Done By: Jahnavi Chintakindi

BCGx: Data Science(Forage)

Task 1: Data extraction and initial analysis Your task is to manually extract key financial data for the last three fiscal years from the 10-K filings of Microsoft, Tesla, and Apple. Following the data collection, you will use Python to analyze this data, focusing on trends and insights that could inform the development of an AI-powered financial chatbot.

Step 1: Data extraction Navigate to the SEC's EDGAR database:

Microsoft Tesla Apple Manual extraction:

For each company, find the 10-K filings for the last three fiscal years. Extract the following financial figures: Total Revenue, Net Income, Total Assets, Total Liabilities, and Cash Flow from Operating Activities. Organize Your Data:

Compile the extracted data into an Excel spreadsheet for easy reference during your Python analysis.

Step 2: Preparing your Jupyter Notebook environment Install Jupyter (if not already installed):

Install Jupyter using pip if you haven't already: pip install notebook Launch Jupyter Notebook: jupyter notebook This command should open Jupyter in your web browser. Create a new notebook:

In the Jupyter interface, create a new notebook for your analysis. Step 3: Python analysis in Jupyter Import pandas:

At the beginning of your notebook, import the pandas library to work with your data. import pandas as pd Load your data:

Convert your Excel file to a CSV file for easier handling, then load it into a DataFrame. df = pd.read_csv('path_to_your_csv_file.csv') Analyzing trends with pandas:

Use pandas to calculate year-over-year changes for each financial metric. You can do this by creating new columns in your DataFrame that represent the percentage change from one year to the next. df['Revenue Growth (%)'] = df.groupby(['Company'])['Total Revenue'].pct_change() * 100 df['Net Income Growth (%)'] = df.groupby(['Company'])['Net Income'].pct_change() * 100 Explore other aggregate functions or groupings to analyze the data across different dimensions (by company, over years, etc.). Summarizing findings:

Conclude your analysis by summarizing your findings directly in the notebook. Use markdown cells to add narrative explanations of your analysis, discussing the trends and changes in financial metrics you've identified. Step 4: Documentation and submission Document your analysis: Use the markdown feature in Jupyter Notebook to document your methodology, observations, and conclusions throughout the notebook. Export your notebook: Once your analysis is complete, export your Jupyter Notebook as a PDF or HTML file for submission. You can do this from the "File" menu in Jupyter, selecting "Download as" and then choosing your preferred format. This approach allows you to focus on the core analytical aspects using pandas within a Jupyter Notebook, providing a clear, documented narrative of your financial analysis process. By the end of this task, you'll have a comprehensive understanding of how to analyze financial data programmatically, a valuable skill set for data-driven decision-making.

```
import pandas as pd
```

```
df = pd.read_csv('/content/DataCollection.csv')
```

```
df.head(11)
```

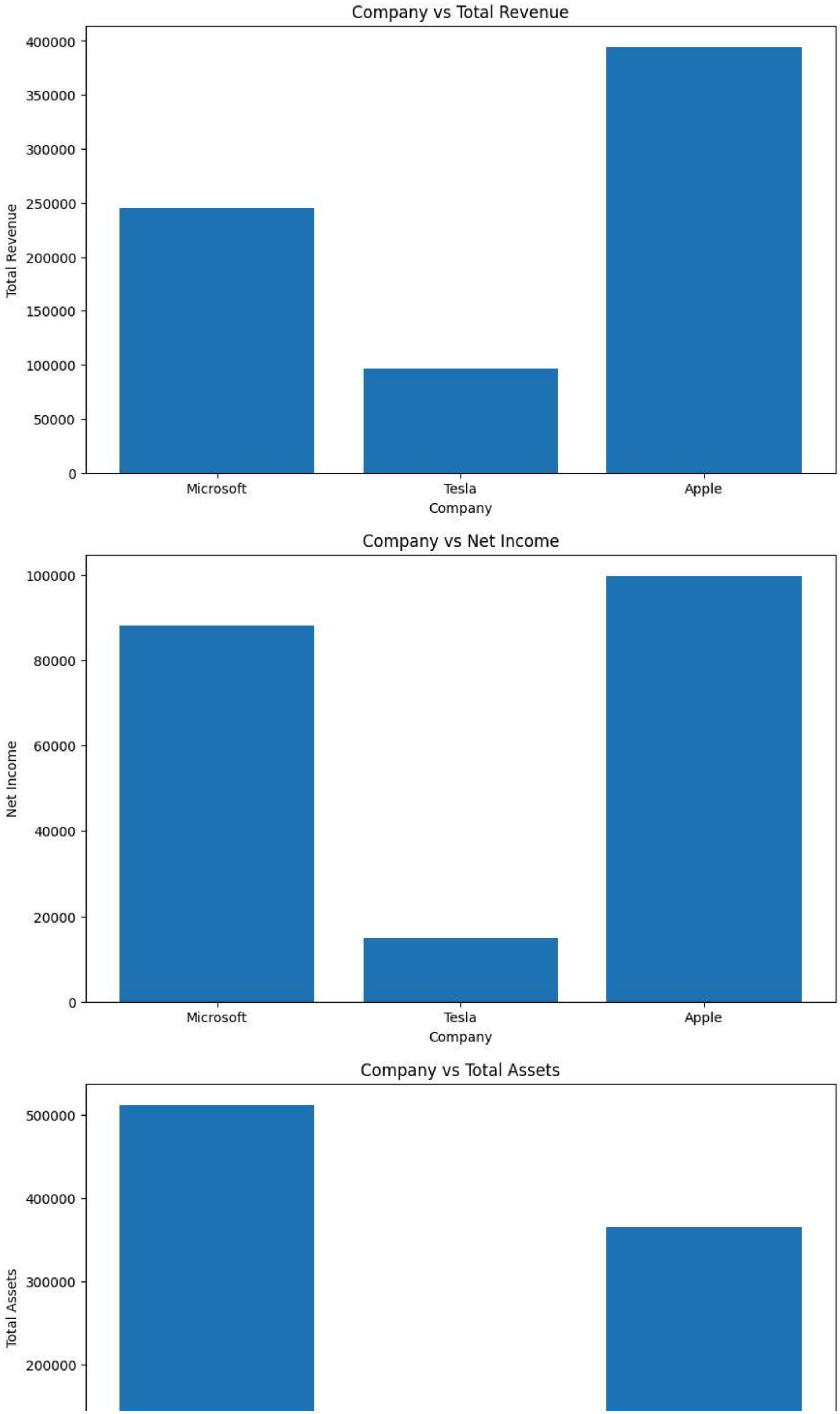
	Company	Fiscal Year	Total Revenue	Net Income	Total Assets	Total Liabilities	Cash Flow from Operating Activities
0	Microsoft	FY2024	$245,122.00	$88,136.00	$512,163.00	$243,686.00	$118,548.00
1	Microsoft	FY2023	$211,915.00	$72,361.00	$411,576.00	$205,753.00	$87,582.00
2	Microsoft	FY2022	$198,270.00	$72,738.00	$312,761.00	$200,101.00	$89,035.00
3	Tesla	FY2023	$96,773.00	$14,974.00	$106,618.00	$43,009.00	$13,256.00
4	Tesla	FY2022	$81,462.00	$12,587.00	$82,338.00	$36,440.00	$14,724.00
5	Tesla	FY2021	$53,823.00	$5,644.00	$62,131.00	$30,548.00	$11,497.00
6	Apple	FY2024	$391,035.00	$93,736.00	$364,980.00	$308,030.00	$118,254.00
7	Apple	FY2023	$383,285.00	$96,995.00	$352,583.00	$290,437.00	$110,543.00
8	Apple	FY2022	$394,328.00	$99,803.00	$352,755.00	$302,083.00	$122,151.00

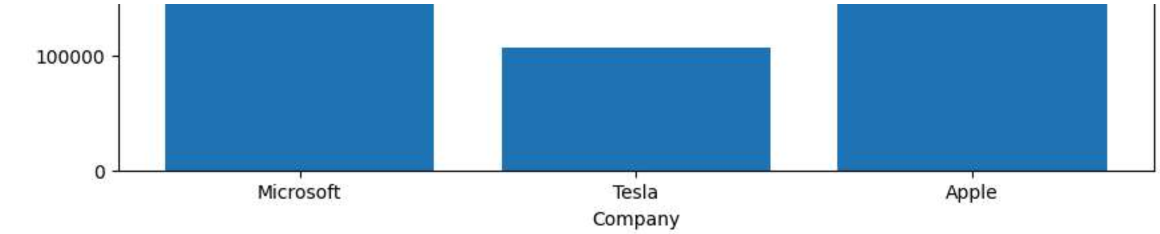
Next steps: Generate code with df View recommended plots New interactive sheet

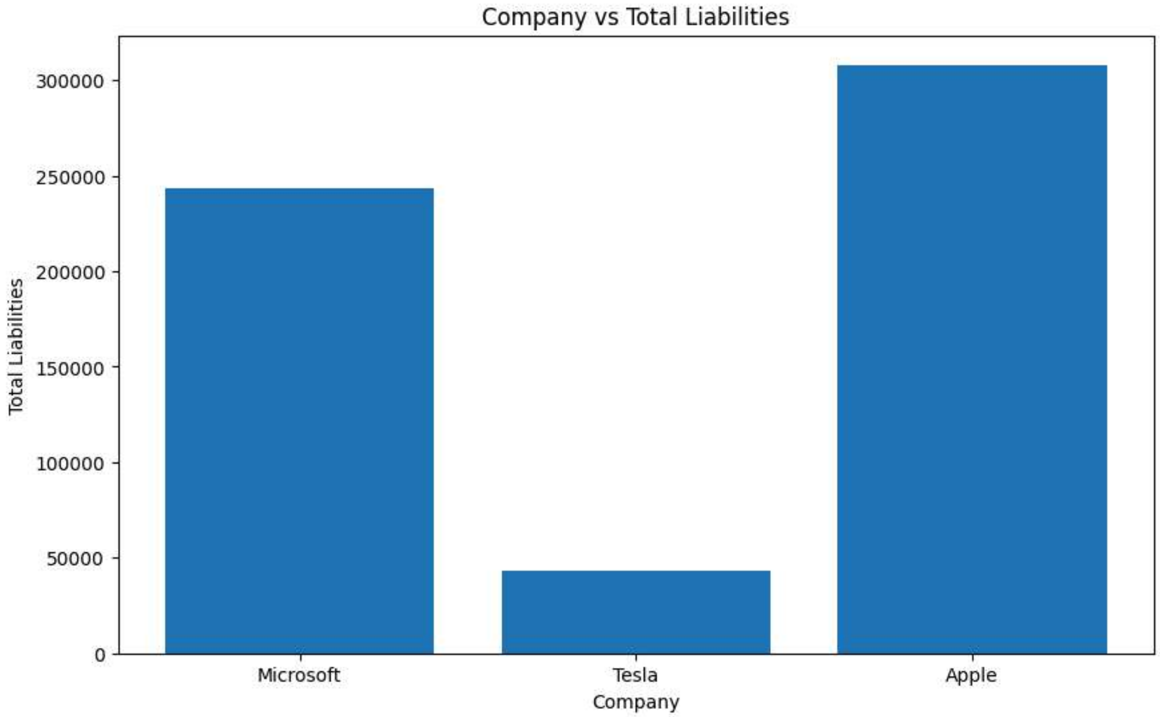
```
df.head()
```

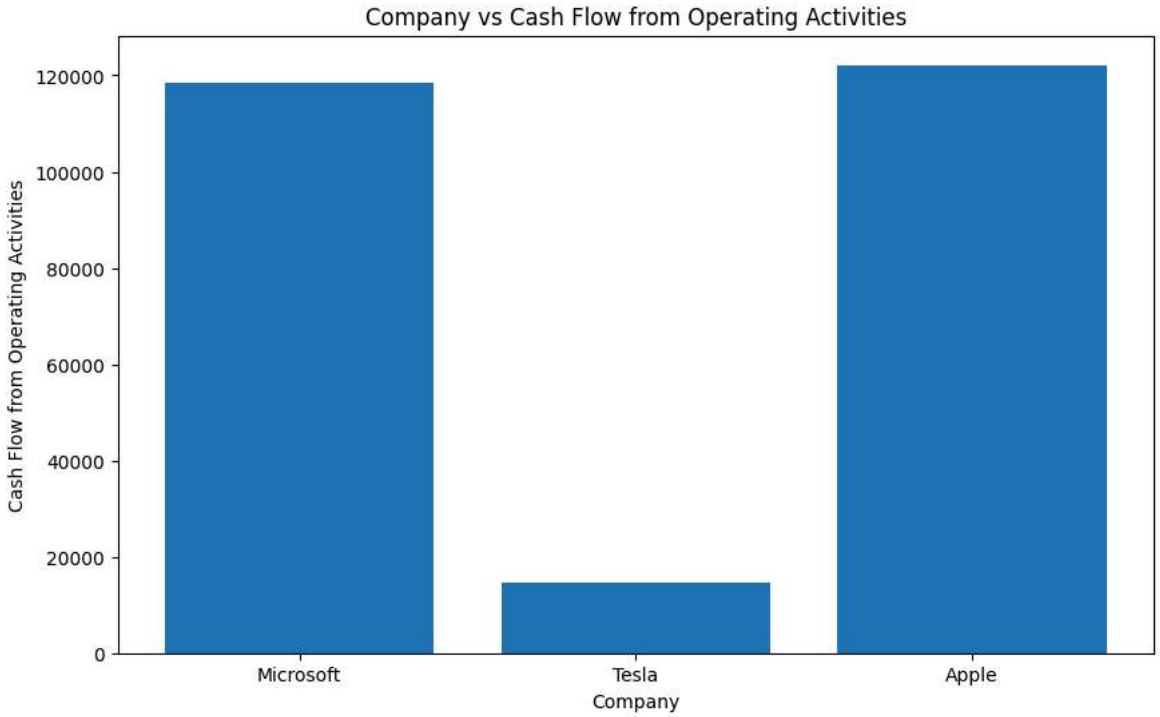
	Company	Fiscal Year	Total Revenue	Net Income	Total Assets	Total Liabilities	Cash Flow from Operating Activities
0	Microsoft	FY2024	245122.0	88136.0	512163.0	243686.0	118548.0
1	Microsoft	FY2023	211915.0	72361.0	411576.0	205753.0	87582.0
2	Microsoft	FY2022	198270.0	72738.0	312761.0	200101.0	89035.0
3	Tesla	FY2023	96773.0	14974.0	106618.0	43009.0	13256.0
4	Tesla	FY2022	81462.0	12587.0	82338.0	36440.0	14724.0

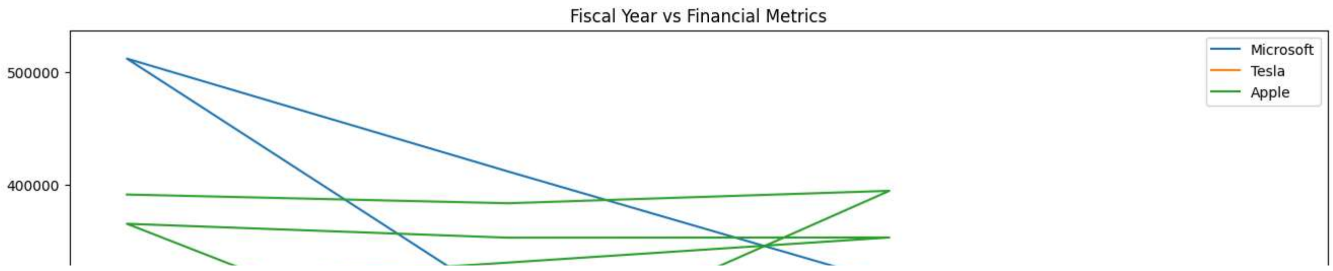
```
# prompt: Using dataframe df: company vs Total Revenue
# company vs    Net Income
# company vs Total Assets
# company vs    Total Liabilities
# company vs    Cash Flow from Operating Activities
# fiscal year of Total Revenue  Net Income  Total Assets    Total Liabilities   Cash Flow from Operating Activities VS company

import pandas as pd
import matplotlib.pyplot as plt

# Company vs Total Revenue
plt.figure(figsize=(10, 6))
plt.bar(df['Company'], df['Total Revenue'])
plt.xlabel('Company')
plt.ylabel('Total Revenue')
plt.title('Company vs Total Revenue')
plt.show()

# Company vs Net Income
plt.figure(figsize=(10, 6))
plt.bar(df['Company'], df['Net Income'])
plt.xlabel('Company')
plt.ylabel('Net Income')
plt.title('Company vs Net Income')
plt.show()

# Company vs Total Assets
plt.figure(figsize=(10, 6))
plt.bar(df['Company'], df['Total Assets'])
plt.xlabel('Company')
plt.ylabel('Total Assets')
plt.title('Company vs Total Assets')
plt.show()


# Company vs Total Liabilities
plt.figure(figsize=(10, 6))
plt.bar(df['Company'], df['Total Liabilities'])
plt.xlabel('Company')
plt.ylabel('Total Liabilities')
plt.title('Company vs Total Liabilities')
plt.show()

# Company vs Cash Flow from Operating Activities
plt.figure(figsize=(10, 6))
plt.bar(df['Company'], df['Cash Flow from Operating Activities'])
plt.xlabel('Company')
plt.ylabel('Cash Flow from Operating Activities')
plt.title('Company vs Cash Flow from Operating Activities')
plt.show()

# Fiscal Year vs Financial Metrics
#Reshape the data for better visualization
df_melted = pd.melt(df, id_vars=['Company', 'Fiscal Year'], value_vars=['Total Revenue', 'Net Income', 'Total Assets', 'Total Liabilities',

#Create the plot
plt.figure(figsize=(16, 8))
for company in df_melted["Company"].unique():
  plt.plot(df_melted[df_melted["Company"] == company]['Fiscal Year'], df_melted[df_melted["Company"] == company]['value'], label=company)
plt.xlabel('Fiscal Year')
plt.ylabel('Value')
plt.title('Fiscal Year vs Financial Metrics')
plt.legend()
plt.show()
```

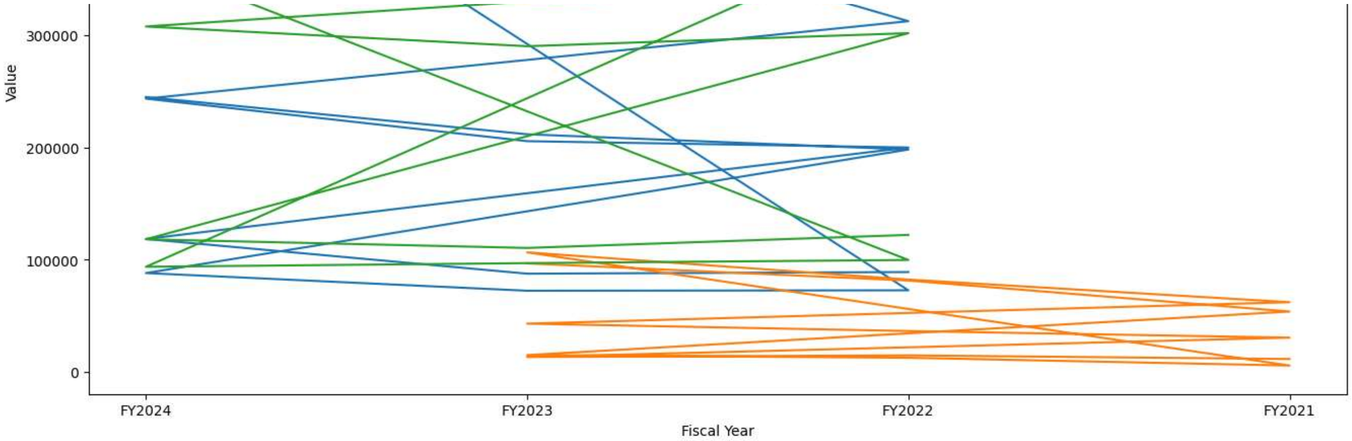

Company vs Total Revenue
Total Revenue
400000
350000
300000
250000
200000
150000
100000
50000
0
Microsoft
Tesla
Apple
Company
Company vs Net Income
Net Income
100000
80000
60000
40000
20000
0
Microsoft
Tesla
Apple
Company
Company vs Total Assets
Total Assets
500000
400000
300000
200000


100000
0
Microsoft
Tesla
Apple
Company


Company vs Total Liabilities
Total Liabilities
300000
250000
200000
150000
100000
50000
0
Microsoft
Tesla
Apple
Company


Company vs Cash Flow from Operating Activities
Cash Flow from Operating Activities
120000
100000
80000
60000
40000
20000
0
Microsoft
Tesla
Apple
Company

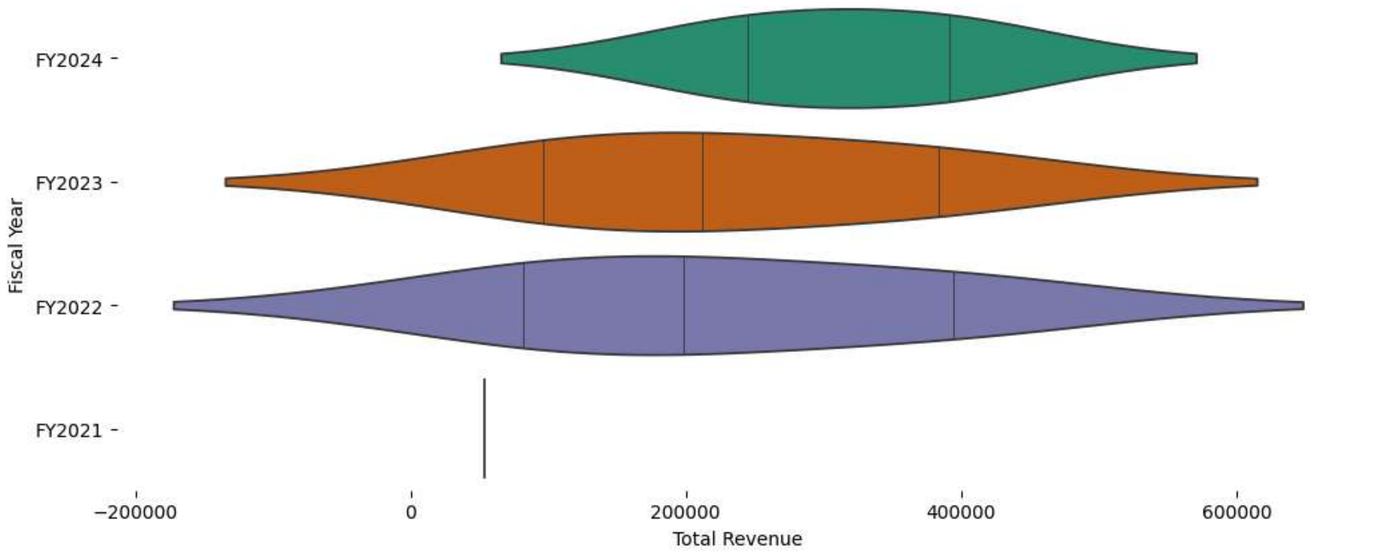

Fiscal Year vs Financial Metrics
500000
400000
Microsoft
Tesla
Apple


Value
300000
200000
100000
0
FY2024
FY2023
FY2022
FY2021
Fiscal Year

Fiscal Year vs Total Revenue

```
# @title Fiscal Year vs Total Revenue

from matplotlib import pyplot as plt
import seaborn as sns
figsize = (12, 1.2 * len(df['Fiscal Year'].unique()))
plt.figure(figsize=figsize)
sns.violinplot(df, x='Total Revenue', y='Fiscal Year', inner='stick', palette='Dark2')
sns.despine(top=True, right=True, bottom=True, left=True)
```

```
<ipython-input-56-643055938599>:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legenc

  sns.violinplot(df, x='Total Revenue', y='Fiscal Year', inner='stick', palette='Dark2')
```



Company vs Total Revenue

```
# @title Company vs Total Revenue

from matplotlib import pyplot as plt
import seaborn as sns
figsize = (12, 1.2 * len(df['Company'].unique()))
plt.figure(figsize=figsize)
sns.violinplot(df, x='Total Revenue', y='Company', inner='stick', palette='Dark2')
sns.despine(top=True, right=True, bottom=True, left=True)
```

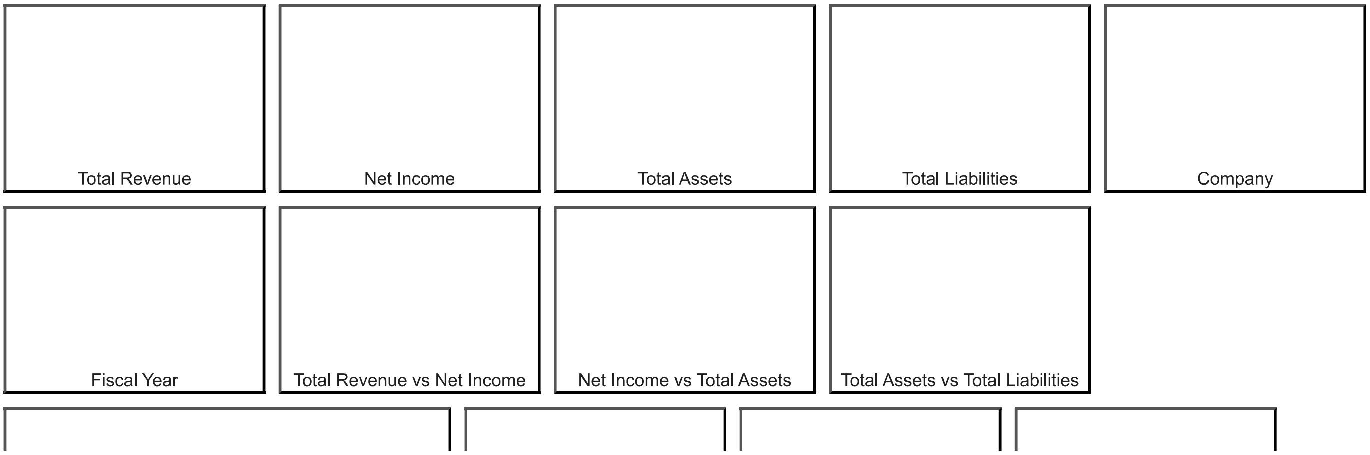
```
<ipython-input-55-0dd03db2012a>:7: FutureWarning:
```

```
df['Revenue Growth (%)'] = df.groupby(['Company'])['Total Revenue'].pct_change() * 100
df['Net Income Growth (%)'] = df.groupby(['Company'])['Net Income'].pct_change() * 100
df['Total Assets Growth (%)'] = df.groupby(['Company'])['Total Assets'].pct_change() * 100
df['Total Liabilities Growth (%)'] = df.groupby(['Company'])['Total Liabilities'].pct_change() * 100
df['Cash Flow from Operating Activities Growth (%)'] = df.groupby(['Company'])['Cash Flow from Operating Activities'].pct_change() * 100
```

```
df.head(10)
```

	Company	Fiscal Year	Total Revenue	Net Income	Total Assets	Total Liabilities	Cash Flow from Operating Activities	Revenue Growth (%)	Net Income Growth (%)	Total Assets Growth (%)	Total Liabilities Growth (%)	Cash Flow from Operating Activities Growth (%)
0	Microsoft	FY2024	245122.0	88136.0	512163.0	243686.0	118548.0	0.000000	0.000000	0.000000	0.000000	0.000000
1	Microsoft	FY2023	211915.0	72361.0	411576.0	205753.0	87582.0	-13.547132	-17.898475	-19.639646	-15.566344	-26.121065
2	Microsoft	FY2022	198270.0	72738.0	312761.0	200101.0	89035.0	-6.438902	0.520999	-24.008932	-2.746983	1.659017
3	Tesla	FY2023	96773.0	14974.0	106618.0	43009.0	13256.0	0.000000	0.000000	0.000000	0.000000	0.000000
4	Tesla	FY2022	81462.0	12587.0	82338.0	36440.0	14724.0	-15.821562	-15.940964	-22.772890	-15.273547	11.074231
5	Tesla	FY2021	53823.0	5644.0	62131.0	30548.0	11497.0	-33.928703	-55.160086	-24.541524	-16.169045	-21.916599
6	Apple	FY2024	391035.0	93736.0	364980.0	308030.0	118254.0	0.000000	0.000000	0.000000	0.000000	0.000000
7	Apple	FY2023	383285.0	96995.0	352583.0	290437.0	110543.0	-1.981920	3.476786	-3.396624	-5.711457	-6.520710
8	Apple	FY2022	394328.0	99803.0	352755.0	302083.0	122151.0	2.881146	2.894995	0.048783	4.009820	10.500891

Next steps: Generate code with df | View recommended plots | New interactive sheet


Total Revenue
Net Income
Total Assets
Total Liabilities
Company
Fiscal Year
Total Revenue vs Net Income
Net Income vs Total Assets
Total Assets vs Total Liabilities