Financial Analytics

Domain : Finance

Introduction

Analyzing competition helps businesses identify their strengths and weaknesses, uncover opportunities for growth, and devise strategic plans to enhance their market position. This report focuses on analyzing the competition among the top 500 companies in India based on their market capitalization and quarterly sales. The dataset provides comprehensive information on these companies, including their market capitalization in crores and their quarterly sales in crores.

Data Description

df.describe()

```
df.describe()
```

	S.No.	Mar Cap - Crore	Sales Qtr - Crore
count	488.000000	479.000000	365.000000
mean	251.508197	28043.857119	4395.976849
std	145.884078	59464.615831	11092.206185
min	1.000000	3017.070000	47.240000
25%	122.750000	4843.575000	593.740000
50%	252.500000	9885.050000	1278.300000
75%	378.250000	23549.900000	2840.750000
max	500.000000	583436.720000	110666.930000

Python coding on Financial Analytics

Import the packages

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

Set the path for better visibility

```
import os

for dirname, _, filenames in os.walk(r"C:\Users\sabir\Downloads\Financial Analytics
data.csv"):

    for filename in filenames:

        print(os.path.join(dirname, filename))
```

Setting the style for Seaborn plots

```
sns.set_style('darkgrid')
```

Reading the file

```
df = pd.read_csv(r"C:\Users\sabir\Downloads\Financial Analytics data.csv")
```

Reading the top 5 files

```
df.head()
```

```
df.head()
```

	S.No.	Name	Mar Cap - Crore	Sales Qtr - Crore	Unnamed: 4
0	1	Reliance Inds.	583436.72	99810.00	NaN
1	2	TCS	563709.84	30904.00	NaN
2	3	HDFC Bank	482953.59	20581.27	NaN
3	4	ITC	320985.27	9772.02	NaN
4	5	H D F C	289497.37	16840.51	NaN

Shape of the file

```
df.shape
```

```
df.shape
```

```
(488, 5)
```

Information of the data set

df.info()

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 488 entries, 0 to 487
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   S.No.             488 non-null    int64
 1   Name              488 non-null    object
 2   Mar Cap - Crore   479 non-null    float64
 3   Sales Qtr - Crore 365 non-null    float64
 4   Unnamed: 4        94 non-null     float64
dtypes: float64(3), int64(1), object(1)
memory usage: 19.2+ KB
```

Finding out the missing values

df.isna().sum()

```
df.isna().sum()
```

```
S.No.               0
Name                0
Mar Cap - Crore     9
Sales Qtr - Crore   123
Unnamed: 4          394
dtype: int64
```

Drop the unwanted column

df=df.drop(['Unnamed: 4'],axis = 1)

Finding the proportion of missing values

df.isna().mean()

```
S.No.              0.000000
Name               0.000000
Mar Cap - Crore    0.018443
Sales Qtr - Crore  0.252049
dtype: float64
```

Replacing the missing values with nan

df=df.replace(r'^\s*$',np.nan,regex = True)

Set the plot size for better visibility

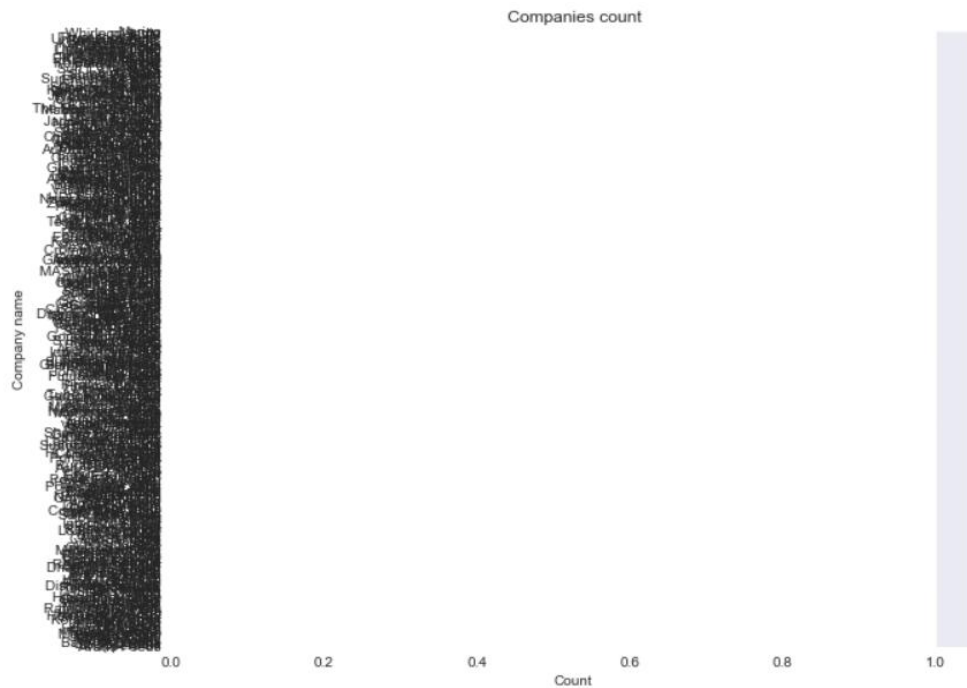plt.figure(figsize=(10, 8))

Create a countplot for the 'Name' column

sns.countplot(y=data['Name'], order=data['Name'].value_counts().index)

plt.title(' Companies count')

plt.xlabel('Count')

plt.ylabel('Company name')

plt.show()

Companies count

Select the top 50 companies
top_50_companies = data.head(50)

Set the plot size for better visibility
plt.figure(figsize=(10, 15))

Create a countplot for the 'Name' column of the top 50 companies
sns.countplot(y=top_50_companies['Name'],
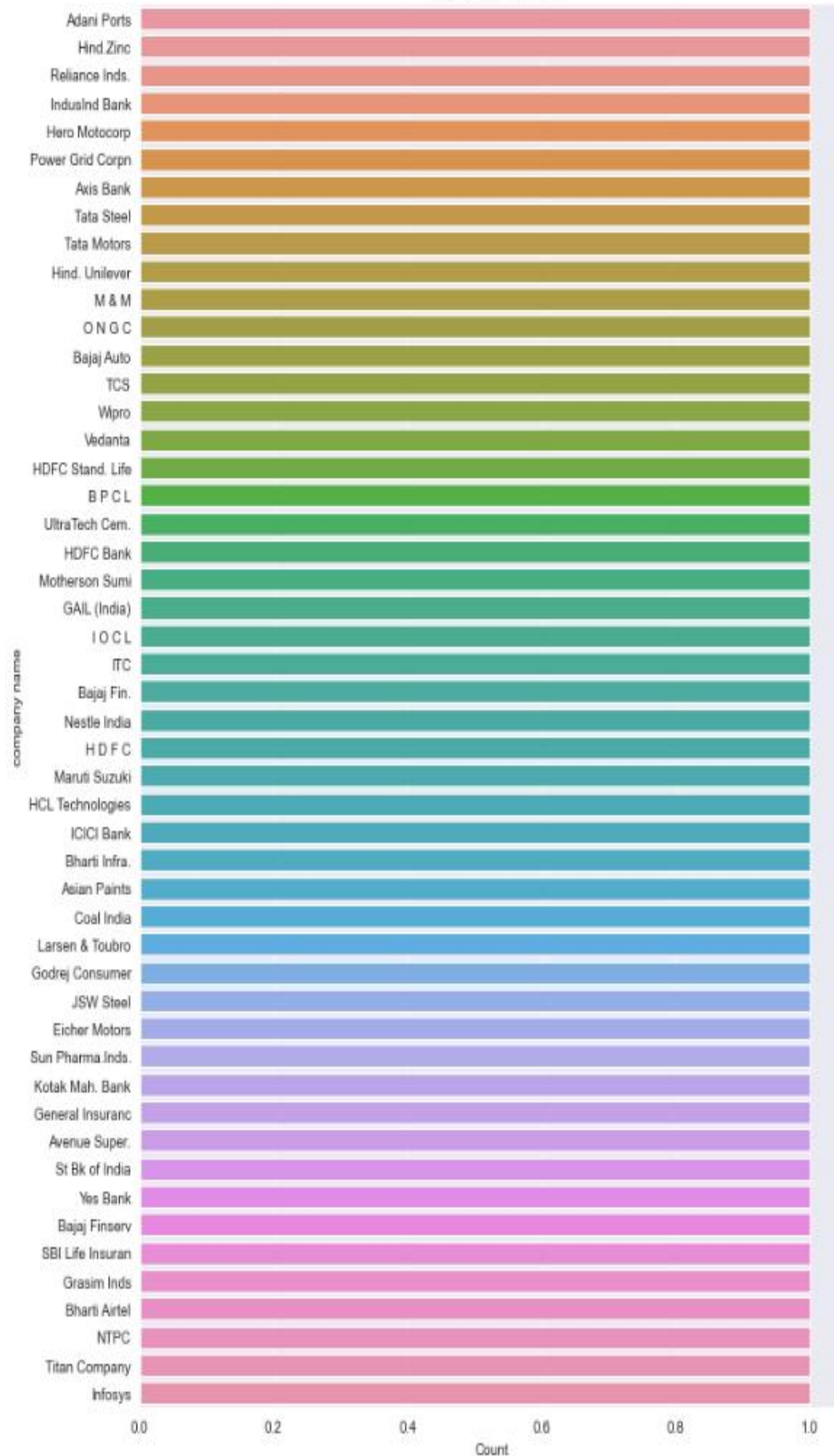order=top_50_companies['Name'].value_counts().index)

plt.title('Top 50 Companies')
plt.xlabel('Count')
plt.ylabel('company name')
plt.show()

Top 50 Companies

Calculate the difference between Mar Cap and Sales Qtr

```
df['Difference'] = df['Mar Cap - Crore'] - df['Sales Qtr - Crore']
plt.figure(figsize=(18, 6))
```

Bar plot for market capitalization, sales, and their difference
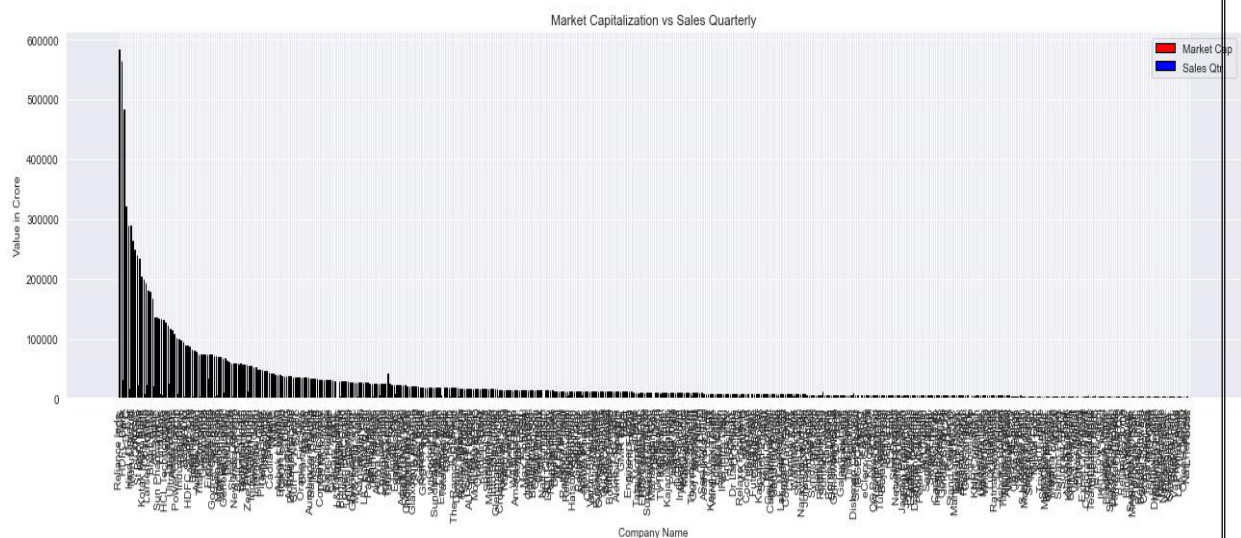
```
width = 0.3
```

Bar plot for Market Cap

```
plt.bar(df['Name'], df['Mar Cap - Crore'], width=width, label='Market Cap', align='center',
color='r', edgecolor='black')
```

Bar plot for Sales Qtr

```
plt.bar(df['Name'], df['Sales Qtr - Crore'], width=width, label='Sales Qtr', align='edge',
color='b', edgecolor='black')
```

Rotate x-axis labels for better readability

```
plt.xticks(rotation=90)
plt.xlabel('Company Name')
plt.ylabel('Value in Crore')
plt.title('Market Capitalization vs Sales Quarterly')
plt.legend()
plt.show()
```
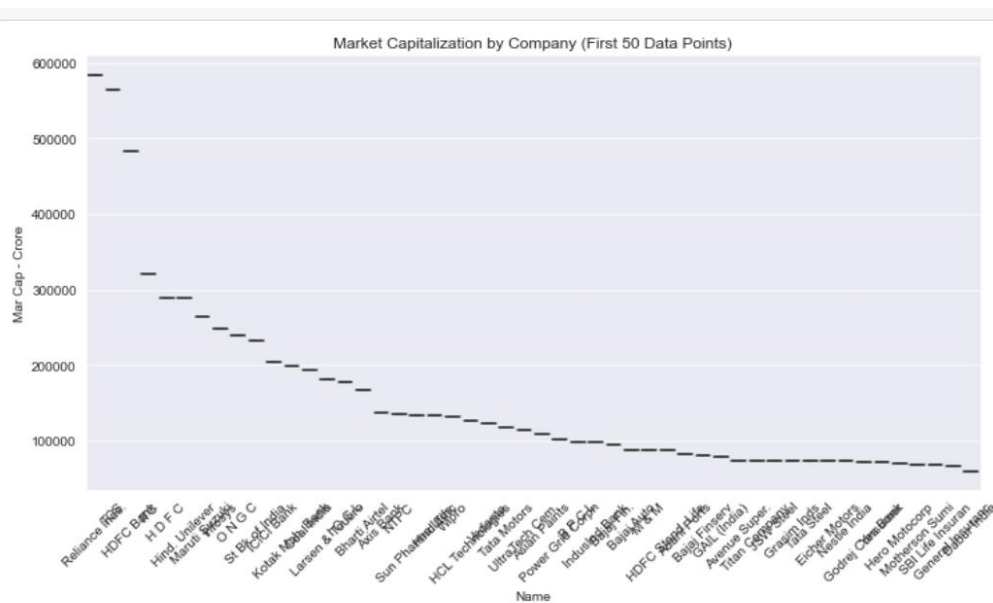
Create the box plot

plt.figure(figsize=(12, 6))

sns.boxplot(x='Name', y='Mar Cap - Crore', data=df_first_50)
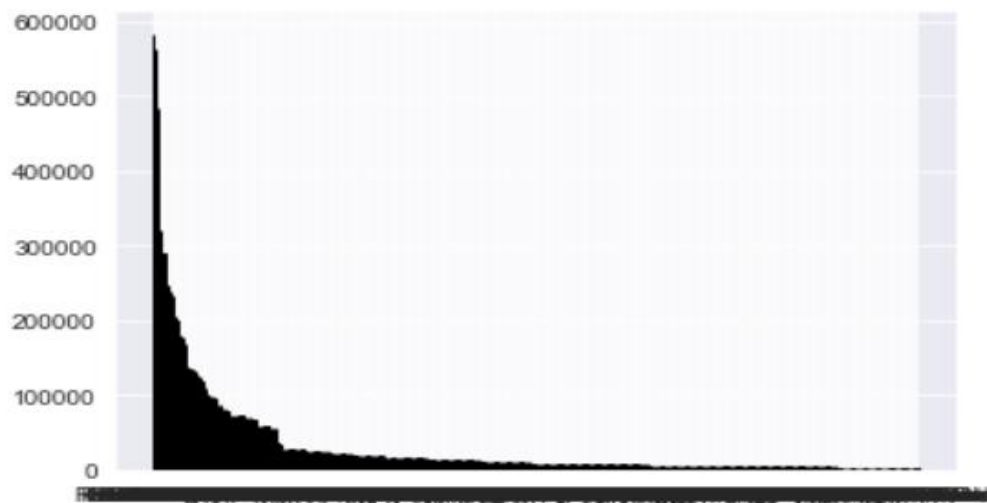
plt.xticks(rotation=45)

plt.title('Market Capitalization by Company (First 50 Data Points)')

plt.show()

plt.bar(df['Name'], df['Mar Cap - Crore'], width=width, label='Market Cap', align='center', color='r', edgecolor='black')

```
<BarContainer object of 365 artists>
```

Sort the DataFrame by 'Mar Cap - Crore' in descending order and take the top 50 companies

```
top_50_df = df.nlargest(50, 'Mar Cap - Crore')

plt.figure(figsize=(14, 6))
```

 # Subplot 1: Bar Plot for Market Cap

```
plt.subplot(1, 2, 1)

sns.barplot(x=top_50_df['Name'], y=top_50_df['Mar Cap - Crore'], palette='viridis')

plt.title('Market Cap by Company (Top 50)')

plt.xlabel('Company Name')

plt.ylabel('Market Cap (Crore)')

plt.xticks(rotation=90)  # Rotate x-axis labels for better readability
```

Subplot 2: Box Plot for Sales

```
plt.subplot(1, 2, 2)

sns.boxplot(x='Name', y='Sales Qtr - Crore', data=top_50_df, palette='plasma')

plt.title('Sales by Company (Top 50)')

plt.xlabel('Company Name')

plt.ylabel('Sales (Crore)')

plt.xticks(rotation=90)  # Rotate x-axis labels for better readability

plt.tight_layout()

plt.show()
```
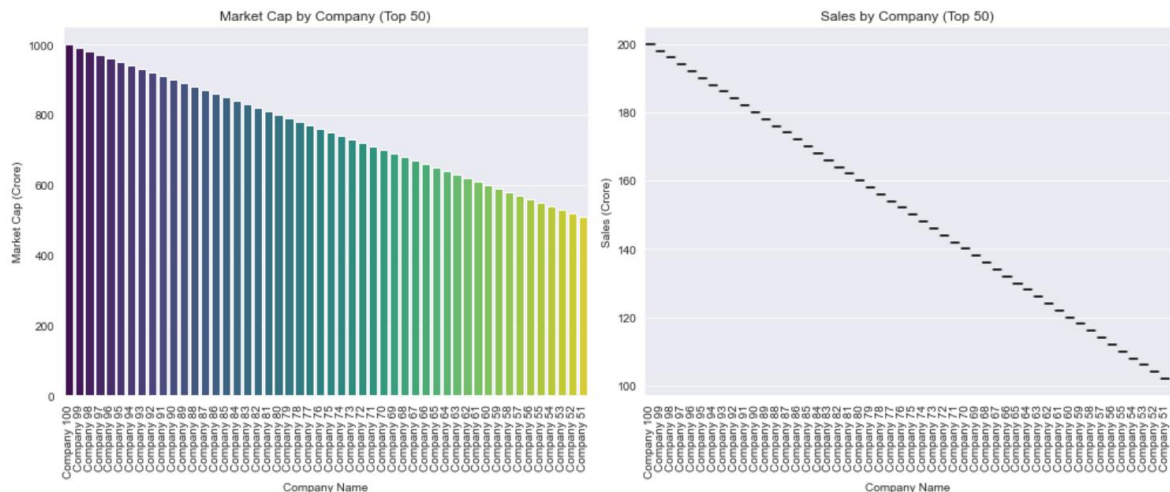
Count of Market Capitalization Categories (assuming Mar Cap - Crore is binned)

plt.subplot(1, 2, 2)

sns.countplot(y=pd.cut(df['Mar Cap - Crore'], bins=10), palette='plasma')

plt.title('Count of Market Capitalization Categories')

plt.tight_layout()

plt.show()

Example data (replace this with your actual data)

data = {

```python
    'Name': [f'Company {i}' for i in range(1, 101)],

    'Mar Cap - Crore': [i * 10 for i in range(1, 101)],

    'Sales Qtr - Crore': [i * 2 for i in range(1, 101)]

}

df = pd.DataFrame(data)

# Sort the DataFrame by 'Mar Cap - Crore' in descending order and take the top 50
companies

top_50_df = df.nlargest(50, 'Mar Cap - Crore')

plt.figure(figsize=(18, 6))


# Subplot 1: Average Market Capitalization by Company

plt.subplot(1, 2, 1)

sns.barplot(x='Name', y='Mar Cap - Crore', data=top_50_df, estimator=np.mean,
palette='viridis', ci=None)

plt.xticks(rotation=90)  # Rotate x-axis labels for better readability

plt.title('Average Market Capitalization by Company')

plt.xlabel('Company Name')

plt.ylabel('Average Market Cap (Crore)')


# Subplot 2: Average Sales per Quarter by Company

plt.subplot(1, 2, 2)

sns.barplot(x='Name', y='Sales Qtr - Crore', data=top_50_df, estimator=np.mean,
palette='plasma', ci=None)

plt.xticks(rotation=90)  # Rotate x-axis labels for better readability

plt.title('Average Sales per Quarter by Company')

plt.xlabel('Company Name')

plt.ylabel('Average Sales (Crore)')

plt.tight_layout()

plt.show()
```
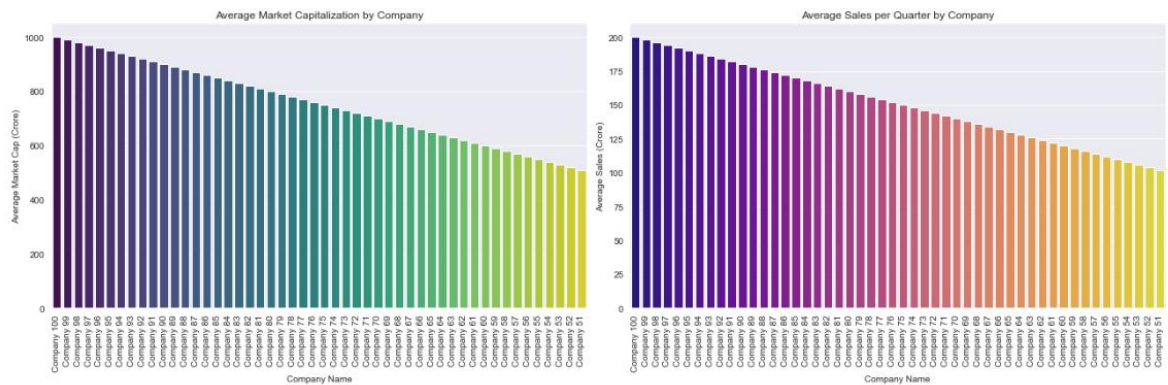
Average Market Capitalization by Company — Average Sales per Quarter by Company

Scatter Plot

```
plt.figure(figsize=(12, 6))

sns.scatterplot(x=df['Mar Cap - Crore'], y=df['Sales Qtr - Crore'], hue=df['Name'],
palette='viridis')

plt.title('Market Capitalization vs. Sales')

plt.xlabel('Market Capitalization (Crore)')

plt.ylabel('Sales per Quarter (Crore)')

plt.legend(loc='upper right')

plt.show()
```
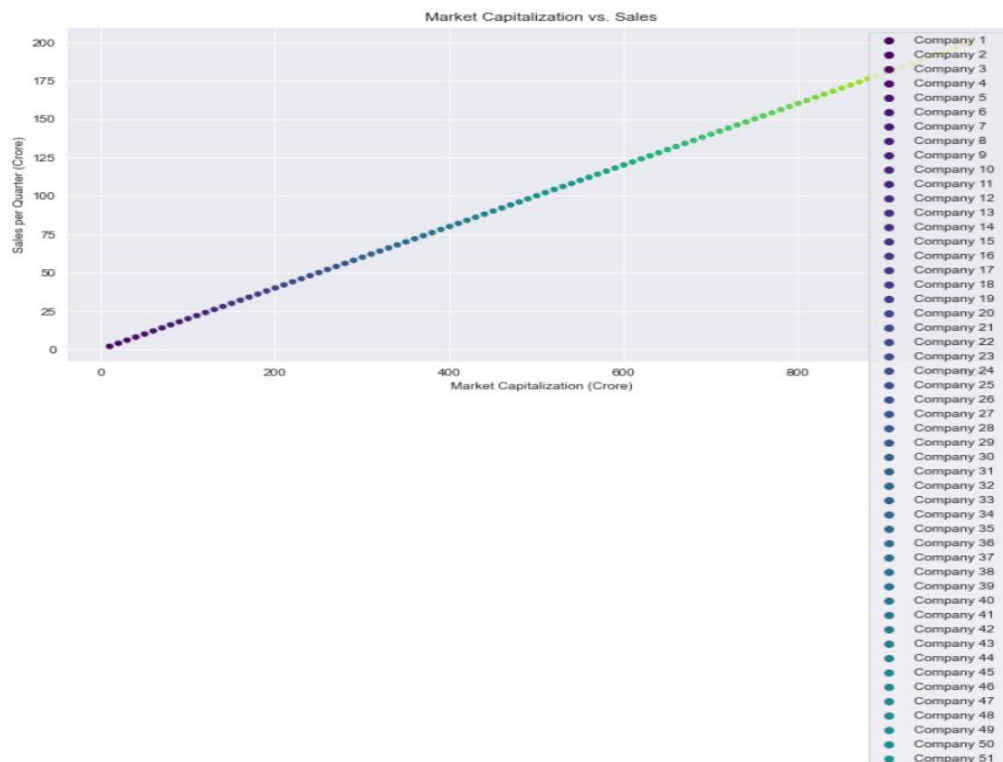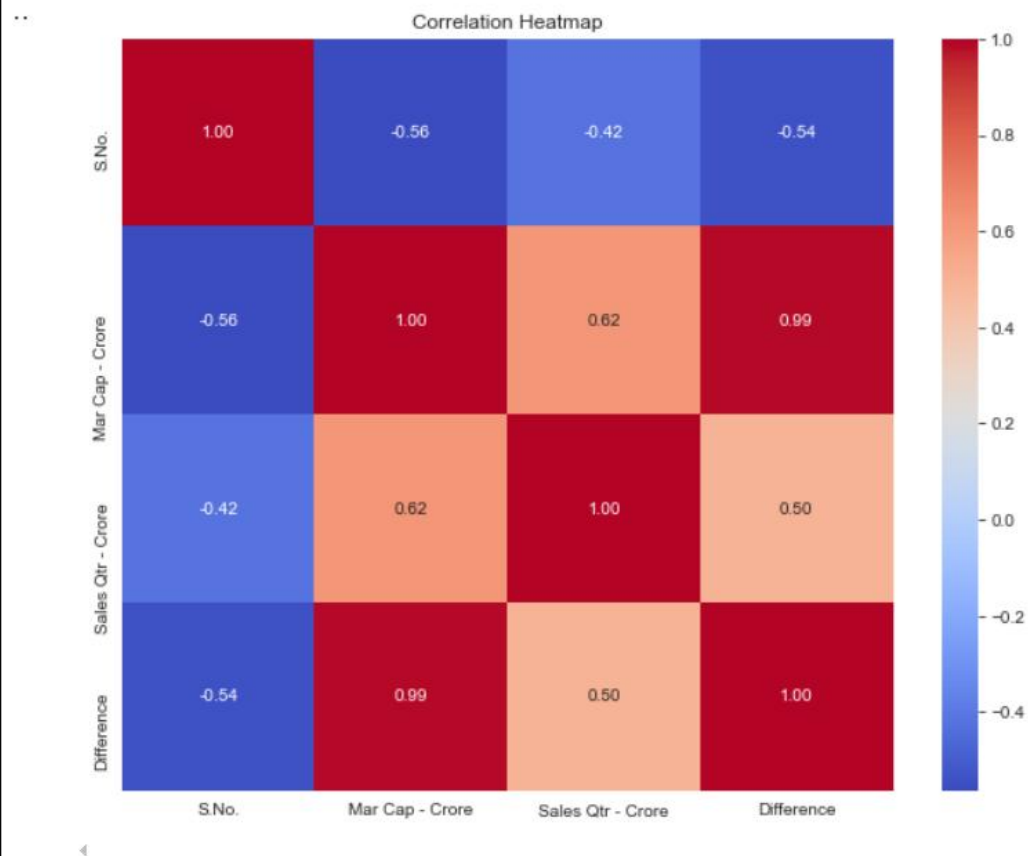

Market Capitalization vs. Sales

Heatmap

```
plt.figure(figsize=(10, 8))

correlation_matrix = df.corr()

sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')

plt.title('Correlation Heatmap')

plt.show()
```

..

Conclusion

This analysis provides a comprehensive overview of the financial metrics of various companies. Key insights include the distribution and comparison of market capitalization and quarterly sales, as well as the correlation between these metrics. Visualizations like count plots, box plots, bar plots, scatter plots, and heatmaps help in understanding the financial standing and performance of the companies in the dataset.