Unified Mentor

Financial Analytics

By Tahir Malik

Problem Statement

Without analysing the competition, it is difficult for a business to survive. You are tasked to analyse the competition for the management to provide better results. This dataset contains information on the market capitalization of the top 500 companies in India.

Serial Number, Name, Name of Company, Mar Cap – Crore (Market Capitalization in Crores), Sales Qtr – Crore (Quarterly Sale in crores) are the columns in the dataset. Find key metrics and factors and show the meaningful relationships between attributes.

Do your own research and come up with your findings.

Introduction

In the ever-evolving landscape of business, the ability to navigate and thrive in a competitive environment is paramount for sustained success. Without a profound understanding of the competition, businesses face the risk of stagnation or even failure. Recognizing this imperative, our analysis aims to delve into the dynamics of competition within the Indian corporate landscape, focusing on the market capitalization of the top 500 companies. The dataset under scrutiny encapsulates critical information such as Serial Number, Company Name, Market Capitalization, and Quarterly Sales. By leveraging analytical methodologies, we aspire to unearth key metrics, discern influential factors, and unravel meaningful relationships between attributes. This investigation is driven by the overarching goal of equipping management with insights that pave the way for informed decision-making, strategic planning, and ultimately, superior business outcomes.

This analysis assumes a holistic perspective, acknowledging the diversity and complexity inherent in the top 500 companies. Through combination of statistical measures, visualizations, and interpretative findings, we aim to deliver a compelling narrative that not only addresses the immediate requirements of the problem statement but also acts as a catalyst for continued exploration and refinement. The report's structure, characterized by modularized and methodical coding practices, is designed to facilitate clarity, reproducibility, and adaptability, ensuring that the analytical journey remains robust and insightful.

Python Code

Data Processing

```
# importing python libraries

import pandas as pd
import matplotlib.pyplot as plt
# importing and reading data stored in dataset-financial-analytics.csv

dataFrame = pd.read_csv("./csv/dataset-financial-analytics.csv")
# printing columns names

dataFrame.columns
# printing dimensions of dataset (rows, columns)
# rows: 488, col: 5

dataFrame.shape
# printing precise summary of our csv

dataFrame.info()
# printing dataset truncated summary

dataFrame
# printing first 10 records

dataFrame.head(10)
# printing bottom 10 records

dataFrame.tail(10)
# printing brief statistical summary of our dataset

dataFrame.describe()
```

Data Cleaning

```
# printing null values of each column

print(f"Null values:")
print(f"[S.No.] = {dataFrame['S.No.'].isnull().sum()}")
print(f"[Name] = {dataFrame['Name'].isnull().sum()}")
print(f"[Mar Cap - Crore] = {dataFrame['Mar Cap -
Crore'].isnull().sum()}")
print(f"[Sales Qtr - Crore] = {dataFrame['Sales Qtr -
Crore'].isnull().sum()}")
print(f"[Unnamed: 4] = {dataFrame['Unnamed: 4'].isnull().sum()}")
# renaming columns

dataFrame.columns = ['S.No.', 'Name of Company', 'Market Cap (in Cr)',
'Quaterly Sales (in Cr)', 'temp-unnamed']
# checking our changes in our dataframe

dataFrame
# I saw we had 488 rows but serial number starts from 1 & ends with 500
# so, there are some values which may got skipped, lets find them out

max_serial_number = dataFrame['S.No.'].max()
all_numbers = set(range(1, max_serial_number + 1))

# get the set of unique numbers present in the column
present_numbers = set(dataFrame['S.No.'])

# find the numbers that are not present in the column
missing_numbers = all_numbers - present_numbers

# print the missing numbers
print("Numbers not present in ['S.No.'] column:", missing_numbers)
# # set display options to show all rows and columns without truncation

# pd.set_option('display.max_rows', None)
# pd.set_option('display.max_columns', None)
dataFrame
# # reset display options to default values

# pd.reset_option('display.max_rows')
# pd.reset_option('display.max_columns')
```

```
# shifting unnamed-column values into Quateryly Sales (in Cr)

for i in range(len(dataFrame)):
    if pd.isnull(dataFrame['temp-unnamed'][i]):
        pass
    else:
        #dataFrame['Sales Qtr - Crore'][i] = dataFrame['temp-
unnamed'][i]
        dataFrame.loc[i, 'Quaterly Sales (in Cr)'] = dataFrame['temp-
unnamed'][i]
```

```
# deleting temp-unnamed column

del dataFrame['temp-unnamed']
del dataFrame['S.No.']
```

```
# lets check quick summary of our dataset

dataFrame.info()
```

```
# printing updated null values of each column

print(f"Null values:")
print(f"[Name of Company] = {dataFrame['Name of
Company'].isnull().sum()}")
print(f"[Market Cap (in Cr)] = {dataFrame['Market Cap (in
Cr)'].isnull().sum()}")
print(f"[Quaterly Sales (in Cr)] = {dataFrame['Quaterly Sales (in
Cr)'].isnull().sum()}")
```

```
print(f"Median for ['Market Cap (in Cr)']: {dataFrame['Market Cap (in
Cr)'].median()}")
print(f"Median for ['Quaterly Sales (in Cr)']: {dataFrame['Quaterly
Sales (in Cr)'].median()}")
```

```
for i in range(len(dataFrame)):
    if pd.isnull(dataFrame['Market Cap (in Cr)'][i]):
        dataFrame.loc[i, 'Market Cap (in Cr)'] = dataFrame['Market Cap
(in Cr)'].median()
```

```
for i in range(len(dataFrame)):
    if pd.isnull(dataFrame['Quaterly Sales (in Cr)'][i]):
        dataFrame.loc[i, 'Quaterly Sales (in Cr)'] =
dataFrame['Quaterly Sales (in Cr)'].median()
```

```
# # reset display options to default values
# pd.reset_option('display.max_rows')
# pd.reset_option('display.max_columns')
```

```
dataFrame.head(10)
```

```
# checking duplicated valus in Name of Company column

duplicate_values = dataFrame.duplicated(subset=['Name of Company'],
keep=False)
print(duplicate_values.sum())
```

```
# striping out spaces from only caps company names

for i in range(len(dataFrame)):
    if dataFrame['Name of Company'][i].isupper():
        dataFrame.loc[i, 'Name of Company'] = dataFrame['Name of
Company'][i].replace(' ', '')
```

```
dataFrame['Name of Company']
```

```
dataFrame
```

```
dataFrame.to_csv('./csv/cleaned-dataset-financial-analytics.csv', index
= False)
```

Data Analysis

```
# calculating basic statistics such as mean, median, minimum, maximum,
and standard deviation for market cap and quarterly sales.
# removing decimals for clean & clear visible records

dataFrame.describe().astype(int)
```

```
sorted_df = dataFrame.sort_values(by='Market Cap (in Cr)',
ascending=False)
print(f"Largest market Cap company: {sorted_df.iloc[0]['Name of
Company']}")
print(f"Market cap: {sorted_df.iloc[0]['Market Cap (in
Cr)'].astype(int)} Cr")
```

```
sorted_df = dataFrame.sort_values(by='Market Cap (in Cr)',
ascending=True)
print(f"Company: {sorted_df.iloc[0]['Name of Company']}")
print(f"Market cap: {sorted_df.iloc[0]['Market Cap (in
Cr)'].astype(int)} Cr")
```

```
sorted_df = dataFrame.sort_values(by='Quaterly Sales (in Cr)',
ascending=False)
print(f"Company: {sorted_df.iloc[0]['Name of Company']}")
print(f"Quaterly Sales: {sorted_df.iloc[0]['Quaterly Sales (in
Cr)'].astype(int)} Cr")
```

```
sorted_df = dataFrame.sort_values(by='Quaterly Sales (in Cr)',
ascending=True)
print(f"Company: {sorted_df.iloc[0]['Name of Company']}")
print(f"Quaterly Sales: {sorted_df.iloc[0]['Quaterly Sales (in
Cr)'].astype(int)} Cr")
```

```
sorted_df = dataFrame.sort_values(by='Market Cap (in Cr)',
ascending=False)
largest_mk_cap_comp = sorted_df
for i in range(10):
    print(f"Market Cap: {sorted_df.iloc[i]['Name of Company']},
{sorted_df.iloc[i]['Market Cap (in Cr)'].astype(int)} Cr")
```

```
sorted_df = dataFrame.sort_values(by='Market Cap (in Cr)',
ascending=True)
smallest_mk_cap_comp = sorted_df
for i in range(10):
    print(f"Market Cap: {sorted_df.iloc[i]['Name of Company']},
{sorted_df.iloc[i]['Market Cap (in Cr)'].astype(int)} Cr")
```

```
sorted_df = dataFrame.sort_values(by='Quaterly Sales (in Cr)',
ascending=False)
largest_qt_sales_comp = sorted_df
for i in range(10):
    print(f"Quaterly Sales: {sorted_df.iloc[i]['Name of Company']},
{sorted_df.iloc[i]['Quaterly Sales (in Cr)'].astype(int)} Cr")
```

```
sorted_df = dataFrame.sort_values(by='Quaterly Sales (in Cr)',
ascending=True)
smallest_qt_sales_comp = sorted_df
for i in range(10):
    print(f"Quaterly Sales: {sorted_df.iloc[i]['Name of Company']},
{sorted_df.iloc[i]['Quaterly Sales (in Cr)'].astype(int)} Cr")
```

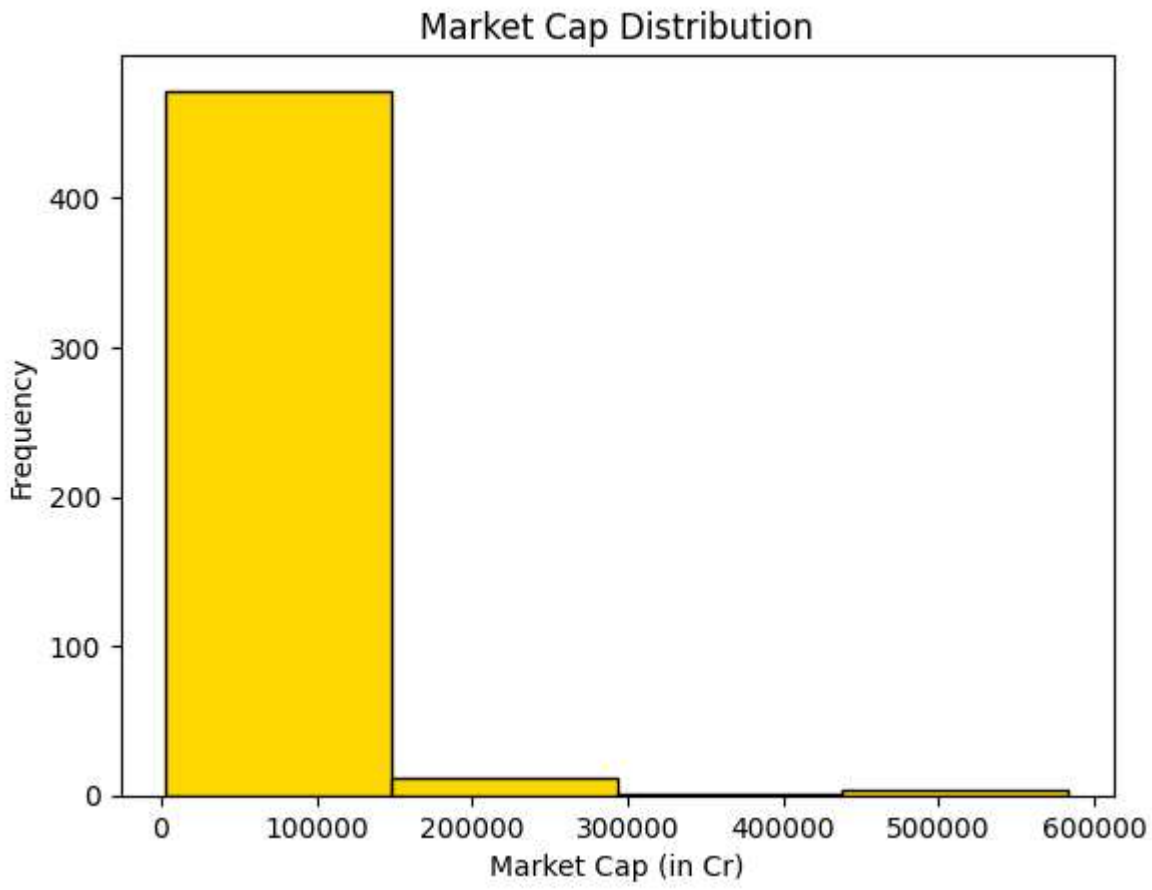
```
# this KPI measures the company's ability to generate market
capitalization per inr of sales
# higher values indicate better sales efficiency

dataFrame['Sales Efficiency Ratio'] = (dataFrame['Market Cap (in Cr)']
/ dataFrame['Quaterly Sales (in Cr)']).round(2)
```

```
sorted_df = dataFrame.sort_values(by='Sales Efficiency Ratio',
ascending=False)
largest_sales_eff_ratio_comp = sorted_df
for i in range(10):
    if sorted_df.iloc[i]['Sales Efficiency Ratio'] == 'inf':
        pass
    print(f"Sales Efficiency Ratio: {sorted_df.iloc[i]['Name of
Company']}, {sorted_df.iloc[i]['Sales Efficiency
Ratio'].astype(float)}")
```

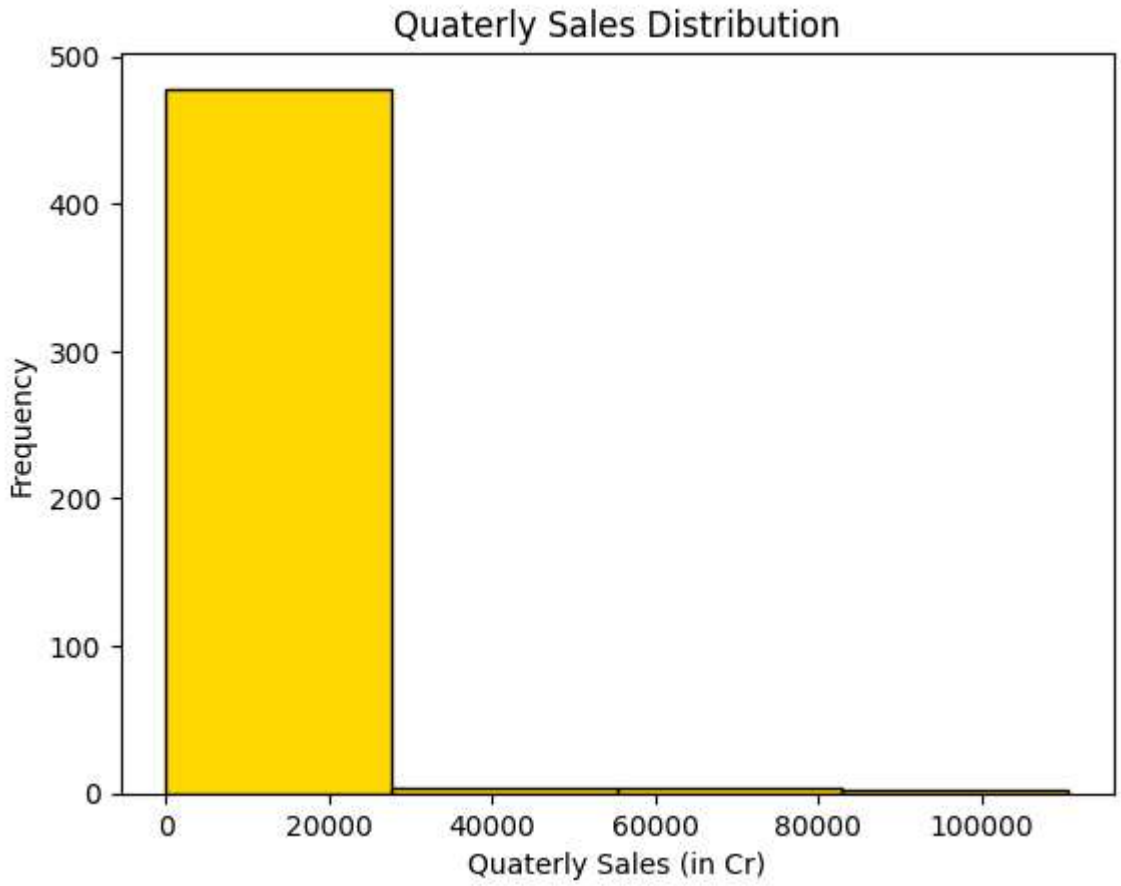
```
sorted_df = dataFrame.sort_values(by='Sales Efficiency Ratio',
ascending=True)
smallest_sales_eff_ratio_comp = sorted_df
for i in range(10):
    print(f"Sales Efficiency Ratio: {sorted_df.iloc[i]['Name of
Company']}, {sorted_df.iloc[i]['Sales Efficiency
Ratio'].astype(float)}")
```

Data Visualization

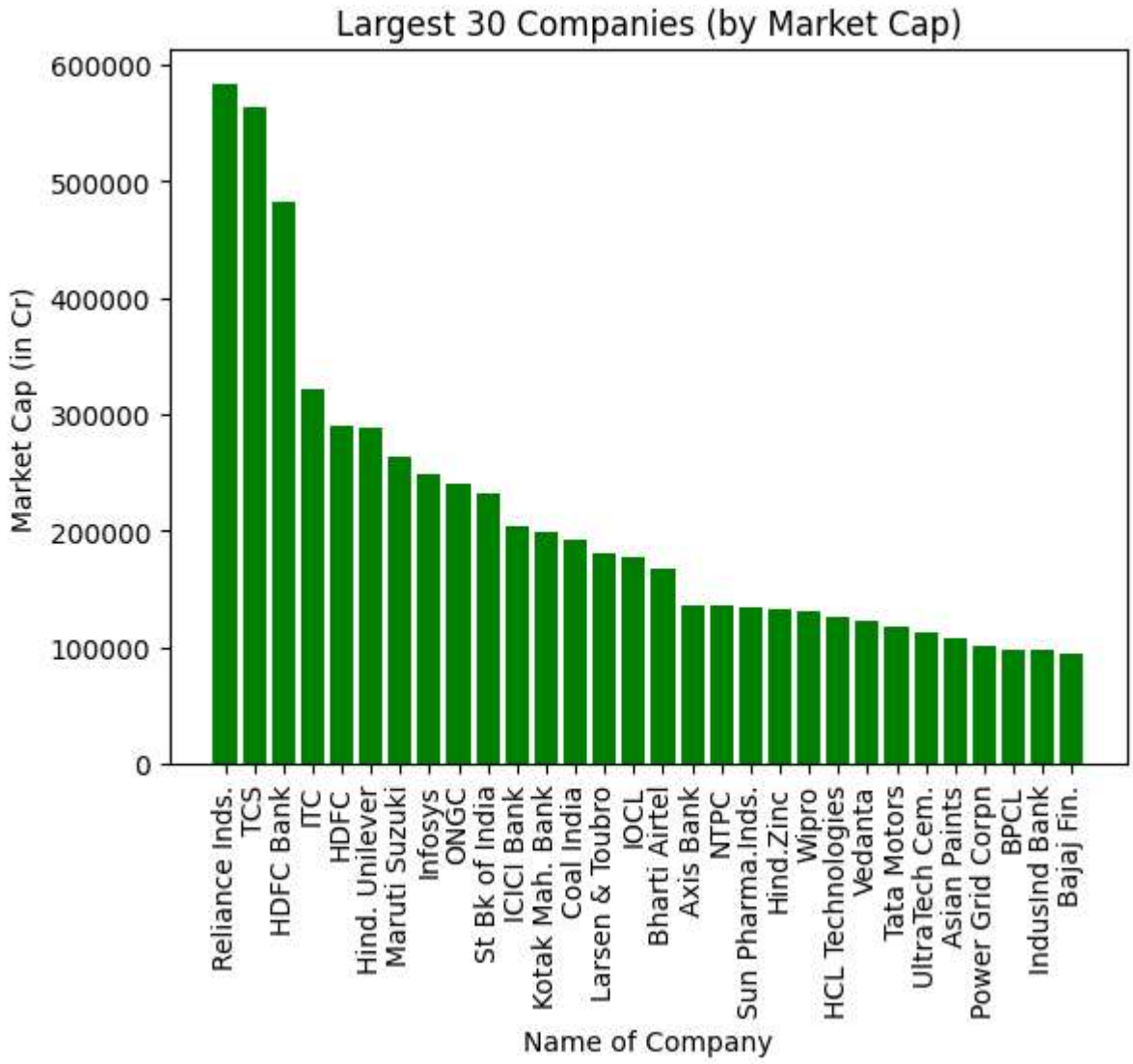
```
# histogram plot for Market Cap Distribution

plt.hist(dataFrame['Market Cap (in Cr)'], bins = 4, color = 'gold',
edgecolor = 'black')
plt.xlabel('Market Cap (in Cr)')
plt.ylabel('Frequency')
plt.title('Market Cap Distribution')
plt.show()
```



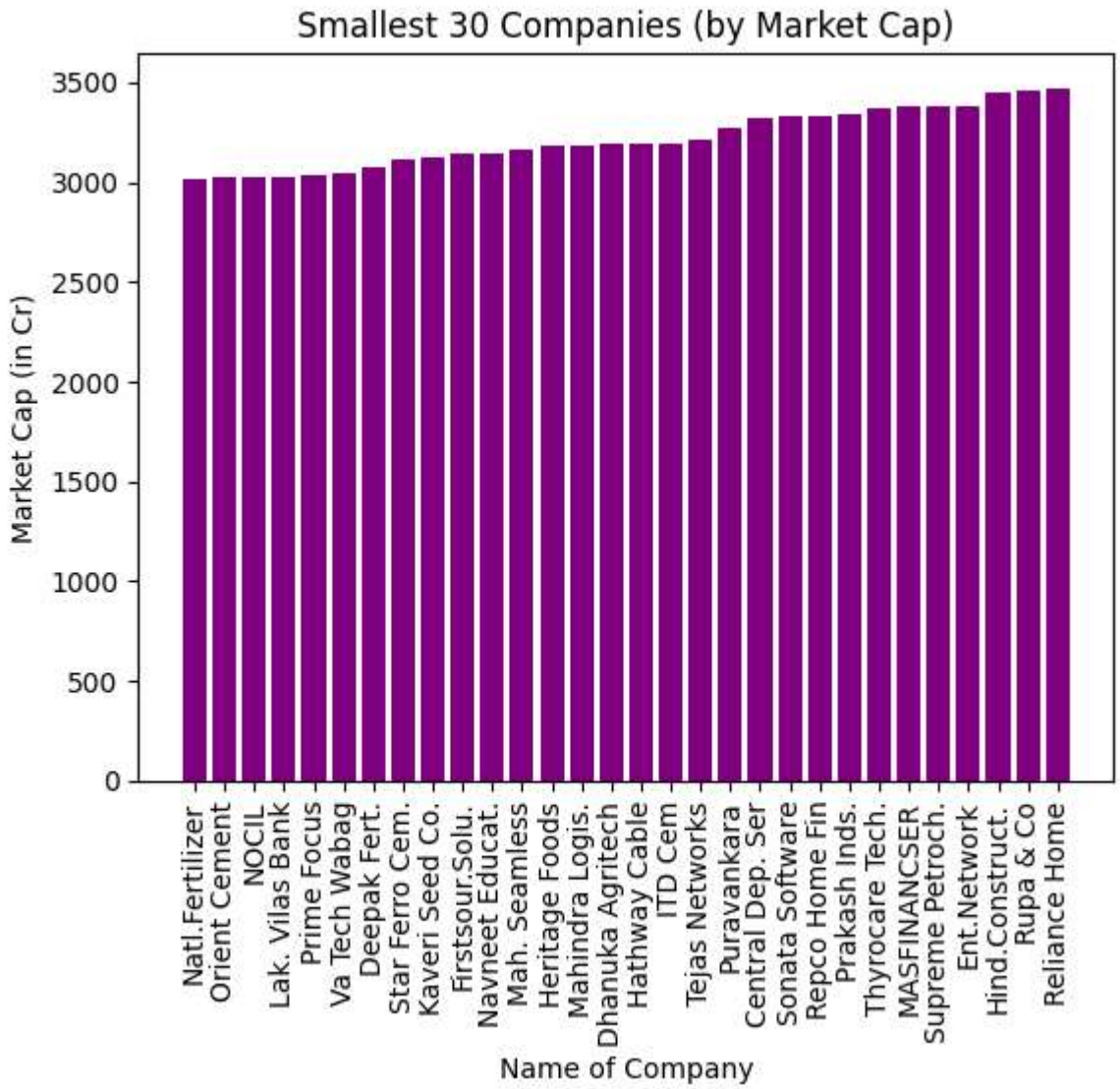
```
# histogram plot for Market Cap Distribution

plt.hist(dataFrame['Quaterly Sales (in Cr)'], bins = 4, color = 'gold',
edgecolor = 'black')
plt.xlabel('Quaterly Sales (in Cr)')
plt.ylabel('Frequency')
plt.title('Quaterly Sales Distribution')
plt.show()
```



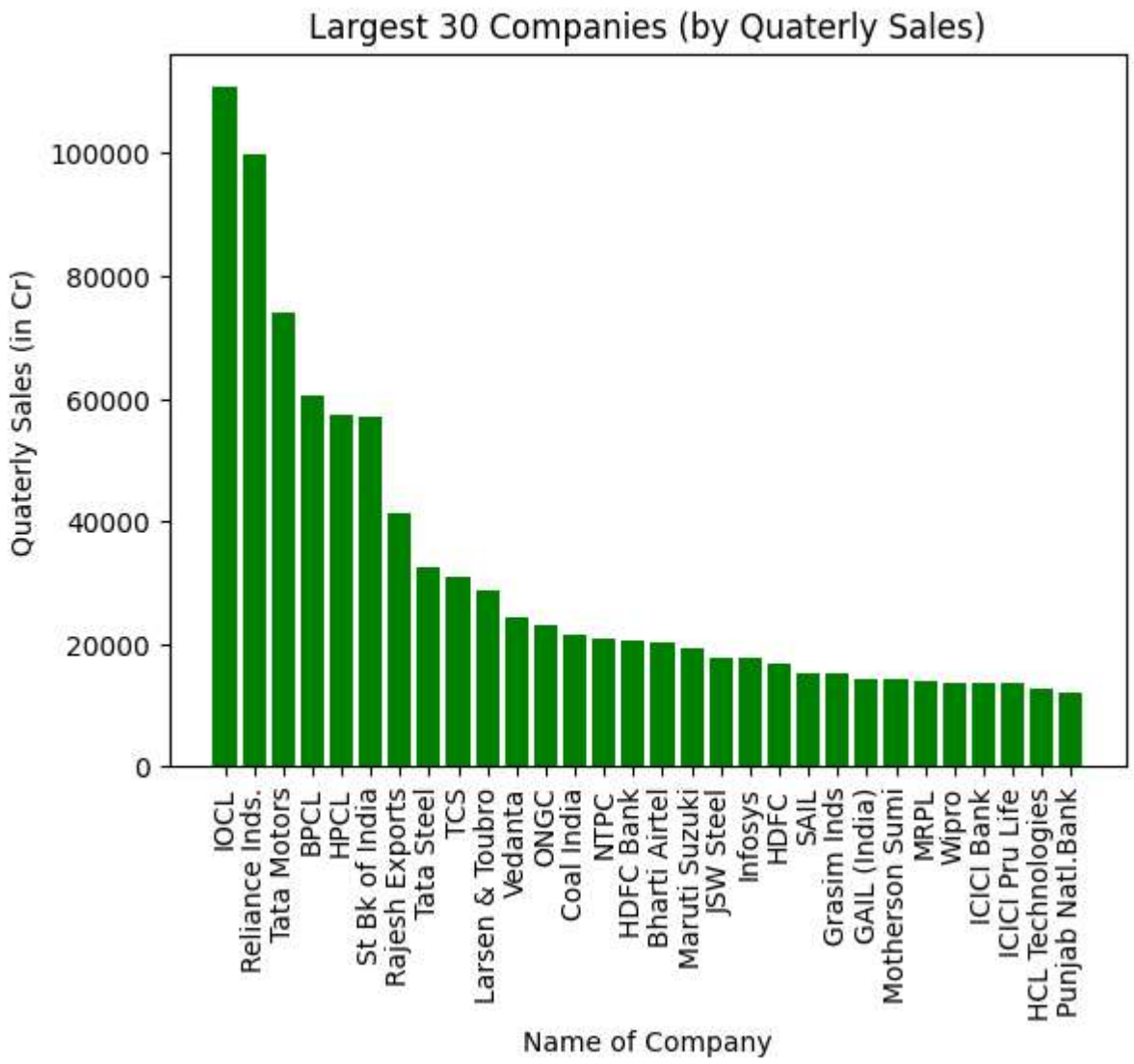
```
x = largest_mk_cap_comp['Name of Company'].head(30)
y = largest_mk_cap_comp['Market Cap (in Cr)'].head(30)
plt.bar(x, y, color = 'green')
plt.xticks(x, rotation = 'vertical')
plt.xlabel('Name of Company')
plt.ylabel('Market Cap (in Cr)')
plt.title('Largest 30 Companies (by Market Cap)')
plt.show()
```



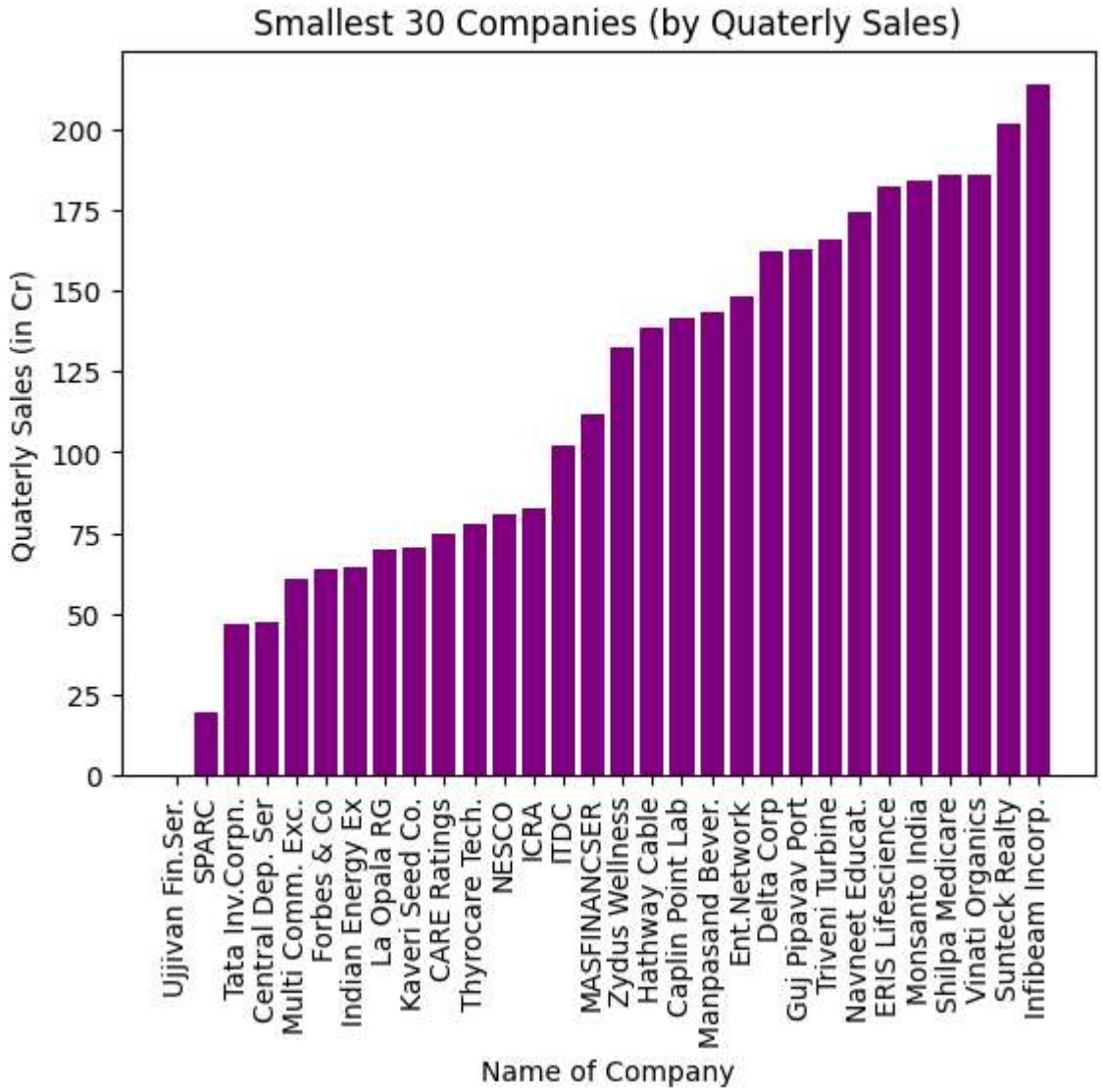
```
x = smallest_mk_cap_comp['Name of Company'].head(30)
y = smallest_mk_cap_comp['Market Cap (in Cr)'].head(30)
plt.bar(x, y, color = 'purple')
plt.xticks(x, rotation = 'vertical')
plt.xlabel('Name of Company')
plt.ylabel('Market Cap (in Cr)')
plt.title('Smallest 30 Companies (by Market Cap)')
plt.show()
```



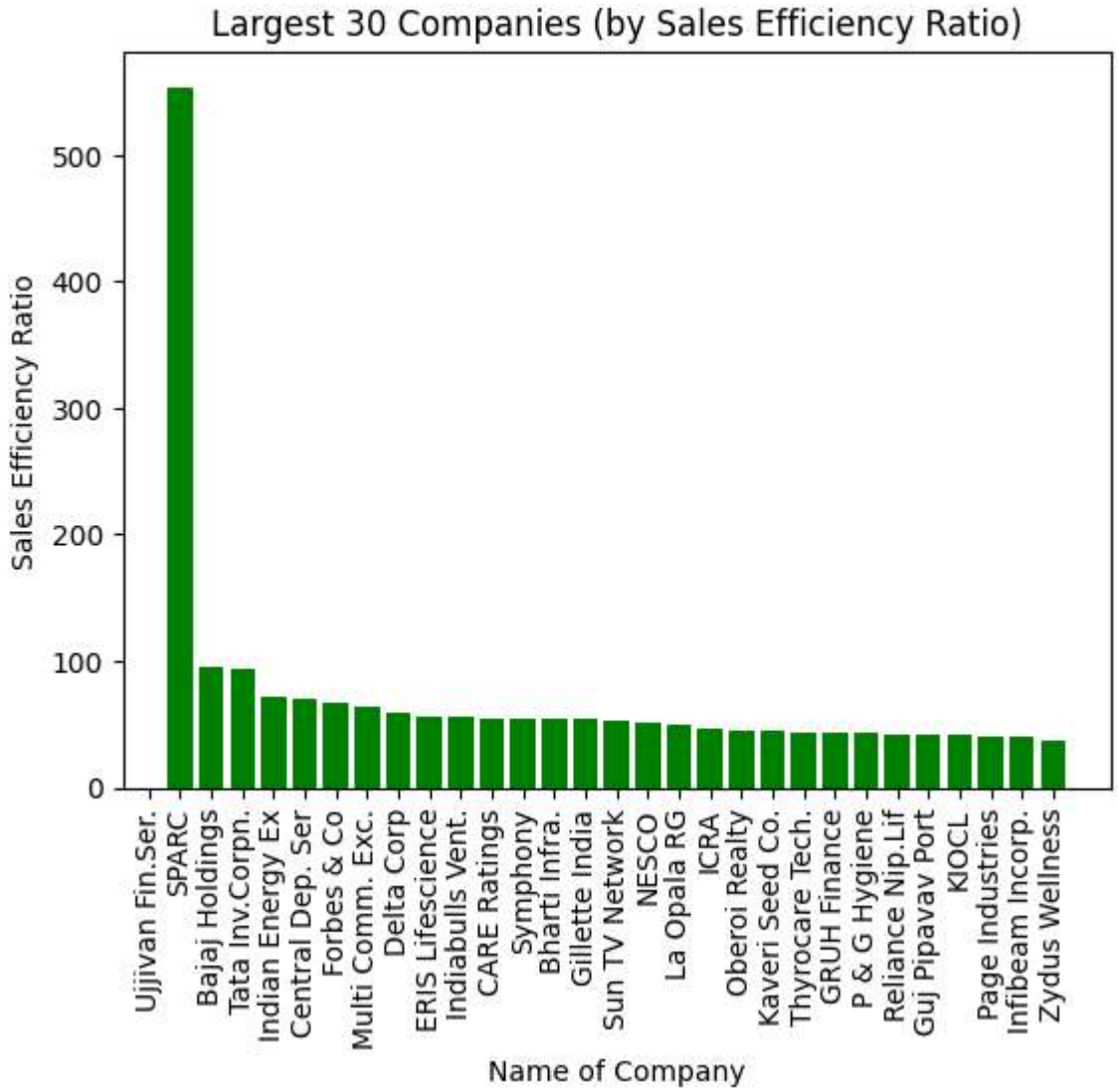
```
x = largest_qt_sales_comp['Name of Company'].head(30)
y = largest_qt_sales_comp['Quaterly Sales (in Cr)'].head(30)
plt.bar(x, y, color = 'green')
plt.xticks(x, rotation = 'vertical')
plt.xlabel('Name of Company')
plt.ylabel('Quaterly Sales (in Cr)')
plt.title('Largest 30 Companies (by Quaterly Sales)')
plt.show()
```



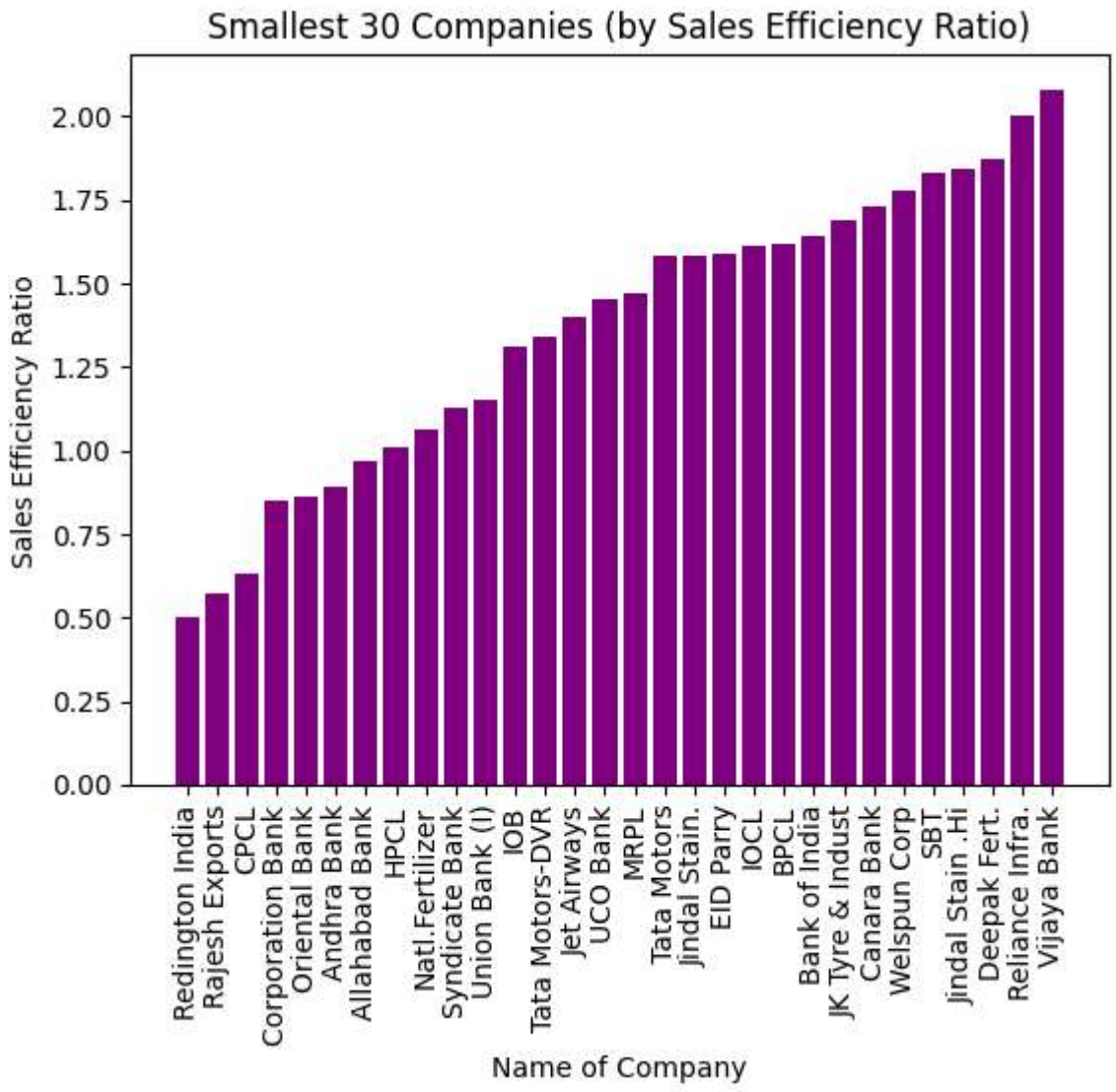
```
x = smallest_qt_sales_comp['Name of Company'].head(30)
y = smallest_qt_sales_comp['Quaterly Sales (in Cr)'].head(30)
plt.bar(x, y, color = 'purple')
plt.xticks(x, rotation = 'vertical')
plt.xlabel('Name of Company')
plt.ylabel('Quaterly Sales (in Cr)')
plt.title('Smallest 30 Companies (by Quaterly Sales)')
plt.show()
```



```
x = largest_sales_eff_ratio_comp['Name of Company'].head(30)
y = largest_sales_eff_ratio_comp['Sales Efficiency Ratio'].head(30)
plt.bar(x, y, color = 'green')
plt.xticks(x, rotation = 'vertical')
plt.xlabel('Name of Company')
plt.ylabel('Sales Efficiency Ratio')
plt.title('Largest 30 Companies (by Sales Efficiency Ratio)')
plt.show()
```



```
x = smallest_sales_eff_ratio_comp['Name of Company'].head(30)
y = smallest_sales_eff_ratio_comp['Sales Efficiency Ratio'].head(30)
plt.bar(x, y, color = 'purple')
plt.xticks(x, rotation = 'vertical')
plt.xlabel('Name of Company')
plt.ylabel('Sales Efficiency Ratio')
plt.title('Smallest 30 Companies (by Sales Efficiency Ratio)')
plt.show()
```



Insights

```
1. Reliance Industries (5,83,436 Cr) has the biggest market cap share
2. Natl. Fertilizers (3,017 Cr) has the smallest market share
3. IOCL (1,10,666 Cr) has most quaterly sales
4. Ujjivan Financial Services  (0 Cr) has least quaterly sales
5. SPARC (553.82) has the highest Sales Efficiency Ratio
6. Ujjivan Financial Services (0.0) has the lowest sales Sales
Efficiency Ratio
7. Majority companies market cap lies between between 0 - 1,50,000 Cr
8. Majority companies quaterly sales lies between 0 - 27,500 Cr
```

Conclusion

```
- Reliance Industries stands out with the largest market
capitalization, while Ujjivan Financial Services appears to be
struggling with both low quarterly sales and low sales efficiency
- Most companies fall within certain ranges of market capitalization
and quarterly sales, indicating the typical distribution in the market
- Overall, these insights provide a snapshot of the market dynamics and
the performance of different companies within it
```