Financial Analysis of 3 major Consultancy Firms using Python

Libraries used:
1- Pandas
2-MatplotLib

IT Industry:

Computer Systems Design and Related Services (SIC code 7373)

The IT industry is a cornerstone of modern society, influencing nearly every aspect of our lives, from communication and entertainment to education and business operations.

The IT industry has a significant global influence, driving economic growth, innovation, and digital transformation worldwide. By understanding the dynamics of this industry, we hope to gain a broader perspective on its global impact and implications.

Companies

Tata Consultancy Services (TCS)

Headquarters: Mumbai, India

IBM

Armonk, New York, United States

Capgemini SE

Headquarters: Paris, France

Inputting financial statements into python

1- Data Sources:

Extracting Financial Data in Excels:

Excel files with the comapny's financial data for the last 4 years i.e 2020 to 2023.
Balance sheet, Income statement and other financial elements

Financial Ratios & Metrices

Profitability Ratios	Liquidity Ratios	Solvency Ratios
Net profit Margin	Current Ratio	Debt to Equity Ratios
Gross profit Margin	Quick Ratio	Debt Ratios
Return on Assets(ROA)		
Return on equity(ROE)		

Financial Ratios & Metrices

Growth Ratios

- Revenue Growth Rate
- Earnings Ratio

Valuation Ratios

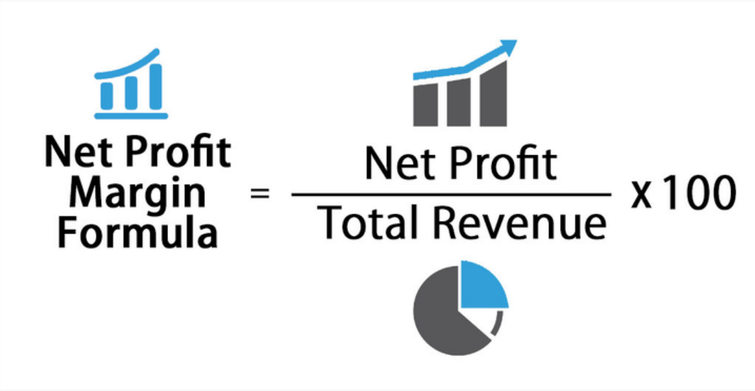
- Price-to-earnings (P/E)
- Price-to-book (P/B)

Effeciency Ratios

- Inventory Turnover Ratio
- Asset Turnover Ratio
- Dividend Yield

Computing different financial metrics of firms

Profitability ratio: calculating Net profit Margin
Company: TCS




```python
# Net profit value TCS ..................................................
import pandas as pd

def net_profit_margin_Tcs(net_profit, total_revenue):
    # Convert net_profit and total_revenue to numeric types (float)
    net_profit = float(net_profit)
    total_revenue = float(total_revenue)

    # Calculate the net profit margin
    net_profit_margin = (net_profit / total_revenue) * 100
    return net_profit_margin

# Read data from Excel file
file_path = r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Tata\Tata_Inc.xlsx'
df = pd.read_excel(file_path, sheet_name='Income statement')

# Iterate over 4 years
for year in range(2, 6):
    # Extract net profit and total revenue as strings for the current year
    net_profit_str = df.iloc[31, year]
    total_revenue_str = df.iloc[0, year]

    # Calculation of the net profit margin for the current year
    margin = net_profit_margin_Tcs(net_profit_str, total_revenue_str)

    print(f"The net profit margin of TCS for the year {2025 - year} is: {margin:.2f}%")

```

Console 1/A

```
...: df = pd.read_excel(file_path, sheet_name='Income sta
...:
...: # Iterate over 4 years
...: for year in range(2, 6):
...:     # Extract net profit and total revenue as string
...:     net_profit_str = df.iloc[31, year]
...:     total_revenue_str = df.iloc[0, year]
...:
...:     # Calculation of the net profit margin for the c
...:     margin = net_profit_margin_Tcs(net_profit_str, t
...:
...:     print(f"The net profit margin of TCS for the yea
{margin:.2f}%")
The net profit margin of TCS for the year 2023 is: 81.31%
The net profit margin of TCS for the year 2022 is: 80.01%
The net profit margin of TCS for the year 2021 is: 80.25%
The net profit margin of TCS for the year 2020 is: 79.39%
```

Computing different financial metrics of firms

Liquidity Ratio: Current Ratio
Company: IBM

$$\text{Current Ratio} = \frac{\text{Current Assets}}{\text{Current Liabilities}}$$

```python
# Current ratio ibm.....................................................
import pandas as pd

def current_ratio_ibm(Current_assets, Current_liabilities):
    # Convert current assets and current liabilities to numeric types (float)
    Current_assets = float(Current_assets)
    Current_liabilities = float(Current_liabilities)

    # Calculate the Current ratio
    curratio= (Current_assets / Current_liabilities)
    return curratio

# Read data from Excel file
file_path = r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Ibm\ibm_inc.xlsx'
df = pd.read_excel(file_path, sheet_name='Income statement')

# Iterate over 4 years
for year in range(2, 6):
    # Extract current assets and current liabilities as strings for the current year
    Current_assets_str = df.iloc[69, year]
    Current_liabilities_str = df.iloc[72, year]

    # Calculate the Current ratio for the current year
    margin = current_ratio_ibm(Current_assets_str, Current_liabilities_str)

    print(f"The Current ratio of IBM for {2025 - year} is: {margin:.2f}%")

```

```
Console 1/A
     ...: df = pd.read_excel(file_path, sheet_name='I
     ...:
     ...: # Iterate over 4 years
     ...: for year in range(2, 6):
     ...:     # Extract current assets and current li
current year
     ...:     Current_assets_str = df.iloc[69, year]
     ...:     Current_liabilities_str = df.iloc[72, y
     ...:
     ...:     # Calculate the Current ratio for the c
     ...:     margin = current_ratio_ibm(Current_asse
     ...:
     ...:     print(f"The Current ratio of IBM for {2
The Current ratio of IBM for 2023 is: 0.96%
The Current ratio of IBM for 2022 is: 0.92%
The Current ratio of IBM for 2021 is: 0.88%
The Current ratio of IBM for 2020 is: 0.98%

In [267]:
```

Computing different financial metrics of firms

Solvency Ratio: Debt to Equity ratio
Company: Capgemini

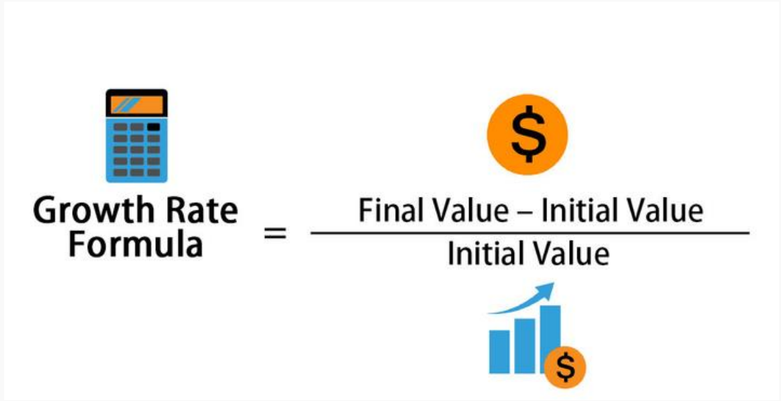
$$\text{Debt to Equity Ratio (D/E)} = \frac{\text{Total Debt}}{\text{Total Shareholders Equity}}$$

```python
import pandas as pd

def debt_to_equity_ratio_cap(total_debt, shareholder_equity):
    # Convert total debt and shareholder equity to numeric types (float)
    total_debt = float(total_debt)
    shareholder_equity = float(shareholder_equity)


    # Calculate the debt to equity ratio
    debt_to_equity_ratio = total_debt / shareholder_equity
    return debt_to_equity_ratio

# Read data from Excel file
file_path = r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Capgemini\Cap_Inc.xlsx'
df = pd.read_excel(file_path, sheet_name='Income statement')

# Iterate over 4 years
for year in range(2, 6):
    # Extract total debt and shareholder equity as strings for the current year
    total_debt_str = df.iloc[59, year]
    shareholder_equity_str = df.iloc[65, year]


    # Calculate the debt to equity ratio for the current year
    margin = debt_to_equity_ratio_cap(total_debt_str, shareholder_equity_str)

    print(f"The Debt to Equity ratio of Capgemini for {2025 - year} is: {margin:.2f}%")
```

```
Console 1/A
     ...:     shareholder_equity_str = df.iloc[65, year]
     ...:
     ...:
     ...:     # Calculate the debt to equity ratio for the current year
     ...:     margin = debt_to_equity_ratio_cap(total_debt_str, shareholder_equity_str)
     ...:
     ...:     print(f"The Debt to Equity ratio of Capgemini for {2025 - year} is:
{margin:.2f}%")
The Debt to Equity ratio of Capgemini for 2023 is: 0.63%
The Debt to Equity ratio of Capgemini for 2022 is: 0.58%
The Debt to Equity ratio of Capgemini for 2021 is: 0.90%
The Debt to Equity ratio of Capgemini for 2020 is: 1.48%

In [268]:
```

Computing different financial metrics of firms

Growth Ratio: Revenue Growth Rate
Company: TCS




```python
import pandas as pd
def revenue_growth_ratio_tcs(Current_revenue, previous_revenue):
    # Convert current year revenue and previous year revenue to numeric types (float)
    Current_revenue = float(Current_revenue)
    previous_revenue = float(previous_revenue)

    # Calculate the growth rate
    growth_rate = ((Current_revenue - previous_revenue) / previous_revenue)*100
    return growth_rate

# Read data from Excel file
file_path = r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Tata\Tata_Inc.xlsx'
df = pd.read_excel(file_path, sheet_name='Income statement')

# Iterate over 4 years
for year in range(2, 5):
    # Extract current year revenue and previous year revenue as strings for the current year
    Current_revenue_str = df.iloc[0, year]
    previous_revenue_str = df.iloc[0, year + 1]

# Calculate the growth rate for the current year
    growth_rate = revenue_growth_ratio_tcs(Current_revenue_str, previous_revenue_str)

 # Check if growth rate is negative
    if growth_rate < 0:
        growth_rate_str = f"{growth_rate:.2f}% (Decrease)"
    else:
        growth_rate_str = f"{growth_rate:.2f}% (Increase)"

    print(f"The growth rate of TCS for year {2025 - year} is: {growth_rate_str}")

```

```
Console 1/A
(Decrease)
    ...:     else:
    ...:         growth_rate_str = f"{growth_rate:.2f}%
(Increase)"
    ...:
    ...:     print(f"The growth rate of TCS for year {2025 -
is: {growth_rate_str}")
The growth rate of TCS for year 2023 is: 17.58% (Increase)
The growth rate of TCS for year 2022 is: 16.80% (Increase)
The growth rate of TCS for year 2021 is: 4.61% (Increase)

In [270]:
```

Computing different financial metrics of firms

Valuation Ratios: Price to earning ratio
Company: IBM

$$\text{P/E Ratio} = \frac{\text{Market Share Price}}{\text{Earnings Per Share (EPS)}}$$

```python
# P/E ratio- ibm.....................................................................
import pandas as pd

def pe_ratio_ibm(mps,eps):
    # Convert market price per share and earnings per share to numeric types (float)
    mps = float(mps)
    eps = float(eps)

    # Calculate the p/e ratio
    pe_ratio= (mps / eps)
    return pe_ratio

# Read data from Excel file
file_path = r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Ibm\ibm_inc.xlsx'
df = pd.read_excel(file_path, sheet_name='Income statement')

# Iterate over 4 years
for year in range(2, 6):
    # Extract market price per share and earnings per share as strings for the current year
    mps_str = df.iloc[87, year]
    eps_str = df.iloc[88, year]

    # Calculate the p/e ratio for the current year
    margin = pe_ratio_ibm(mps_str,eps_str)

    print(f"The Price to earnings ratio of IBM for {2025 - year} is: {margin:.2f}%")

```

```
Console 1/A
is: {margin:.2f}%")
The Price to earnings ratio of IBM for 2023 is: 11.42%
The Price to earnings ratio of IBM for 2022 is: 60.94%
The Price to earnings ratio of IBM for 2021 is: 18.46%
The Price to earnings ratio of IBM for 2020 is: 19.98%

In [271]:
```

Computing different financial metrics of firms

Effeciency Ratio: Asset Turnover Ratio
Company: Capgemini

$$\text{Asset Turnover Ratio} = \frac{\text{Net Sales}}{\text{Average Total Assets}}$$

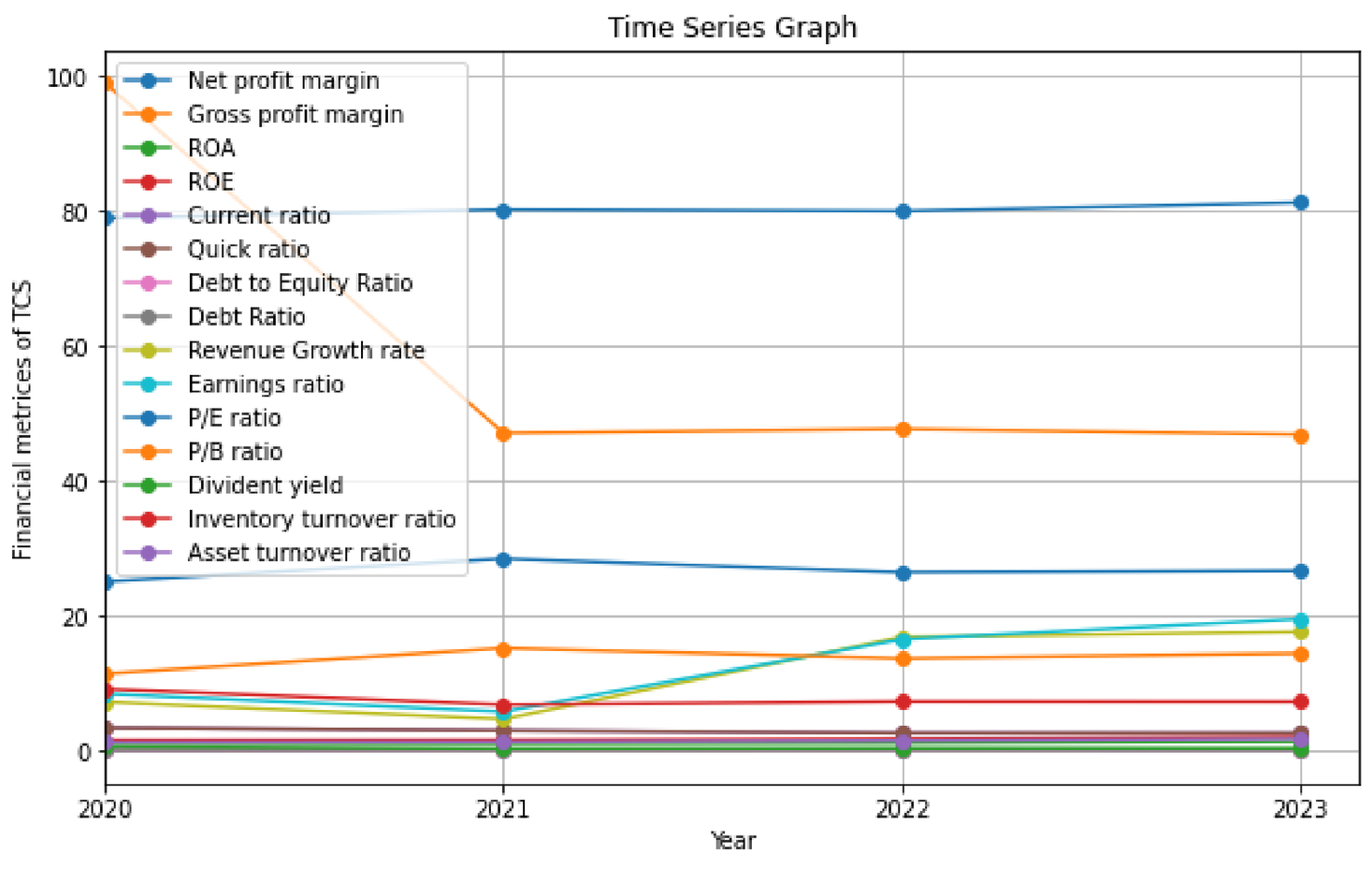
```python
import pandas as pd

def asset_turnover_ratio_cap(net_sales, avg_total_assets):
    # Convert net sales and average total assets to numeric types (float)
    net_sales = float(net_sales)
    avg_total_assets = float(avg_total_assets)

    # Calculate the asset turnover ratio
    asset_turnover_ratio= (net_sales / avg_total_assets)
    return asset_turnover_ratio

# Read data from Excel file
file_path = r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Capgemini\Cap_Inc.xlsx'
df = pd.read_excel(file_path, sheet_name='Income statement')

# Iterate over 4 years
for year in range(2, 6):
    # Extract net sales and average total assets as strings for the current year
    net_sales_str = df.iloc[0, year]
    avg_total_assets = df.iloc[104, year]

    # Calculate the asset turnover ratio for the current year
    margin = asset_turnover_ratio_cap(net_sales_str, avg_total_assets)

    print(f"The Asset Turnover ratio of Capgemini for {2025 - year} is: {margin:.2f}%")
```

```
Console 1/A
     ...:     margin = asset_turnover_ratio_cap(net_sales_str, avg_
     ...:
     ...:     print(f"The Asset Turnover ratio of Capgemini for {20
{margin:.2f}%")
The Asset Turnover ratio of Capgemini for 2023 is: 0.94%
The Asset Turnover ratio of Capgemini for 2022 is: 0.91%
The Asset Turnover ratio of Capgemini for 2021 is: 0.76%
The Asset Turnover ratio of Capgemini for 2020 is: 0.66%

In [272]:
```

Graphical Representation of all financial metrices for the 3 companies

TCS

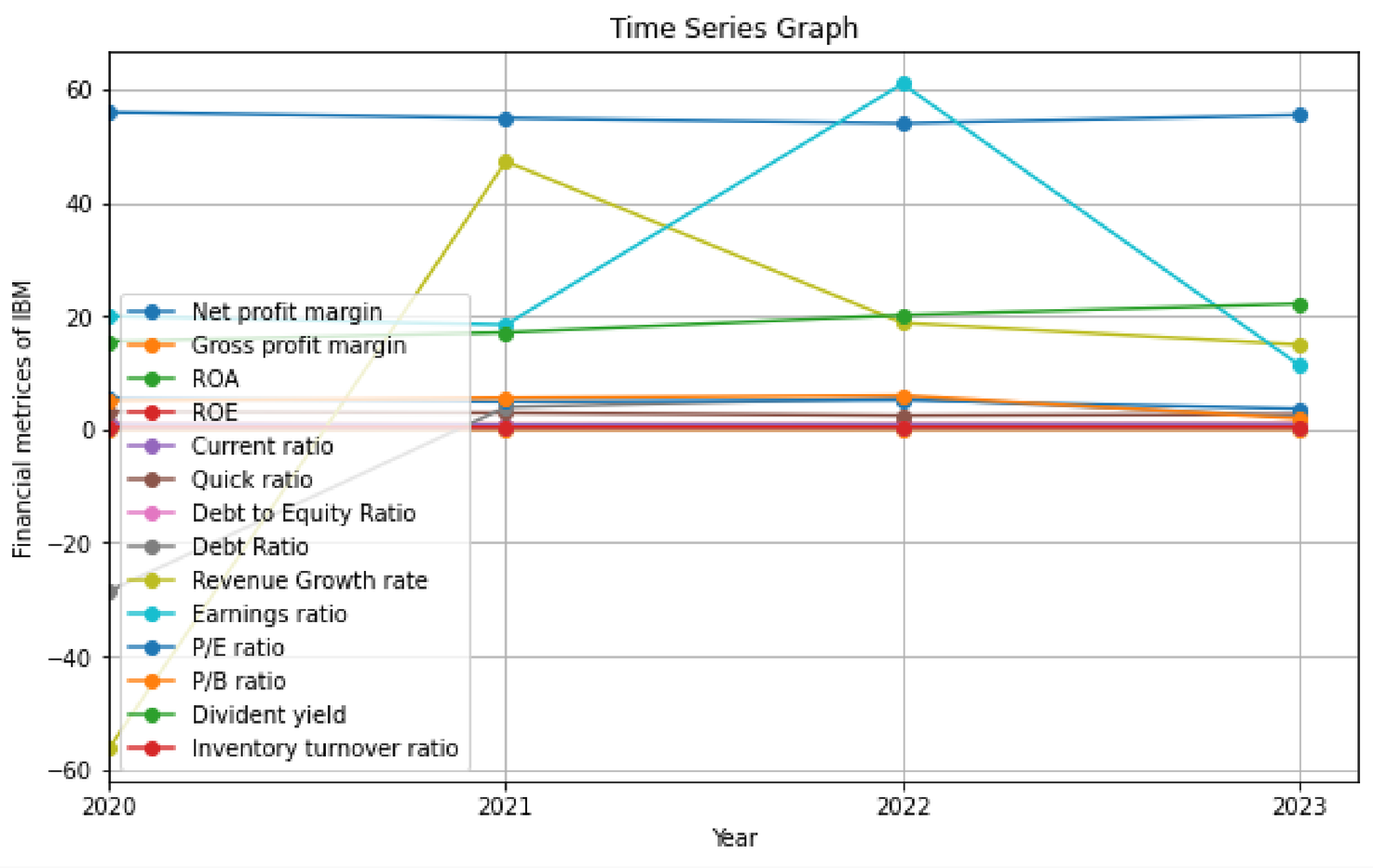
Elements	2020	2021	2022	2023
Net profit Margin	79.00	80.25	80.01	81.31
Gross profit Margin	99.00	47.08	47.71	46.88
Return on Assets(ROA)	1.03	1.01	1.08	1.28
Return on Equity (roe)	1.48	1.51	1.71	2.01
Current Ratio	3.33	2.91	2.56	2.53
Quick ratio	3.33	2.90	2.55	2.53
Debt to Equity Ratio	0.01	0.09	0.09	0.08
Debt Ratio	0.01	0.06	0.06	0.05
Revenue Growth rate	7.16	4.61	16.80	17.58
Earnings ratio	8.37	5.73	16.46	19.48
P/E ratio	25.04	28.43	26.40	26.64
P/B ratio	11.36	15.15	13.63	14.37
Divident yield	0.50	0.22	0.26	0.28
Inventory turnover ratio	9.08	6.76	7.23	7.19
Asset turnover ratio	1.17	1.22	1.43	1.68



Graphical Representation of all financial metrices for the 3 companies

IBM

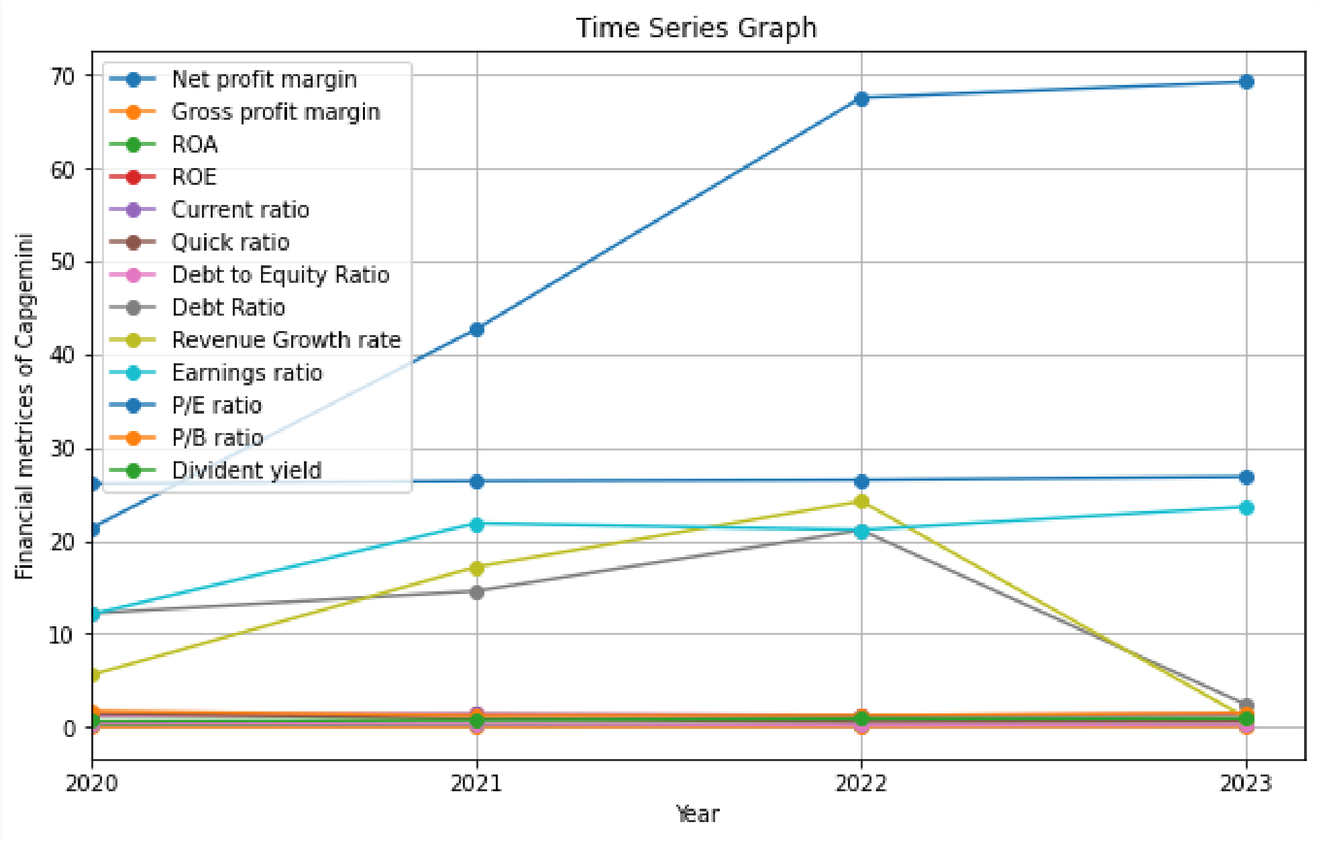
Elements	2020	2021	2022	2023
Net profit Margin	8.45	11.97	13.47	15.15
Gross profit Margin	55.94	54.90	54.00	55.45
Return on Assets(ROA)	0.03	0.05	0.06	0.07
Return on Equity (roe)	0.22	0.36	0.37	0.41
Current Ratio	0.98	0.88	0.92	0.96
Quick ratio	0.94	0.83	0.88	0.93
Debt to Equity Ratio	3.14	2.90	2.45	2.65
Debt Ratio	0.42	0.42	0.42	0.44
Revenue Growth rate	-28.48	3.94	5.54	2.20
Earnings ratio	-56.13	47.25	18.79	14.95
P/E ratio	19.98	18.46	60.94	11.42
P/B ratio	5.53	4.91	5.19	3.65
Divident yield	5.05	5.57	6.01	1.91
Inventory turnover ratio	15.52	17.10	20.16	22.17
Asset turnover ratio	0.40	0.42	0.44	0.45



Graphical Representation of all financial metrices for the 3 companies

Capgemini

Elements	2020	2021	2022	2023
Net profit Margin	11.09	11.34	11.63	11.47
Gross profit Margin	26.10	26.39	26.52	26.85
Return on Assets(ROA)	0.08	0.09	0.10	0.10
Return on Equity (roe)	0.29	0.24	0.26	0.25
Current Ratio	1.31	1.38	1.28	1.34
Quick ratio	1.31	1.38	1.28	1.34
Debt to Equity Ratio	1.48	0.90	0.58	0.63
Debt Ratio	0.41	0.32	0.22	0.27
Revenue Growth rate	12.20	14.59	21.12	2.40
Earnings ratio	5.59	17.18	24.22	0.98
P/E ratio	12.09	21.86	21.15	23.62
P/B ratio	21.32	42.67	67.52	69.20
Divident yield	1.72	1.24	1.26	1.53
Asset turnover ratio	0.66	0.76	0.91	0.94

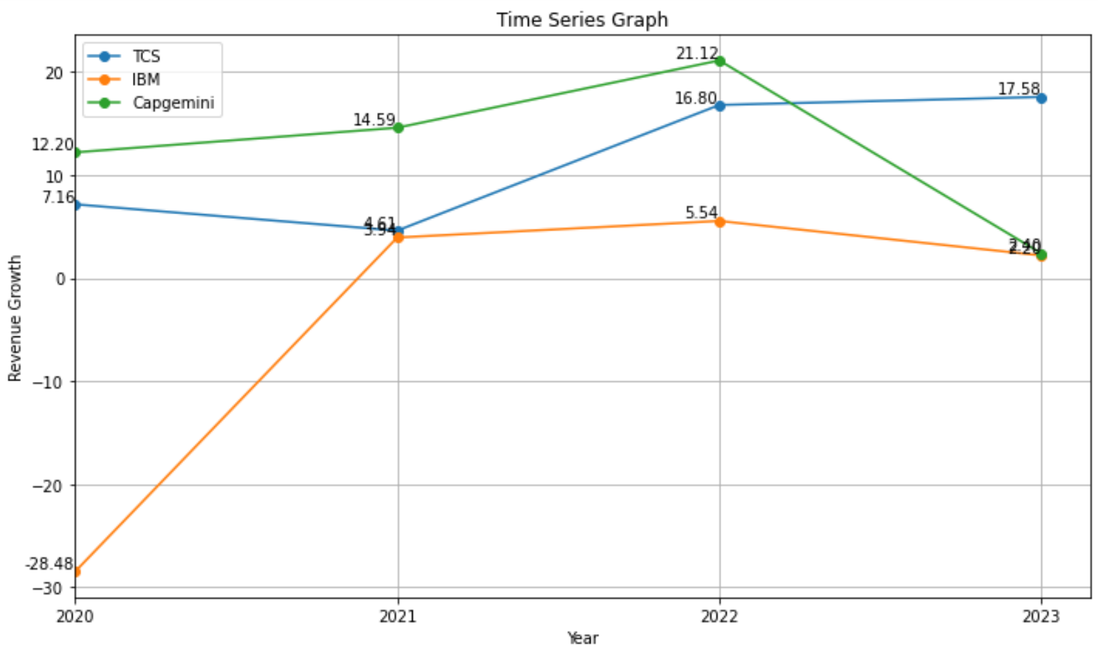


Python Program used to plot graph for company's financials (Capgemini)

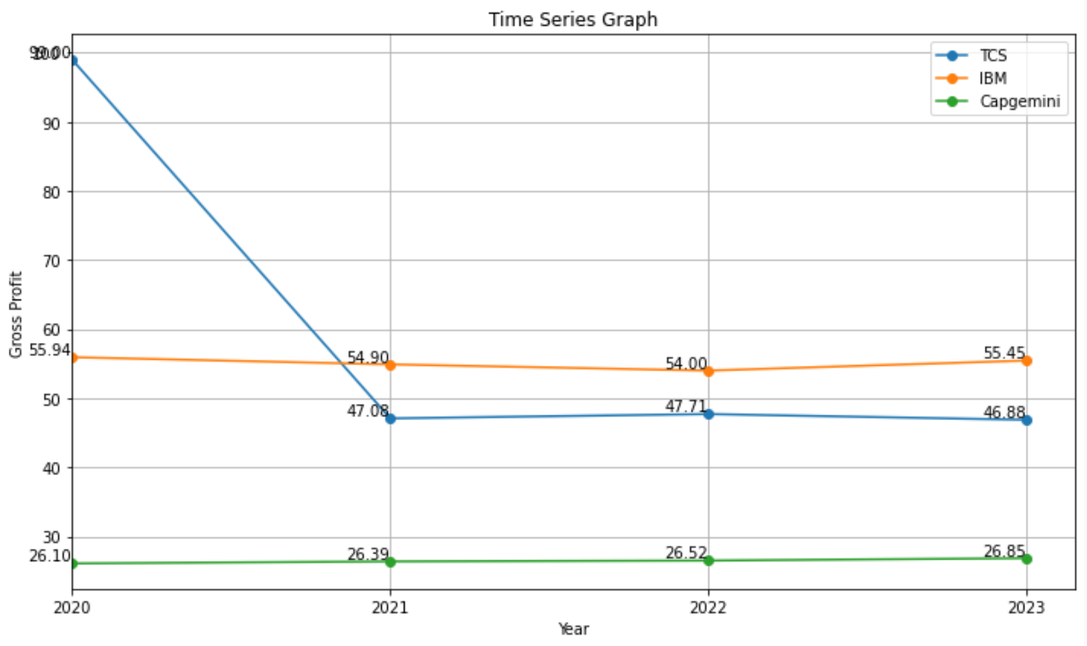
```python
import pandas as pd
import matplotlib.pyplot as plt

# Read values from Excel file
file_path = r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Ibm\ibm_inc.xlsx'
df = pd.read_excel(file_path, header=None)

# Extract elements data
elements_data = df.iloc[110:124, 1:5]

# Define labels for the elements
element_labels = ['Net profit margin', 'Gross profit margin', 'ROA',
                  'ROE', 'Current ratio', 'Quick ratio','Debt to Equity Ratio',
                  'Debt Ratio','Revenue Growth rate','Earnings ratio','P/E ratio','P/B ratio',
                  'Divident yield','Inventory turnover ratio','Asset turnover ratio'
]

# Plot the graph
plt.figure(figsize=(10, 6))
for i, row in enumerate(elements_data.index):
    plt.plot(elements_data.loc[row], marker='o', label=element_labels[i])

plt.title('Time Series Graph')
plt.xlabel('Year')
plt.ylabel('Financial metrices of IBM')
plt.xticks(range(5), [2019, 2020, 2021, 2022, 2023])  # Set x-axis labels manually
plt.xlim(left=1)
plt.legend()
plt.grid(True)
```

Comparing metrices for the 3 companies over last 4 years

Revenue Growth



Gross Profit



Python programs for plotting comparison of Revenue growth and Gross profit for the 3 companies

Revenue Growth

```python
import pandas as pd
import matplotlib.pyplot as plt

file_paths = [
    r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Tata\Tata_Inc.xlsx',
    r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Ibm\ibm_inc.xlsx',
    r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Capgemini\Cap_Inc.xlsx']
# Initialize empty lists to store elements data and labels
elements_data_list = []
element_labels = []

# Read data from each Excel file and store elements data and labels
for i, file_path in enumerate(file_paths):
    df = pd.read_excel(file_path, header=None)
    if i == 0:
        row_index = 103  # Read 1st row from the first Excel file
    elif i == 1:
        row_index = 110  # Read 2nd row from the second Excel file
    else:
        row_index = 112  # Read 7th row from the third Excel file
    elements_data_list.append(df.iloc[row_index, 1:5])  # Extract data from the specified row
    element_labels.append(f'File {i+1}')

element_labels = ['TCS', 'IBM', 'Capgemini']
# Plot the graph
plt.figure(figsize=(10, 6))
for i, elements_data in enumerate(elements_data_list):
    plt.plot(elements_data, marker='o', label=element_labels[i])
    for x, y in zip(range(1, len(elements_data) + 1), elements_data):
       plt.text(x, y, f'{y:.2f}', ha='right', va='bottom', fontsize=10)
plt.title('Time Series Graph')
plt.xlabel('Year')
plt.ylabel('Gross Profit')
plt.xticks(range(5), [2019, 2020, 2021, 2022, 2023])  # Set x-axis labels manually
plt.xlim(left=1)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

Gross Profit

```python
import pandas as pd
import matplotlib.pyplot as plt

file_paths = [
    r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Tata\Tata_Inc.xlsx',
    r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Ibm\ibm_inc.xlsx',
    r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Capgemini\Cap_Inc.xlsx']
# Initialize empty lists to store elements data and labels
elements_data_list = []
element_labels = []
# Read data from each Excel file and store elements data and labels
for i, file_path in enumerate(file_paths):
    df = pd.read_excel(file_path, header=None)
    if i == 0:
        row_index = 110  # Read 1st row from the first Excel file
    elif i == 1:
        row_index = 117  # Read 2nd row from the second Excel file
    else:
        row_index = 119  # Read 7th row from the third Excel file
    elements_data_list.append(df.iloc[row_index, 1:5])  # Extract data from the specified row
    element_labels.append(f'File {i+1}')

element_labels = ['TCS', 'IBM', 'Capgemini']
# Plot the graph
plt.figure(figsize=(10, 6))
for i, elements_data in enumerate(elements_data_list):
    plt.plot(elements_data, marker='o', label=element_labels[i])
    for x, y in zip(range(1, len(elements_data) + 1), elements_data):
       plt.text(x, y, f'{y:.2f}', ha='right', va='bottom', fontsize=10)

plt.title('Time Series Graph')
plt.xlabel('Year')
plt.ylabel('Revenue Growth')
plt.xticks(range(5), [2019, 2020, 2021, 2022, 2023])  # Set x-axis labels manually
plt.xlim(left=1)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

Conclusions of the financial analysis

Profitability Ratio:

Highest: IBM is doing well with an average of **55 %** gross profit margin
Lowest: Capgemini with 25% average gross profit margin.
Remarks: For TCS, the average Gross profit margin is **46%**.

Liquidity Ratio:
Highest: TCS is doing well in this domain with an average **2.6** rating in current ratio.
Lowest: IBM scores lowest with an average **0.8** rating in current ratio**.**
Remarks: For Capgemini, the average rating is **1.0.**

Solvency Ratio:
Highest : IBM's average debt ratio is **0.4%** which is higher compared to others**.**
Lowest: Average debt ratio for TCS is **0.04**% which is the lowest among all three**.**
Remarks: Capgemini's average debt ratio is **0.27%**

Growth Ratios:
Highest : TCS's revenue growth rate is **12%** average which is highest among the others.
Lowest: Capgemini's revenue growth rate decreased in 2023 which was just **2.40%**
Remarks: IBM has a slow revenue growth rate which was just **2.20%** in 2023.

Valuation Ratios:
Highest: Price to earnings ratio was highest for IBM in 2022 which was **60%**
Lowest: nil
Remarks: Price to earnings ratio for TCS and Capgemini has been quite consistent which was somewhere around **26%** for TCS and **20%** for Capgemini.

Efficiency Ratios
Highest: Dividend yield for IBM has been comparatively high but dropped down to **1.91%** in 2023**.**
Lowest: Dividend yield for TATA has been low which is **0.26%** average.
Remarks: Dividend yield for Capgemini has been consistent which is **1.3%**

Summary:

- **TCS:**
  - Demonstrates strong liquidity
  - Exhibits conservative debt management
  - Has the highest revenue growth rate among the companies analyzed
- **IBM:**
  - Demonstrates strong profitability and potential valuation
  - Faces challenges in liquidity
  - Experiences slower revenue growth compared to TCS
- **Capgemini:**
  - Maintains stable performance in liquidity, debt ratio, and valuation
  - Experiences lower profitability compared to IBM and TCS
  - Faces a decrease in revenue growth

Forecasting values for the next 2 years (2024 and 2025)

- Utilizing **Linear regression** for forecasting future values.
- Linear regression is a statistical method used for modeling the relationship between a dependent variable and one or more independent variables.
- In this context, linear regression is employed to model the relationship between the years (independent variable) and the elements' values (dependent variable) extracted from the dataset.
- For each element, the program fits a linear regression model using the years as independent variables and the element's values as dependent variables.
- After fitting the model, the program predicts future values for the specified number of years beyond the last observed year in the dataset.

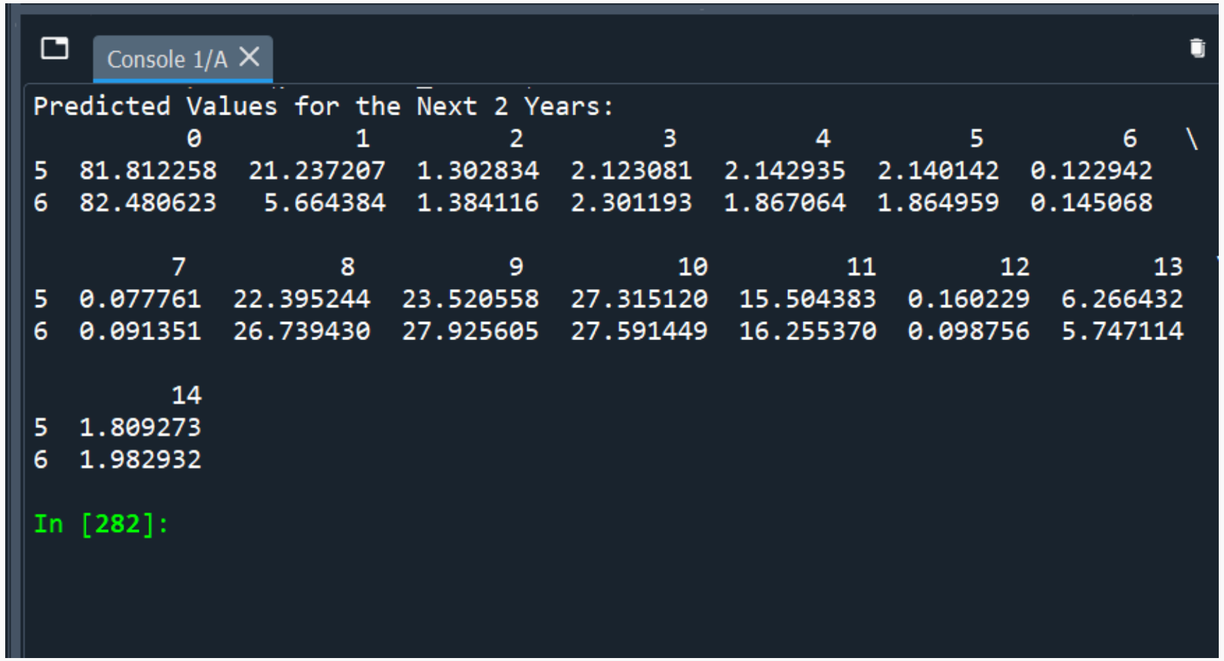
Python Program

```python
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.impute import SimpleImputer

def read_data(file_path, sheet_name, start_row, end_row, start_col, end_col):
    """
    Read data from specific rows and columns in an Excel file.
    Args:
        file_path (str): Path to the Excel file.
        sheet_name (str): Name of the sheet containing the data.
        start_row (int): Row number to start reading from.
        end_row (int): Row number to end reading at.
        start_col (int): Column number to start reading from.
        end_col (int): Column number to end reading at.
    Returns:
        pandas.DataFrame: DataFrame containing the read data.
    """
    # Read data from Excel file and convert non-numeric values to NaN
    data = pd.read_excel(file_path, sheet_name=sheet_name, header=None, skiprows=start_row-1,
                         nrows=end_row-start_row+1, usecols=range(start_col-1, end_col))
    return data

def predict_future_values(data, years_to_predict):
    """
    Predict future values using linear regression.
    Args:
        data (pandas.DataFrame): DataFrame containing the data.
        years_to_predict (int): Number of years to predict.
    Returns:
        pandas.DataFrame: DataFrame containing the predicted values.
    """
    # Extract years and elements
    years = data.columns.values
    elements = data.values

    # Initialize DataFrame for predicted values
    predicted_values = pd.DataFrame(index=range(years[-1] + 1, years[-1] + 1 + years_to_predict))

    # Perform linear regression for each element
    for i, element in enumerate(data.index):
        # Extract values for the element and handle NaN values
        values = elements[i].reshape(-1, 1)

        # Check if values are numeric
        if pd.api.types.is_numeric_dtype(values.flatten()):
            # Initialize imputer to handle missing values
            imputer = SimpleImputer(strategy='mean')
            values_imputed = imputer.fit_transform(values)

            # Initialize linear regression model
            model = LinearRegression()

            # Fit the model
            model.fit([[year] for year in years], values_imputed)

            # Predict future values
            future_years = [year for year in range(years[-1] + 1, years[-1] + 1 + years_to_predict)]
            future_values = model.predict([[year] for year in future_years])

            # Add predicted values to DataFrame
            predicted_values[element] = future_values.flatten()
        else:
            # If the values are not numeric, fill with NaNs
            predicted_values[element] = [None] * years_to_predict

    return predicted_values

# Usage
file_path = r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Tata\Tata_Inc.xlsx'
sheet_name = 'Income statement'
start_row = 103
end_row = 117
start_col = 2
end_col = 5
years_to_predict = 2

# Read data from Excel file
data = read_data(file_path, sheet_name, start_row, end_row, start_col, end_col)

# Predict future values
predicted_values = predict_future_values(data, years_to_predict)

predicted_values_transposed = predicted_values.transpose()


pd.set_option('display.max_columns', None)

# Print predicted values
print("Predicted Values for the Next 2 Years:")
print(predicted_values)
```

Forecasting values for the next 2 years

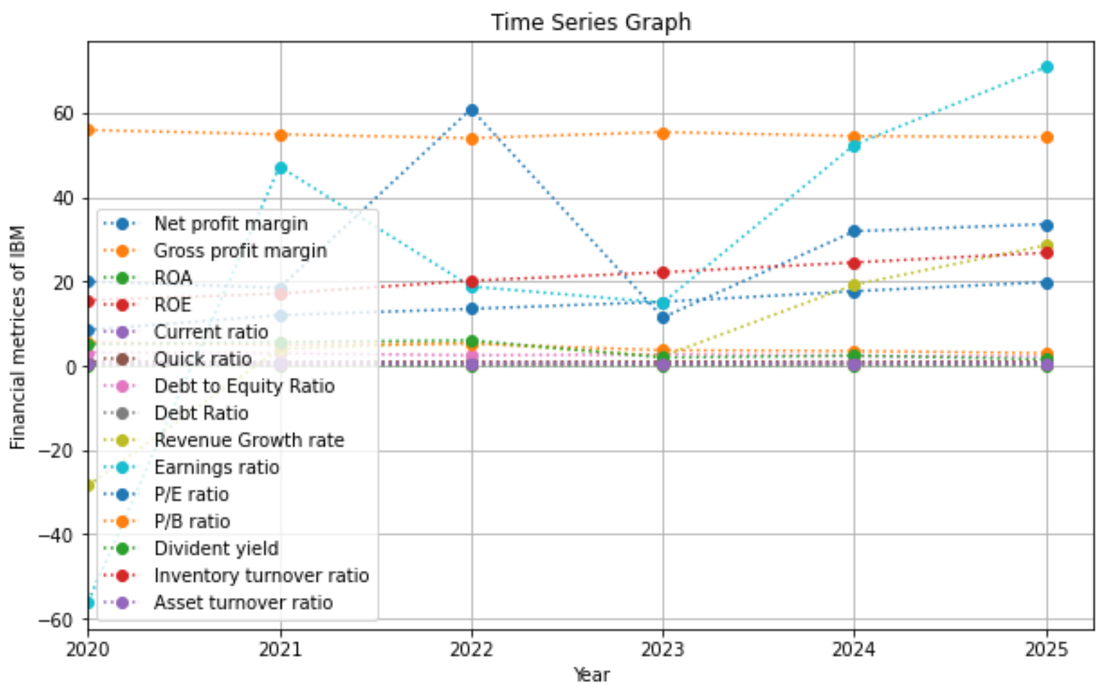
- The program takes as input the file path to the Excel file, the sheet name containing the data, the range of rows and columns containing the historical data, and the number of years to predict future values.

- It reads the data from the Excel file, performs forecasting using linear regression, and prints the predicted values for the specified number of years.

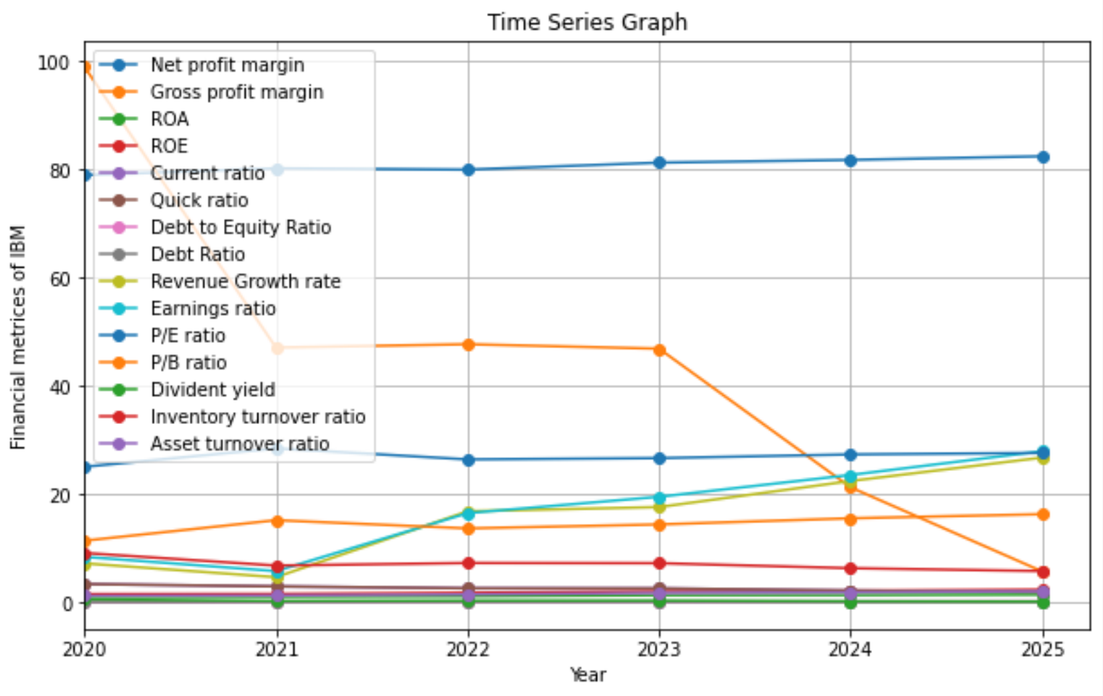


Forecasted values

IBM

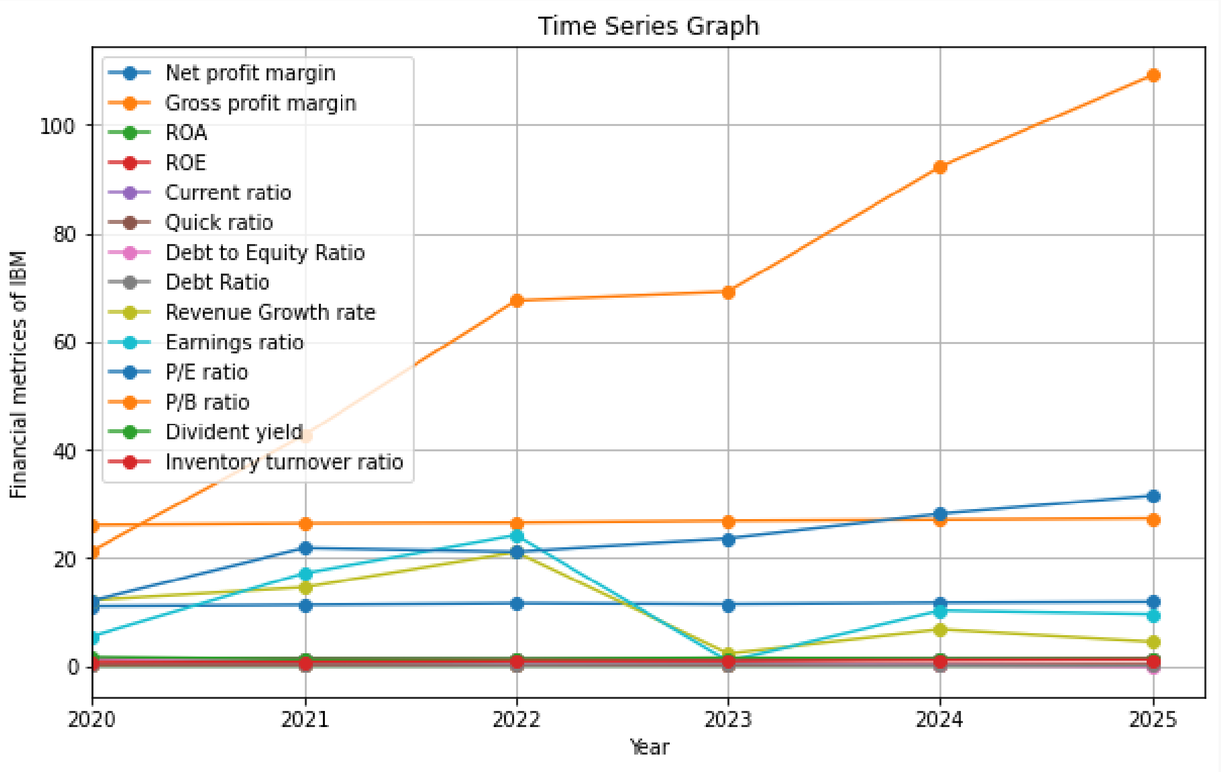


TCS



Forecasted values

Capgemini



Findings from the forecasted elements

- Tcs’s gross profit margin is expected to have a drastic fall i.e. from 99% in 2020 to 5.6% in 2025 however the company’s revenue growth rate and earning ratio is expected to rise. Despite the decline in gross profit margin, the expected rise in revenue growth rate indicates that TCS is still able to increase its overall sales volume or expand its market share. Investors may react differently to these trends. While some may view the decline in gross profit margin negatively due to its impact on profitability, others may focus on the positive aspects such as revenue growth and improving earnings performance.

- For IBM both net profit and gross profit margin is expected to rise in the coming years. This could suggest that investors are optimistic about the company's future growth prospects or performance.

- Capgemini's P/B ratio despite unchanged net and gross profit levels implies that investors are placing a higher valuation on the company's book value and are optimistic about its future growth potential and overall performance.

Python Program to plot the forecasted values

```python
#code to plot forecasted values for ibm.....................................................

import pandas as pd
import matplotlib.pyplot as plt

# Read values from Excel file
file_path = r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Ibm\ibm_inc.xlsx'
df = pd.read_excel(file_path, header=None)

# Extract elements data
elements_data = df.iloc[109:124, 1:7]

# Define labels for the elements
element_labels = ['Net profit margin', 'Gross profit margin', 'ROA', 'ROE',
                  'Current ratio', 'Quick ratio','Debt to Equity Ratio','Debt Ratio',
                  'Revenue Growth rate','Earnings ratio','P/E ratio','P/B ratio','Divident yield',
                  'Inventory turnover ratio','Asset turnover ratio'
]

# Check if the value in row 109 is 2024 or 2025
for i in [0,6]:
    if df.iloc[108, i] in [2024, 2025]:
        line_style = ':'
    else:
        line_style = '-'

# Plot the graph
plt.figure(figsize=(10, 6))
for i, row in enumerate(elements_data.index):
    plt.plot(elements_data.loc[row], marker='o', linestyle=line_style, label=element_labels[i])

plt.title('Time Series Graph')
plt.xlabel('Year')
plt.ylabel('Financial metrices of IBM')
plt.xticks(range(7), [2019, 2020, 2021, 2022, 2023, 2024, 2025])  # Set x-axis labels manually

plt.xlim(left=1)
plt.legend()
plt.grid(True)
plt.show()
```

Environment, Social and Governance (ESG) Risk Ratings

Environmental Factors: These factors focus on a company's impact on the environment. Key considerations include:

- Carbon footprint and greenhouse gas emissions
- Energy efficiency and renewable energy use
- Waste management and recycling practices
- Water usage and conservation efforts
- Environmental regulations compliance
- Biodiversity conservation

Social Factors: Social factors pertain to how a company manages relationships with its employees, customers, suppliers, and the communities in which it operates. This includes:

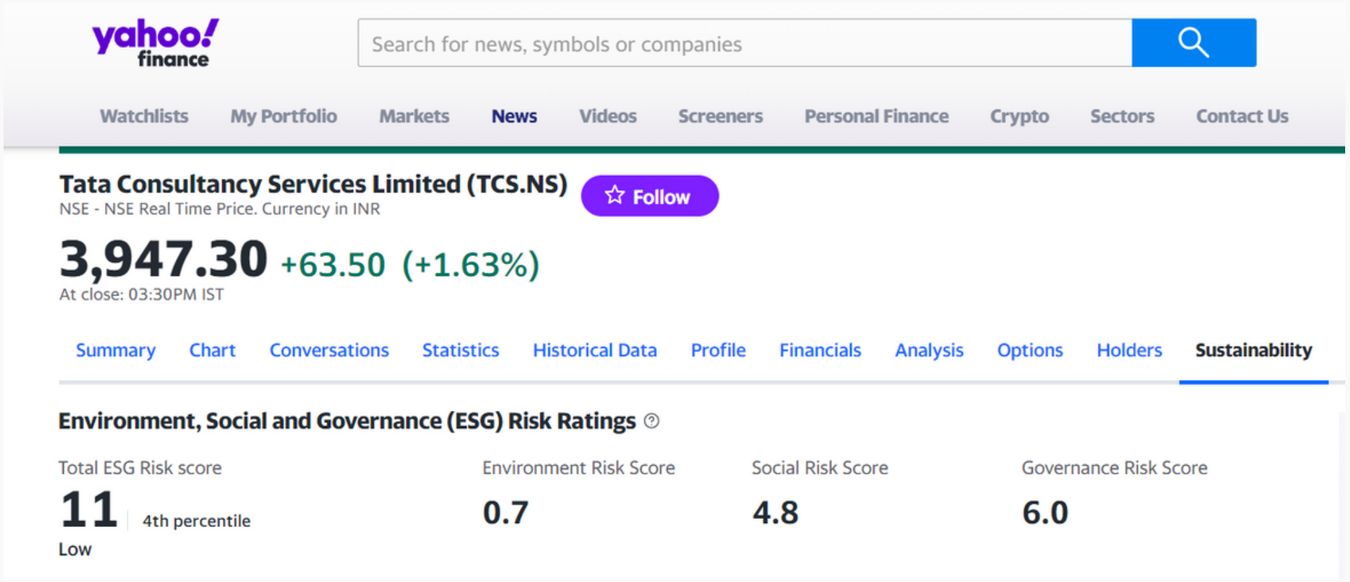
- Labor practices and human rights
- Workplace diversity, inclusion, and safety
- Employee benefits and working conditions
- Customer satisfaction and product safety
- Community engagement and philanthropy
- Supply chain management and ethical sourcing

Governance Factors: Governance factors refer to the way a company is governed, including its leadership structure, policies, and practices. This encompasses:

- Board diversity, independence, and expertise
- Executive compensation and incentives
- Transparency and disclosure practices
- Shareholder rights and activism
- Ethical business conduct and anti-corruption measures
- Risk management and compliance frameworks

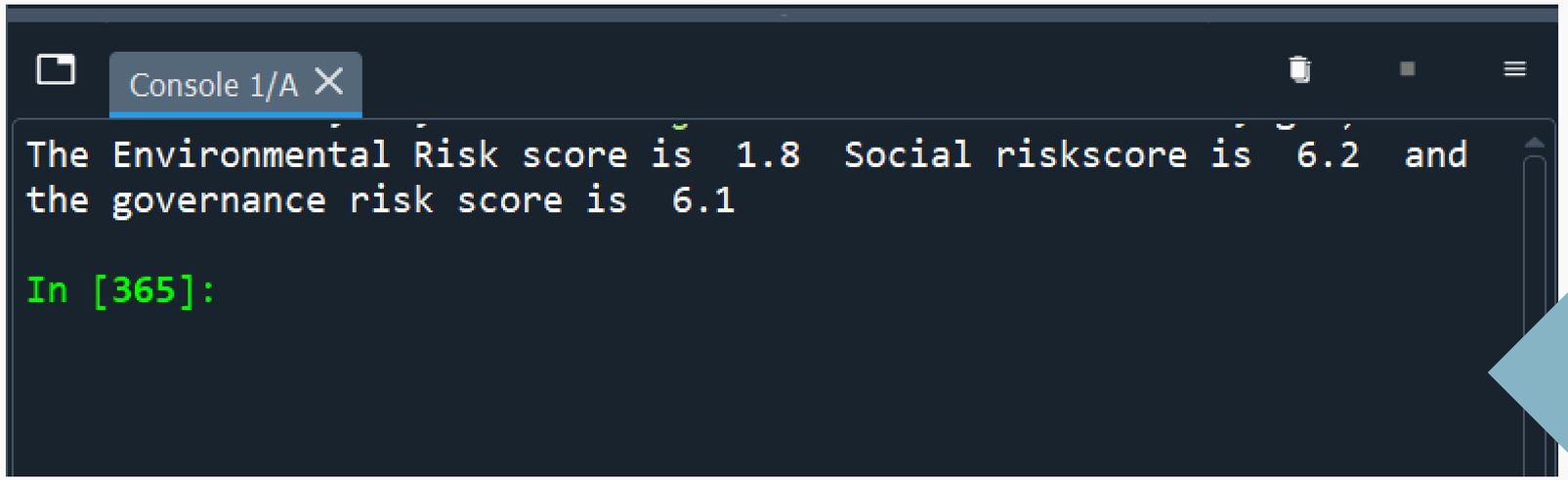
Extracting ESG factors

Source



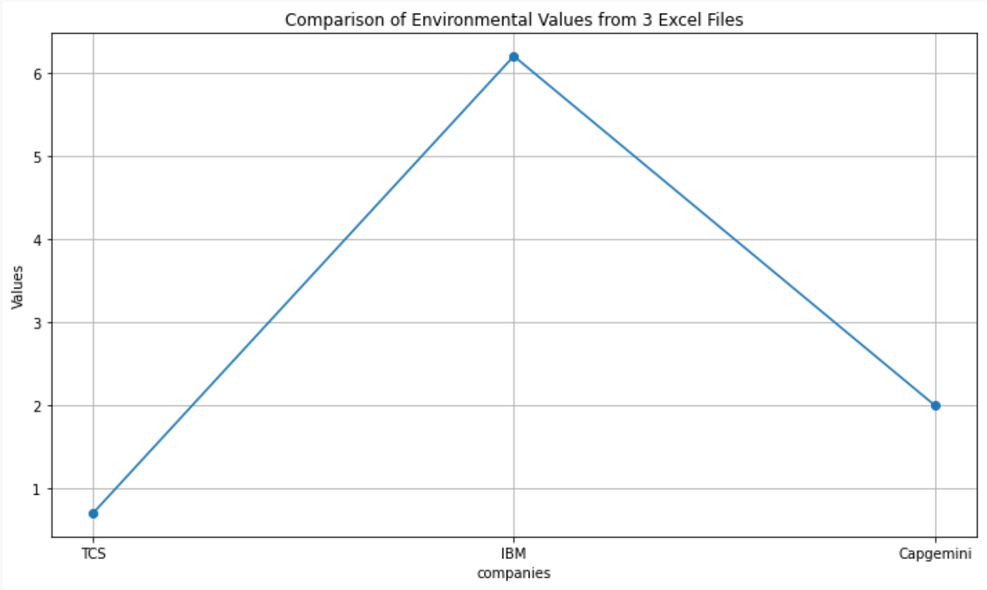
Input

```python
import pandas as pd
# Read data from Excel file
file_path = r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Ibm\ibm_inc.xlsx'
df = pd.read_excel(file_path, sheet_name='Income statement')

# Extract esg values from the excel
Env = df.iloc[126, 1]
soc = df.iloc[127, 1]
gov = df.iloc[128, 1]

print("The Environmental Risk score is ", Env, " Social riskscore is ",soc,
      " and the governance risk score is ", gov)
```

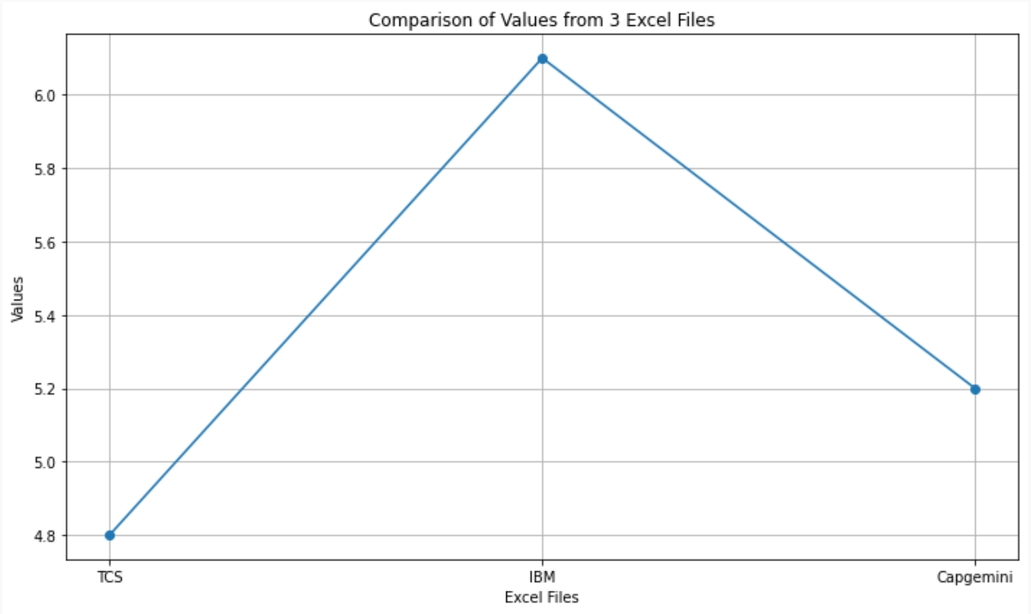


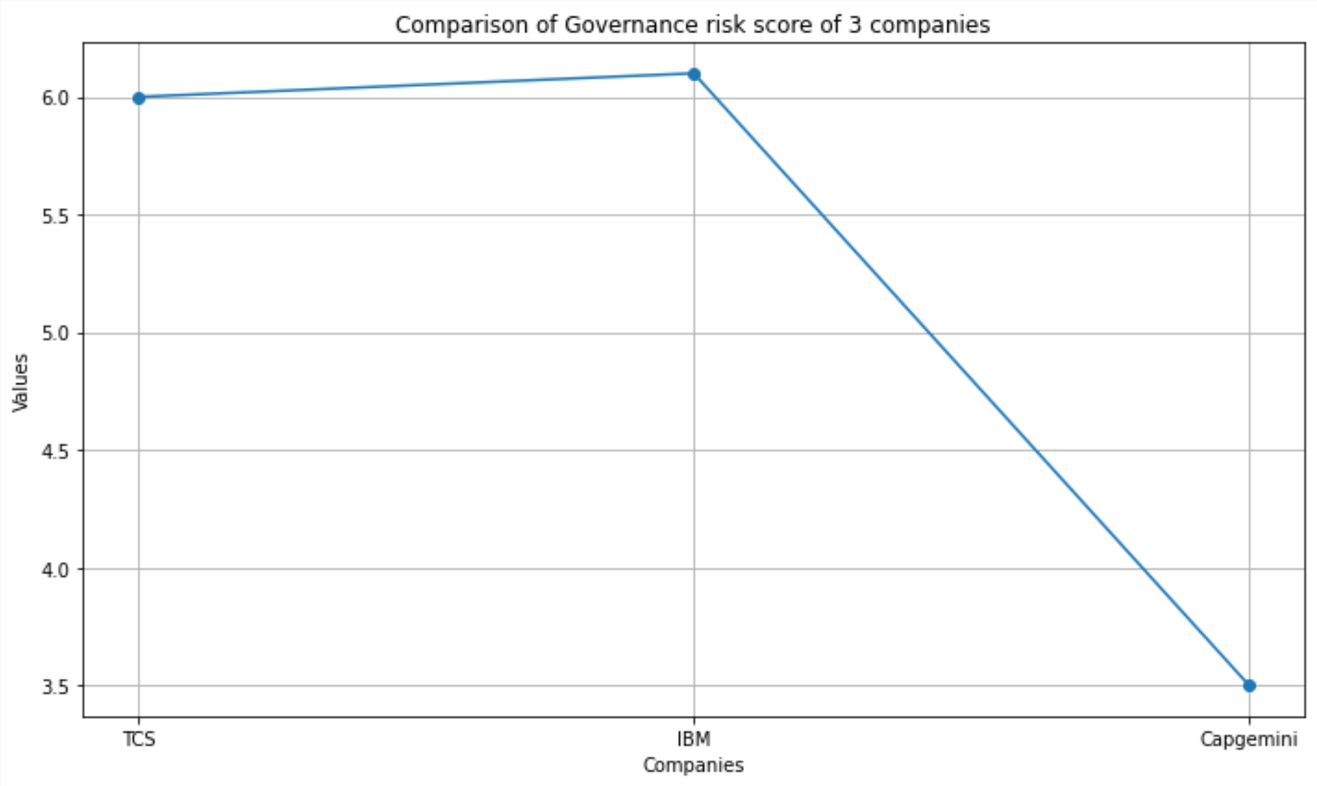
Output

Environmental risk



Social risk





Governance Risk risk

Python Program for plotting ESG ratings

```python
# Plotting graph for governance risk score
# Read values from Excel files
import pandas as pd
import matplotlib.pyplot as plt

file_paths = [
    r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Tata\Tata_Inc.xlsx',
    r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Ibm\ibm_inc.xlsx',
    r'C:\Users\Mukul Kapri\Desktop\KDP Projects\Python assignment\Excels\Capgemini\Cap_Inc.xlsx'
]

# Cell coordinates for each file
cell_coordinates = [
    (122, 1),   # Cell coordinates for the first file (row index, column index)
    (130, 1),   # Cell coordinates for the second file (row index, column index)
    (130, 1)   # Cell coordinates for the third file (row index, column index)
]

# Initialize lists to store the values from each Excel file
values = []

# Read data from the specified cells in each Excel file
for file_path, (row_index, col_index) in zip(file_paths, cell_coordinates):
    df = pd.read_excel(file_path, header=None)
    value = df.iloc[row_index, col_index]
    values.append(value)

# Plot the graph
plt.figure(figsize=(10, 6))
plt.plot(values, marker='o', linestyle='-')
plt.title('Comparison of Governance risk score of 3 companies')
plt.xlabel('Companies')
plt.ylabel('Values')
plt.xticks(range(3), ['TCS', 'IBM', 'Capgemini'])  # Set x-axis labels manually
plt.grid(True)
plt.tight_layout()
plt.show()
```

Relations between ESG ratings and Financial metrices

IBM

Forecasting- Time series Analysis	2024	2025
Net profit Margin	17.67	19.82
Gross profit Margin	54.49	54.24
Return on Assets(ROA)	0.087	0.1
Return on Equity (roe)	0.49	0.54
Current Ratio	0.94	0.93
Quick ratio	0.9	0.9
Debt to Equity Ratio	2.3	2.11
Debt Ratio	0.44	0.46
Revenue Growth rate	19.2	28.57
Earnings ratio	52.4	70.89
P/E ratio	31.9	33.59
P/B ratio	3.47	2.94
Divident yield	2.39	1.49
Inventory turnover ratio	24.49	26.8
Asset turnover ratio	0.47	0.49

ESG Factors	Values
Environmental	1.8
Social	6.2
Governmental	6.1

Higher the ESC risk factors, especially Social and Governance risks, it might negatively affect several financial elements like Net profit margin, Profit ratios, dividend yield etc.

Relations between ESG ratings and Financial metrices

TCS

Forecasting- Time series Analysis	2024	2025
Net profit Margin	81.81	82.48
Gross profit Margin	21.23	5.67
Return on Assets(ROA)	1.3	1.38
Return on Equity (roe)	2.12	2.3
Current Ratio	2.14	1.87
Quick ratio	2.14	1.87
Debt to Equity Ratio	0.12	0.15
Debt Ratio	0.07	0.09
Revenue Growth rate	22.4	26.73
Earnings ratio	23.52	27.92
P/E ratio	27.32	27.59
P/B ratio	15.5	16.26
Divident yield	0.16	0.1
Inventory turnover ratio	6.27	5.74
Asset turnover ratio	1.8	1.98

ESC Factors	Values
Environment Risk Score	0.7
Social Risk Score	4.8
Governance Risk Score	6

Lower the ESC risk factors, especially Social and Governance risks, it might positively affect several financial elements like Net profit margin, Profit ratios, dividend yield etc.

Relations between ESG ratings and Financial metrices

- A deterioration in social and governmental risk factors may negatively affect Capgemini's reputation among stakeholders, including customers, investors, employees, and regulators.
- Increasing social and governmental risks may create operational challenges for IBM, such as regulatory compliance issues, legal disputes, supply chain disruptions, or labor unrest.
- Regulatory fines, legal settlements, or penalties associated with non-compliance can negatively impact profitability. Investors may perceive IBM as a riskier investment due to the worsening social and governmental risk factors.
- On the other hand, Low social and governmental risk factors indicate a favorable operating environment for TCS. A stable operating environment and favorable social and governmental risk factors contribute to building trust and credibility with TCS's customers.
- Overall, an increase in social and governmental risk factors may lead to poor financial performance while low social and governmental risk factors contributing to good or stable financials


THANK YOU!