (2021–2023)

Methodology

- Extracted key financial metrics (Revenue, Net Income, Assets, Liabilities, Cash Flow) from 10-K and 10-Q filings for Microsoft, Tesla, and Apple from 2021 to 2023.
- Organized data into a structured format for multi-year, multi-company comparison in an Excel File.
- Performed growth rate calculations for each metric year-over-year.
- Visualized trends using bar plots and line graphs for quick insights.
- Used pandas, matplotlib, and seaborn for data analysis and visualization.

Key Observations

- Revenue Growth: Apple maintained the highest revenue overall, while Tesla showed the fastest growth rate across all years.
- Net Income Trends: Microsoft had stable and strong profitability. Tesla's net income significantly improved from 2021 to 2023.
- Assets vs. Liabilities: All three companies maintained healthy asset positions, with Microsoft having the most consistent asset-to-liability ratio.
- Cash Flow Insights: Apple's operating cash flow remained strong, while Tesla showed marked improvement in free cash flow.

Conclusion

- Tesla is in a strong growth phase with improving fundamentals.
- Microsoft and Apple maintain financial stability with high profitability and cash reserves.
- This structured data extraction and visualization lays the foundation for automating financial analysis with GenAI tools.