In [ ]:

```
import pandas as pd
data= pd.read_csv('/content/Financial Analytics data.csv')
#print(data.describe())
print(data.head(10))
#print(data.info())
```

```
   S.No.            Name  Mar Cap - Crore  Sales Qtr - Crore
0      1  Reliance Inds.        583436.72           99810.00
1      2             TCS        563709.84           30904.00
2      3       HDFC Bank        482953.59           20581.27
3      4             ITC        320985.27            9772.02
4      5         H D F C        289497.37           16840.51
5      6  Hind. Unilever        288265.26            8590.00
6      7   Maruti Suzuki        263493.81           19283.20
7      8         Infosys        248320.35           17794.00
8      9         O N G C        239981.50           22995.88
9     10  St Bk of India        232763.33           57014.08
```

In [ ]:

```
#check the proportion of the missing value
missing_value=data.isnull().sum()
missing_value
```

Out[ ]:

	0
S.No.	0
Name	0
Mar Cap - Crore	9
Sales Qtr - Crore	29

dtype: int64

In [ ]:

```
# Impute missing values using the median
data['Mar Cap - Crore'].fillna(data['Mar Cap - Crore'].median(), inplace=
data['Sales Qtr - Crore'].fillna(data['Sales Qtr - Crore'].median(), inpl
```

In [ ]:

```
# Verify that there are no more missing values
data.isnull().sum()
```

Out[ ]:

	0
S.No.	0
Name	0
Mar Cap - Crore	0
Sales Qtr - Crore	0

dtype: int64

In [ ]:

```
#Removing S.NO
data=data.drop('S.No.',axis=1)
```

In [ ]:

```
# Calculate summary statistics for numerical columns
summary_stats = data.describe()
```

```
# Display the summary statistics
summary_stats
```

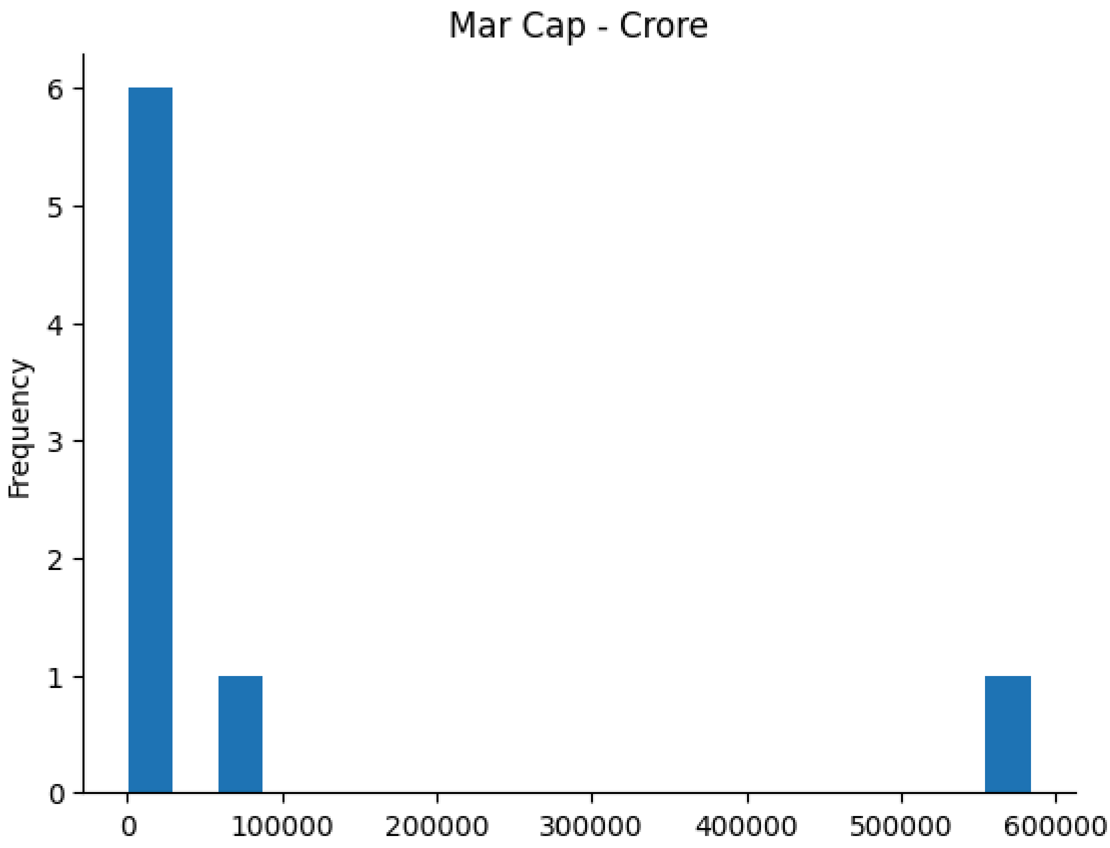
Out[ ]:

	Mar Cap - Crore	Sales Qtr - Crore
count	488.000000	488.000000
mean	27708.961086	3649.084570
std	58963.329098	9708.054143
min	3017.070000	0.000000
25%	4879.612500	570.035000
50%	9885.050000	1137.170000
75%	23400.815000	2580.797500
max	583436.720000	110666.930000

In [ ]:

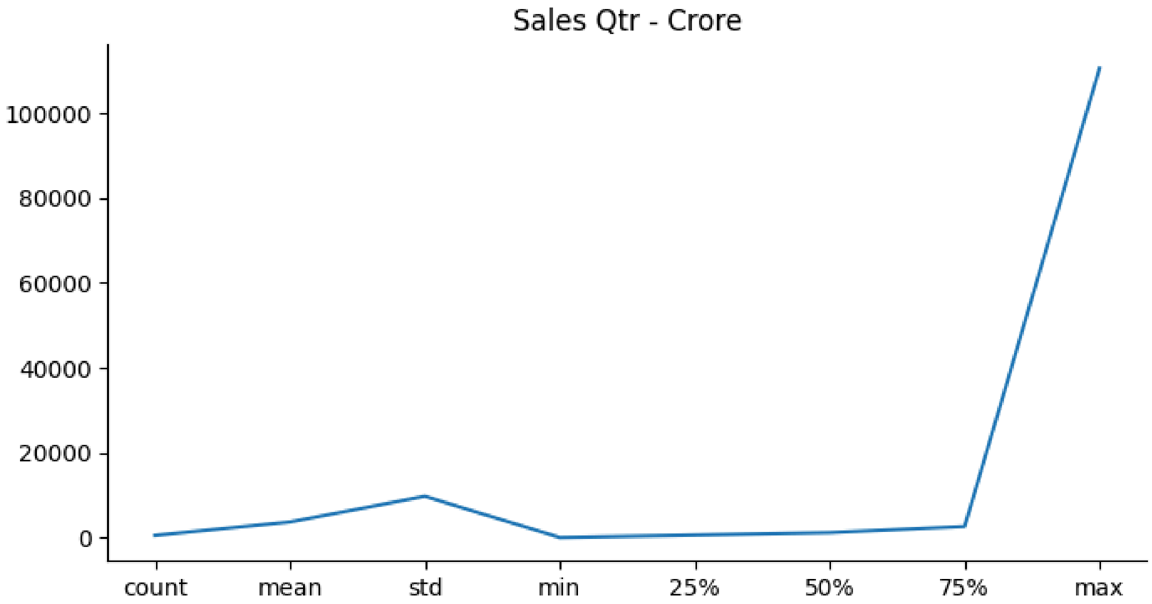
In [ ]:

```
from matplotlib import pyplot as plt
summary_stats['Mar Cap - Crore'].plot(kind='hist', bins=20, title='Mar Ca
plt.gca().spines[['top', 'right',]].set_visible(False)
```

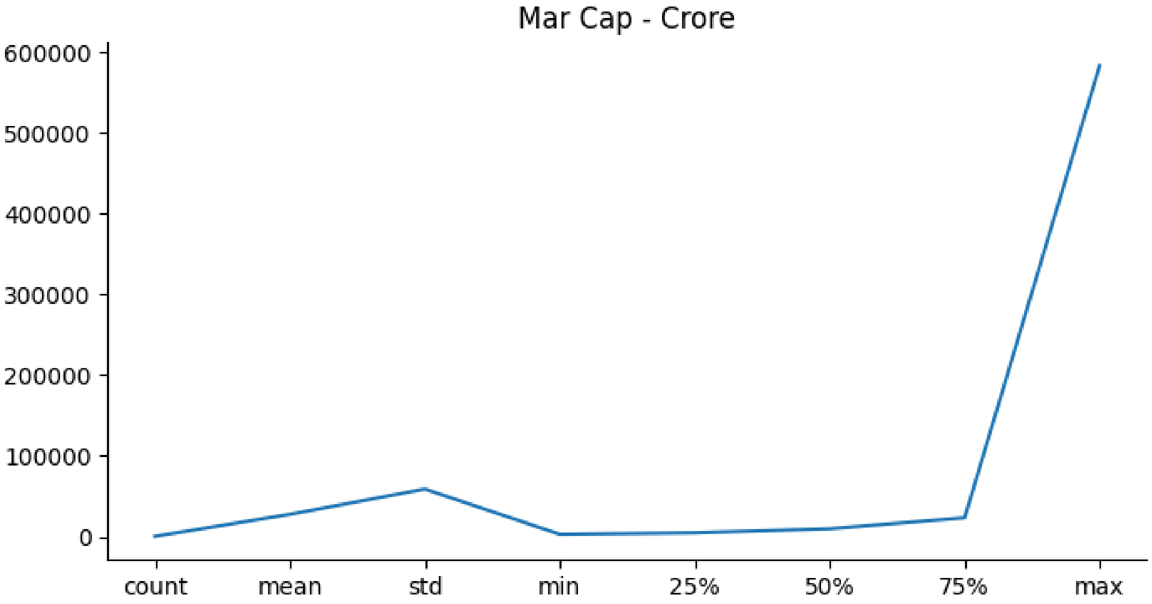


In [ ]:

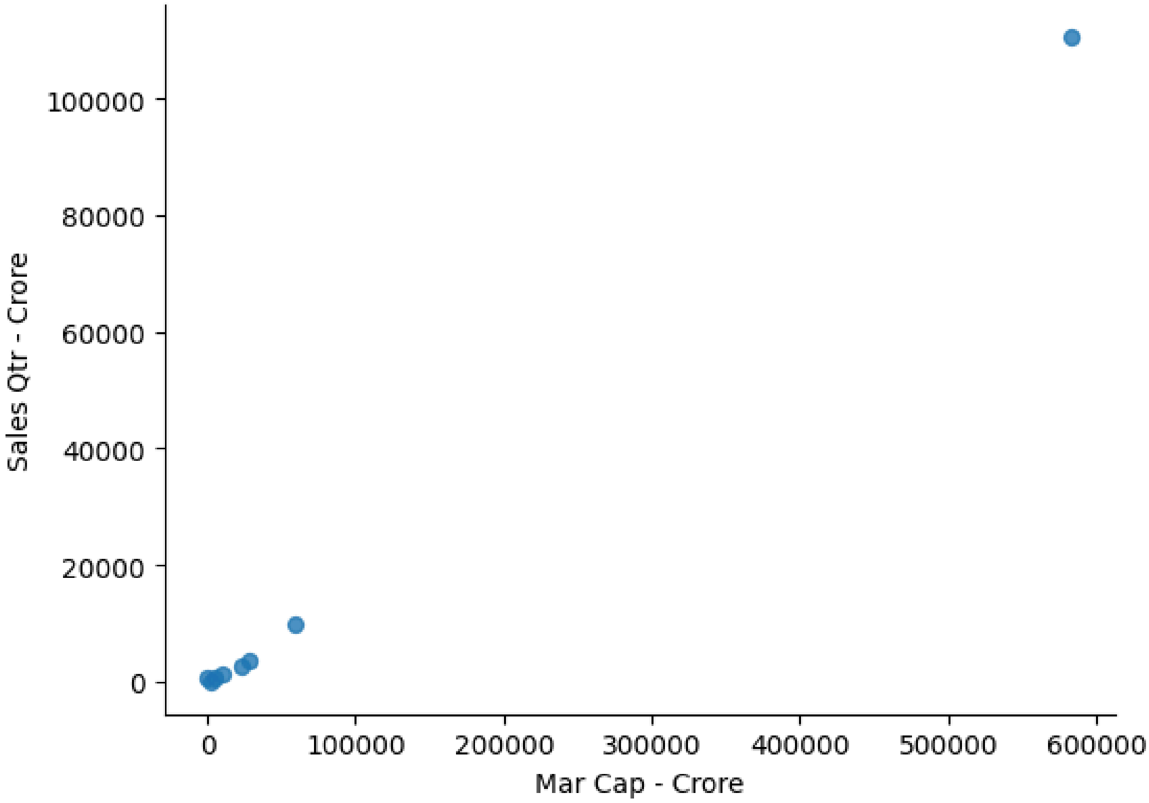
```
from matplotlib import pyplot as plt
summary_stats['Sales Qtr - Crore'].plot(kind='line', figsize=(8, 4), titl
plt.gca().spines[['top', 'right']].set_visible(False)
```



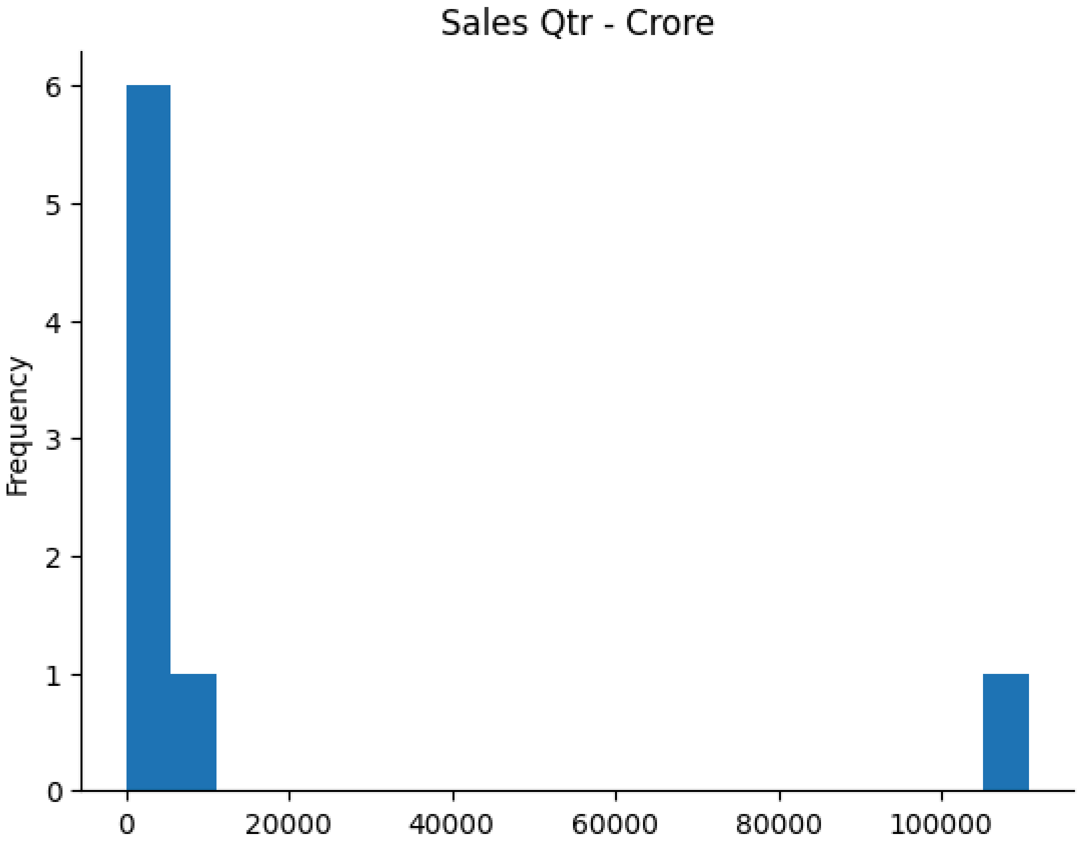
```
In [ ]: from matplotlib import pyplot as plt
        summary_stats['Mar Cap - Crore'].plot(kind='line', figsize=(8, 4), title=
        plt.gca().spines[['top', 'right']].set_visible(False)
```



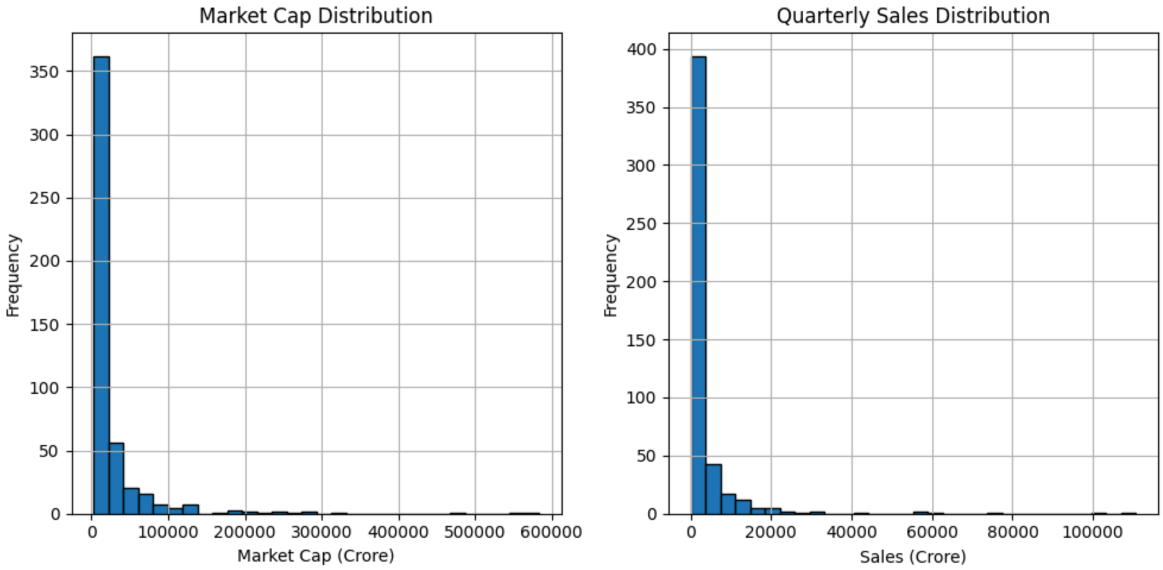
```
In [ ]: from matplotlib import pyplot as plt
        summary_stats.plot(kind='scatter', x='Mar Cap - Crore', y='Sales Qtr - Cr
        plt.gca().spines[['top', 'right',]].set_visible(False)
```



In [ ]:
```
from matplotlib import pyplot as plt
summary_stats['Sales Qtr - Crore'].plot(kind='hist', bins=20, title='Sale
plt.gca().spines[['top', 'right',]].set_visible(False)
```

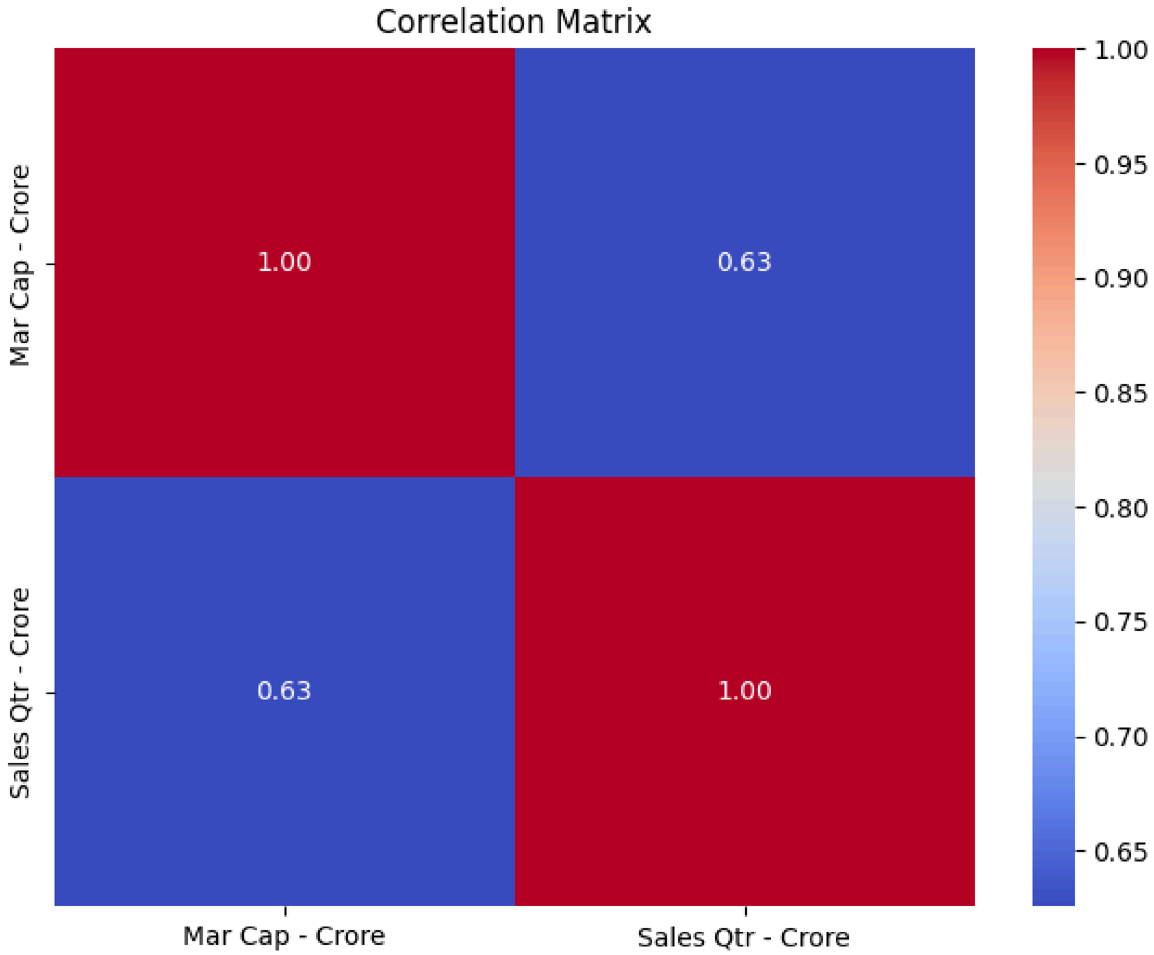


In [ ]:
```
import matplotlib.pyplot as plt

# Histograms
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
data['Mar Cap - Crore'].hist(bins=30, edgecolor='k')
plt.title('Market Cap Distribution')
plt.xlabel('Market Cap (Crore)')
plt.ylabel('Frequency')

plt.subplot(1, 2, 2)
data['Sales Qtr - Crore'].hist(bins=30, edgecolor='k')
plt.title('Quarterly Sales Distribution')
plt.xlabel('Sales (Crore)')
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```



In [ ]:

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


# Select only numeric columns for correlation calculation
numeric_data = data.select_dtypes(include=[float, int])

# Compute the correlation matrix
corr_matrix = numeric_data.corr()

# Plot the correlation matrix
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix')
plt.show()
```



In [ ]:

```
correlation_coefficient = data['Mar Cap - Crore'].corr(data['Sales Qtr -
correlation_coefficient
```

Out[ ]:

```
0.6256901302344715
```

In [ ]:

```
total_market_cap = data['Mar Cap - Crore'].sum()
total_sales = data['Sales Qtr - Crore'].sum()
avg_market_cap = data['Mar Cap - Crore'].mean()
avg_sales = data['Sales Qtr - Crore'].mean()

print(f"Total Market Cap: {total_market_cap}")
print(f"Total Quarterly Sales: {total_sales}")
print(f"Average Market Cap: {avg_market_cap}")
print(f"Average Quarterly Sales: {avg_sales}")
```

```
Total Market Cap: 13521973.01
Total Quarterly Sales: 1780753.2699999998
Average Market Cap: 27708.961086065574
Average Quarterly Sales: 3649.084569672131
```

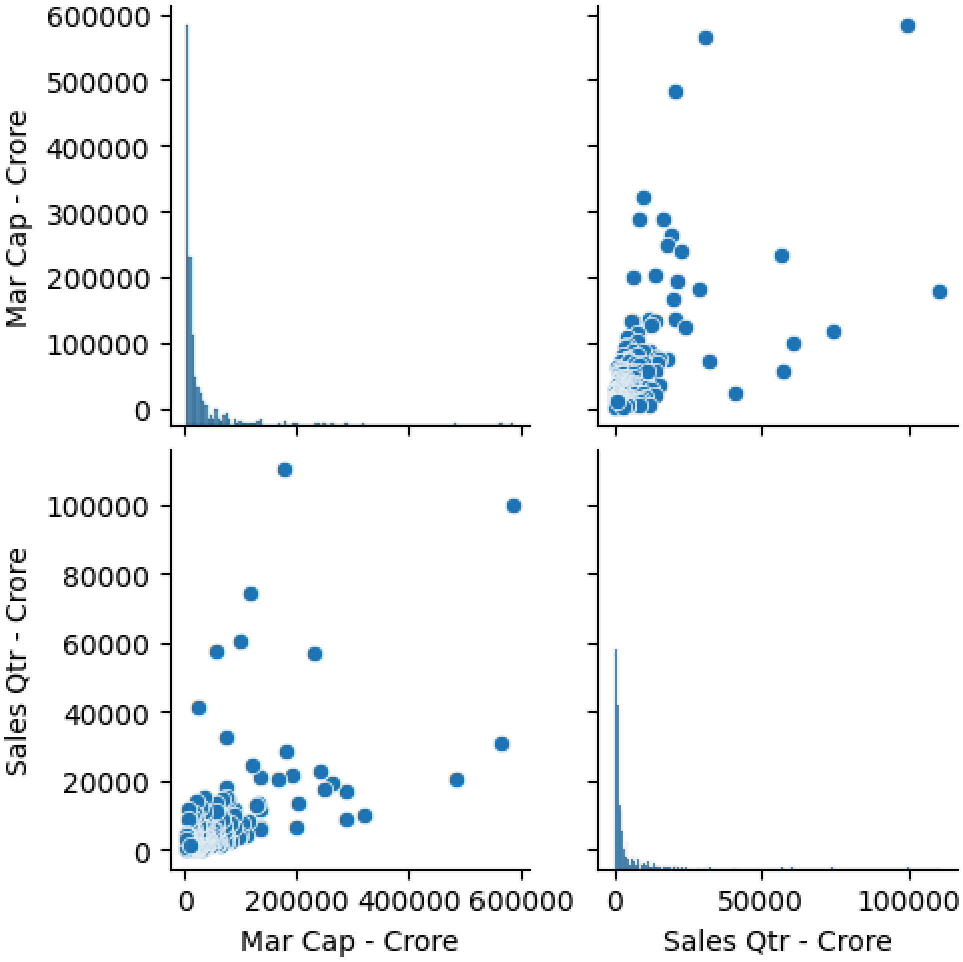
In [ ]:

```
category_analysis = data.groupby('Name').agg({
    'Mar Cap - Crore': ['sum', 'mean', 'median'],
    'Sales Qtr - Crore': ['sum', 'mean', 'median']
})
print(category_analysis)
```

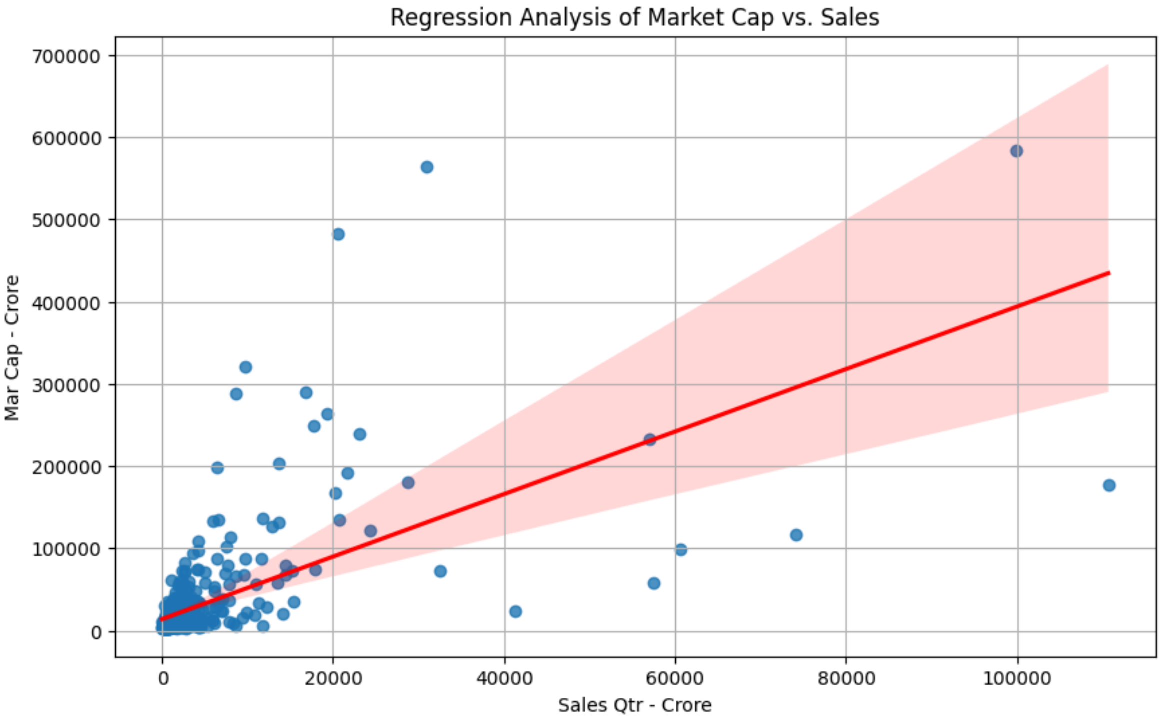
```
                 Mar Cap - Crore                     Sales Qtr - Crore  \
                             sum      mean    median               sum   
Name                                                                     
3M India                23101.19  23101.19  23101.19            645.77   
A B B                   31983.33  31983.33  31983.33           2779.40   
ACC                     30803.68  30803.68  30803.68           3494.24   
AIA Engg.               13593.35  13593.35  13593.35            572.16   
APL Apollo               4775.03   4775.03   4775.03           1314.38   
...                          ...       ...       ...               ...   
Yes Bank                71028.13  71028.13  71028.13           5070.30   
Zee Entertainmen        54817.89  54817.89  54817.89           1838.07   
Zensar Tech.             4066.42   4066.42   4066.42            793.76   
Zydus Wellness           4921.45   4921.45   4921.45            132.40   
eClerx Services          5259.14   5259.14   5259.14            339.89   


                     mean   median  
Name                                
3M India           645.77   645.77  
A B B             2779.40  2779.40  
ACC               3494.24  3494.24  
AIA Engg.          572.16   572.16  
APL Apollo        1314.38  1314.38  
...                   ...      ...  
Yes Bank          5070.30  5070.30  
Zee Entertainmen  1838.07  1838.07  
Zensar Tech.       793.76   793.76  
Zydus Wellness     132.40   132.40  
eClerx Services    339.89   339.89  

[488 rows x 6 columns]
```

In [ ]:

```
# Pairplot to see relationships between variables
sns.pairplot(data)
plt.show()
```



```
In [ ]: import statsmodels.api as sm
        import matplotlib.pyplot as plt
        import seaborn as sns

        # Independent variable (Sales)
        X = data['Sales Qtr - Crore']
        # Dependent variable (Market Cap)
        y = data['Mar Cap - Crore']

        # Add a constant to the independent variable
        X = sm.add_constant(X)

        # Fit the regression model
        model = sm.OLS(y, X).fit()

        # Get the regression results
        summary = model.summary()

        # Scatter plot with regression line
        plt.figure(figsize=(10, 6))
        sns.regplot(x='Sales Qtr - Crore', y='Mar Cap - Crore',data=data, line_kw
        plt.title('Regression Analysis of Market Cap vs. Sales')
        plt.xlabel('Sales Qtr - Crore')
        plt.ylabel('Mar Cap - Crore')
        plt.grid(True)
        plt.show()

        summary
```


Regression Analysis of Market Cap vs. Sales
Mar Cap - Crore
700000
600000
500000
400000
300000
200000
100000
0
0
20000
40000
60000
80000
100000
Sales Qtr - Crore

Out[ ]:

OLS Regression Results

Dep. Variable:	Mar Cap - Crore	**R-squared:**	0.391
Model:	OLS	**Adj. R-squared:**	0.390
Method:	Least Squares	**F-statistic:**	312.7
Date:	Sat, 17 Aug 2024	**Prob (F-statistic):**	2.18e-54
Time:	10:57:57	**Log-Likelihood:**	-5931.3
No. Observations:	488	**AIC:**	1.187e+04
Df Residuals:	486	**BIC:**	1.187e+04
Df Model:	1		
Covariance Type:	nonrobust		

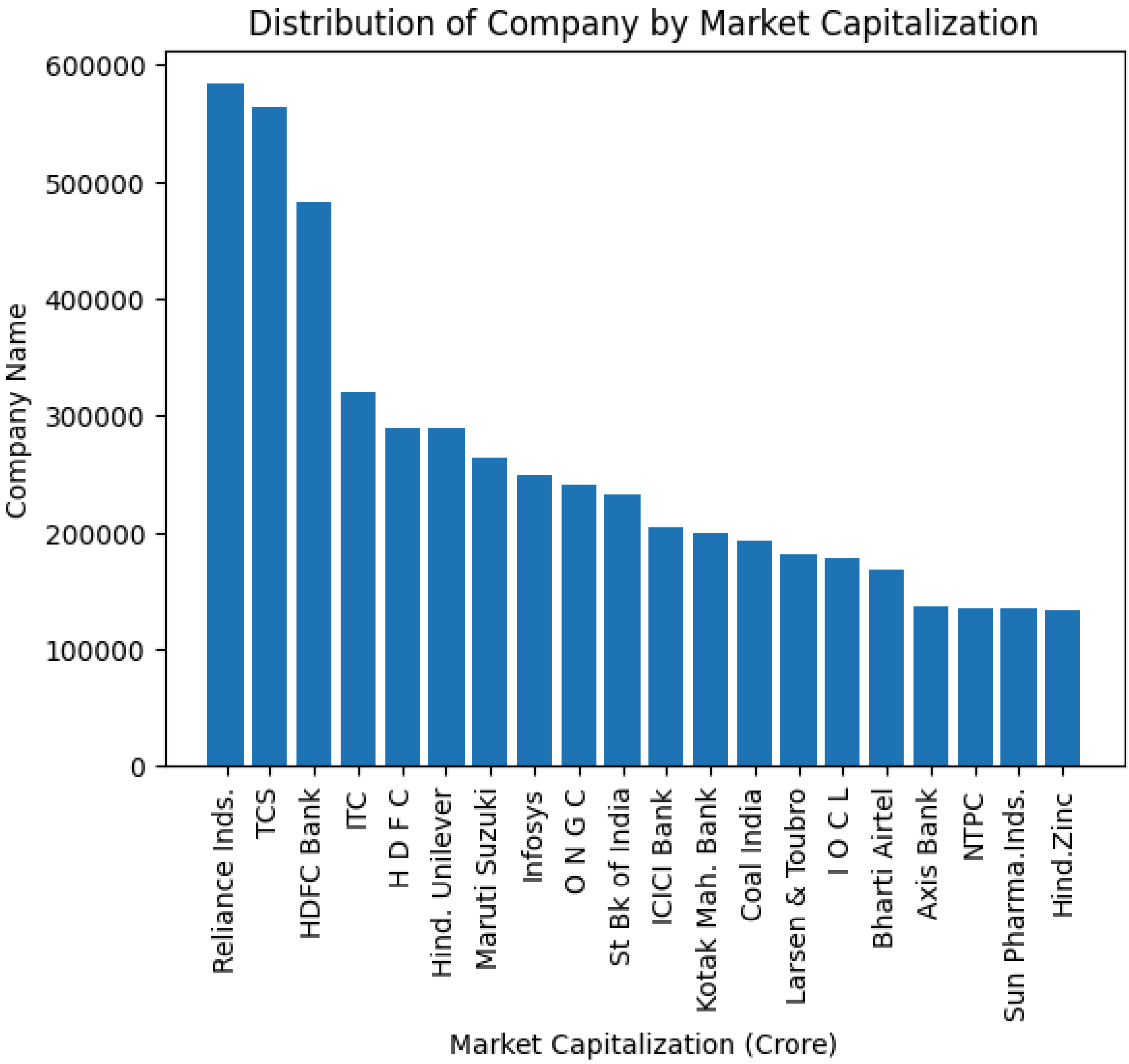
	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	1.384e+04	2226.925	6.216	0.000	9466.035	1.82e+04
Sales Qtr - Crore	3.8002	0.215	17.682	0.000	3.378	4.223

Omnibus:	458.190	**Durbin-Watson:**	0.693
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	25240.135
Skew:	3.852	**Prob(JB):**	0.00
Kurtosis:	37.380	**Cond. No.**	1.11e+04

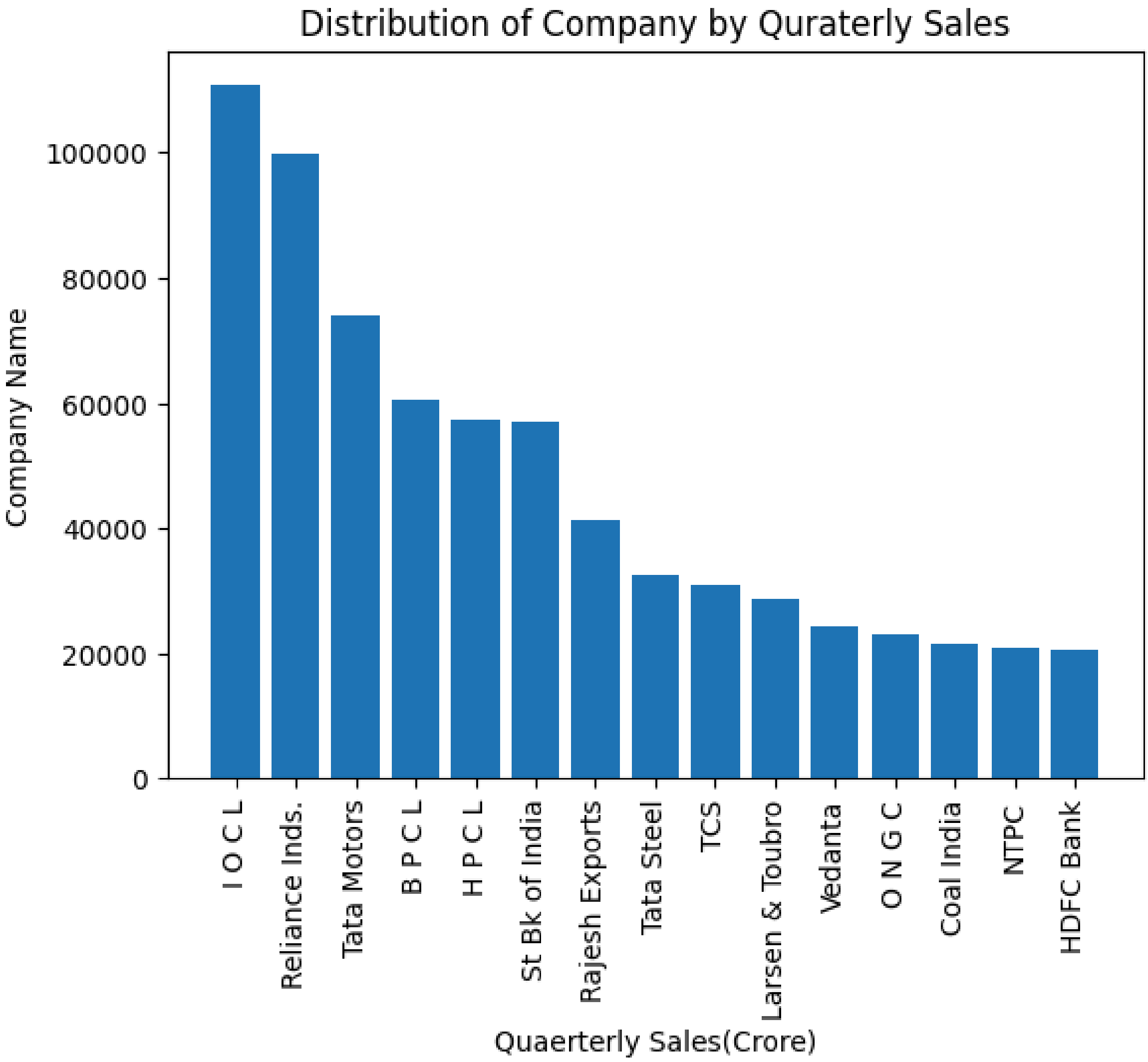
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.11e+04. This might indicate that there are strong multicollinearity or other numerical problems.

In [ ]:

```python
data=data.sort_values('Mar Cap - Crore', ascending=False)
plt.bar(data['Name'].head(20), data['Mar Cap - Crore'].head(20),)
plt.ylabel("Company Name")  # Corrected typo
plt.xlabel("Market Capitalization (Crore)")  # Fixed label based on data
plt.title("Distribution of Company by Market Capitalization")
plt.xticks(rotation='vertical')# Adjusted title
plt.show()
```



In [ ]:
```
data=data.sort_values('Sales Qtr - Crore', ascending=False)
plt.bar(data['Name'].head(15), data['Sales Qtr - Crore'].head(15))
plt.ylabel("Company Name")
plt.xlabel("Quaerterly Sales(Crore)")
plt.title("Distribution of Company by Quraterly Sales")
plt.xticks(rotation='vertical')
plt.show()
```

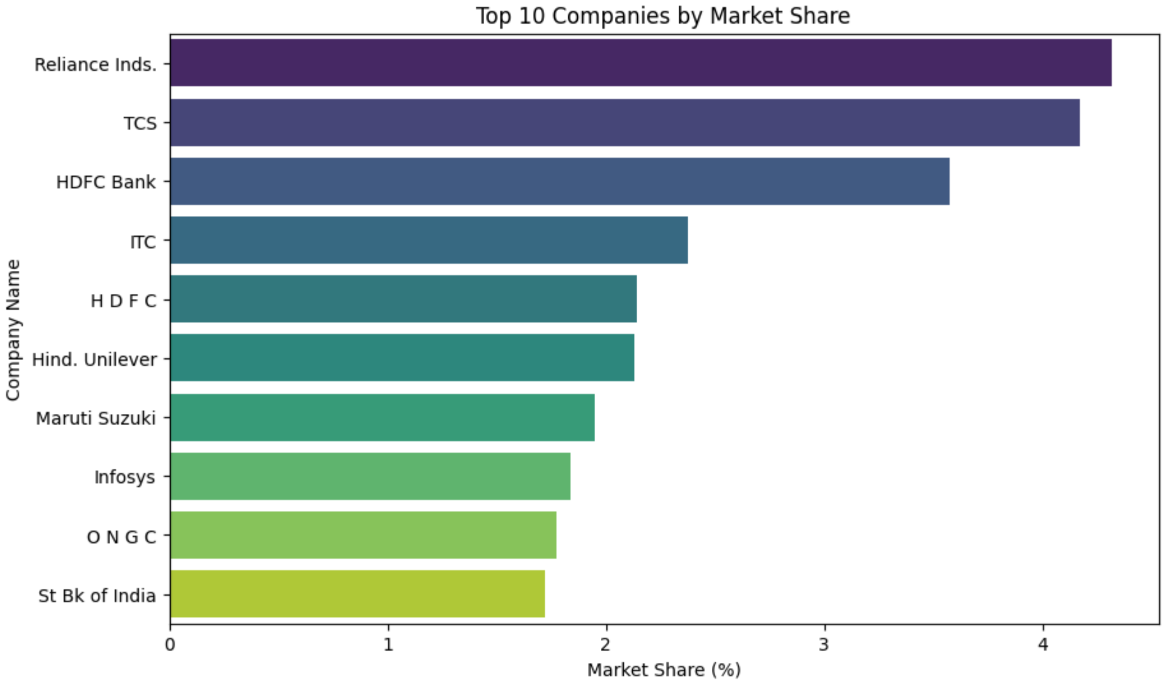


In [ ]:

```
import pandas as pd


# Financial Analysis
data=data.sort_values('Mar Cap - Crore', ascending=False)
# Market Share Analysis (assuming 'Mar Cap - Crore' represents total mark
total_market_cap = data['Mar Cap - Crore'].sum()
data['Market Share (%)'] = data['Mar Cap - Crore'] / total_market_cap * 1
print("\nMarket Share:")
print(data[['Name', 'Market Share (%)']].round(2))  # Round to 2 decimal
# Recalculate Market Share (%)
data['Market Share (%)'] = data['Mar Cap - Crore'] / data['Mar Cap - Cror

# Visualize Top 10 Companies by Market Share
top_market_share = data.sort_values('Market Share (%)', ascending=False).
plt.figure(figsize=(10, 6))
sns.barplot(x='Market Share (%)', y='Name', data=top_market_share, palett
plt.title('Top 10 Companies by Market Share')
plt.xlabel('Market Share (%)')
plt.ylabel('Company Name')
plt.show()
```

```
Market Share:
                Name  Market Share (%)
0     Reliance Inds.              4.31
1                TCS              4.17
2          HDFC Bank              3.57
3                ITC              2.37
4            H D F C              2.14
..               ...               ...
482      Prime Focus              0.02
483  Lak. Vilas Bank              0.02
484            NOCIL              0.02
485    Orient Cement              0.02
486  Natl.Fertilizer              0.02

[488 rows x 2 columns]
```

```
<ipython-input-27-679793326d9c>:17: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be remove
d in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for
the same effect.

  sns.barplot(x='Market Share (%)', y='Name', data=top_market_share, palet
te='viridis')
```


Top 10 Companies by Market Share
Company Name
Reliance Inds.
TCS
HDFC Bank
ITC
H D F C
Hind. Unilever
Maruti Suzuki
Infosys
O N G C
St Bk of India
Market Share (%)