In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
# Load the dataset

df = pd.read_csv(r"C:\Users\ASD\Desktop\Financial Analytics\Financial Analytics dat
df.head()
```

Out[4]:

	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

In [6]:
```
new_df=df[['Name','Mar Cap - Crore','Sales Qtr - Crore']]
new_df
```

Out[6]:

	Name	Mar Cap - Crore	Sales Qtr - Crore
0	Reliance Inds.	583436.72	99810.00
1	TCS	563709.84	30904.00
2	HDFC Bank	482953.59	20581.27
3	ITC	320985.27	9772.02
4	H D F C	289497.37	16840.51
...	...	...	...
483	Lak. Vilas Bank	3029.57	790.17
484	NOCIL	3026.26	249.27
485	Orient Cement	3024.32	511.53
486	Natl.Fertilizer	3017.07	2840.75
487	L T Foods	NaN	NaN

488 rows × 3 columns

In [7]:
```
# Check Column and Rows Counts

new_df.shape
```

Out[7]:
```
(488, 3)
```

In [8]:
```
# Check for missing values

new_df.isna().sum()
```

Out[8]:
```
Name                  0
Mar Cap - Crore       9
Sales Qtr - Crore    123
dtype: int64
```

In [9]:
```
# drop Null values

ef= new_df.dropna()
ef
```

Out[9]:

	Name	Mar Cap - Crore	Sales Qtr - Crore
0	Reliance Inds.	583436.72	99810.00
1	TCS	563709.84	30904.00
2	HDFC Bank	482953.59	20581.27
3	ITC	320985.27	9772.02
4	H D F C	289497.37	16840.51
...	...	...	...
482	Prime Focus	3031.50	609.61
483	Lak. Vilas Bank	3029.57	790.17
484	NOCIL	3026.26	249.27
485	Orient Cement	3024.32	511.53
486	Natl.Fertilizer	3017.07	2840.75

365 rows × 3 columns

In [10]:
```
# Check Column and Rows Counts

ef.shape
```

Out[10]:
```
(365, 3)
```

In [11]:
```
# Check details about data

ef.info()
```

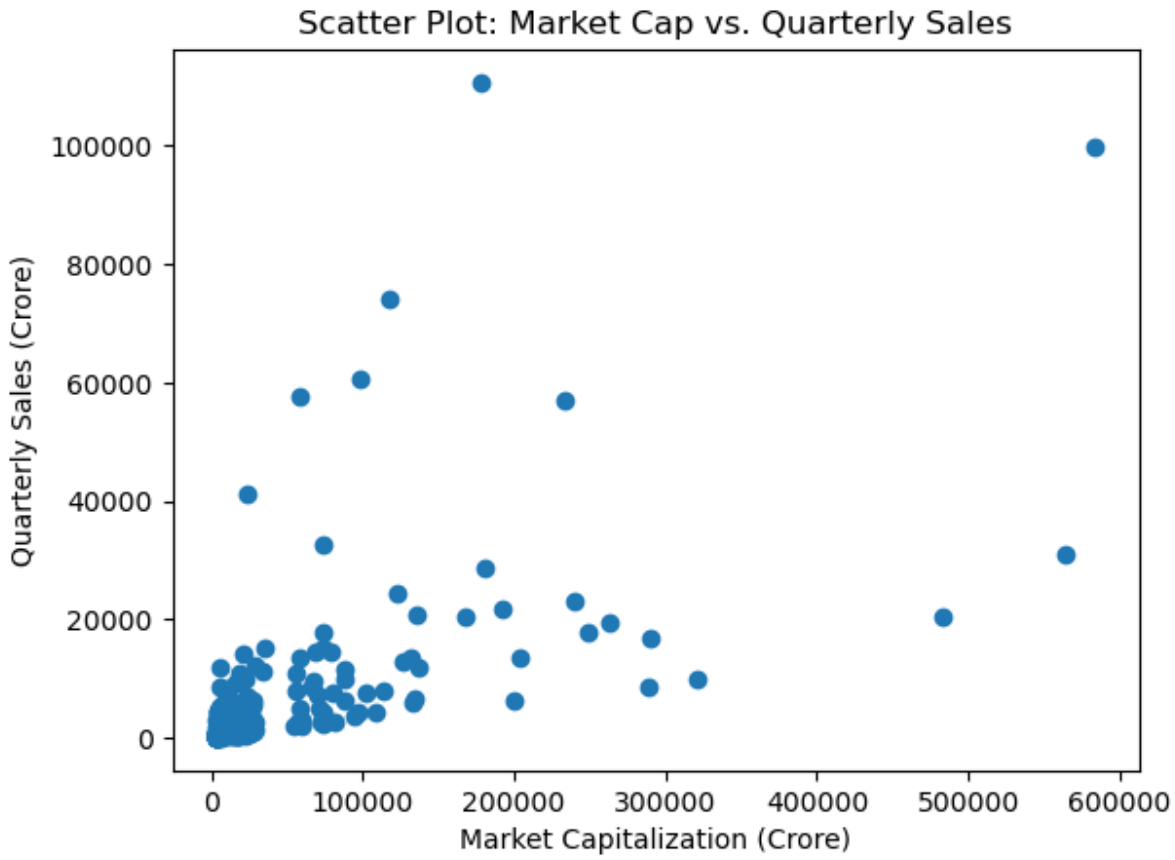
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 365 entries, 0 to 486
Data columns (total 3 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Name               365 non-null    object 
 1   Mar Cap - Crore    365 non-null    float64
 2   Sales Qtr - Crore  365 non-null    float64
dtypes: float64(2), object(1)
memory usage: 11.4+ KB
```

In [12]:
```
# Check Statistical Details

ef.describe()
```

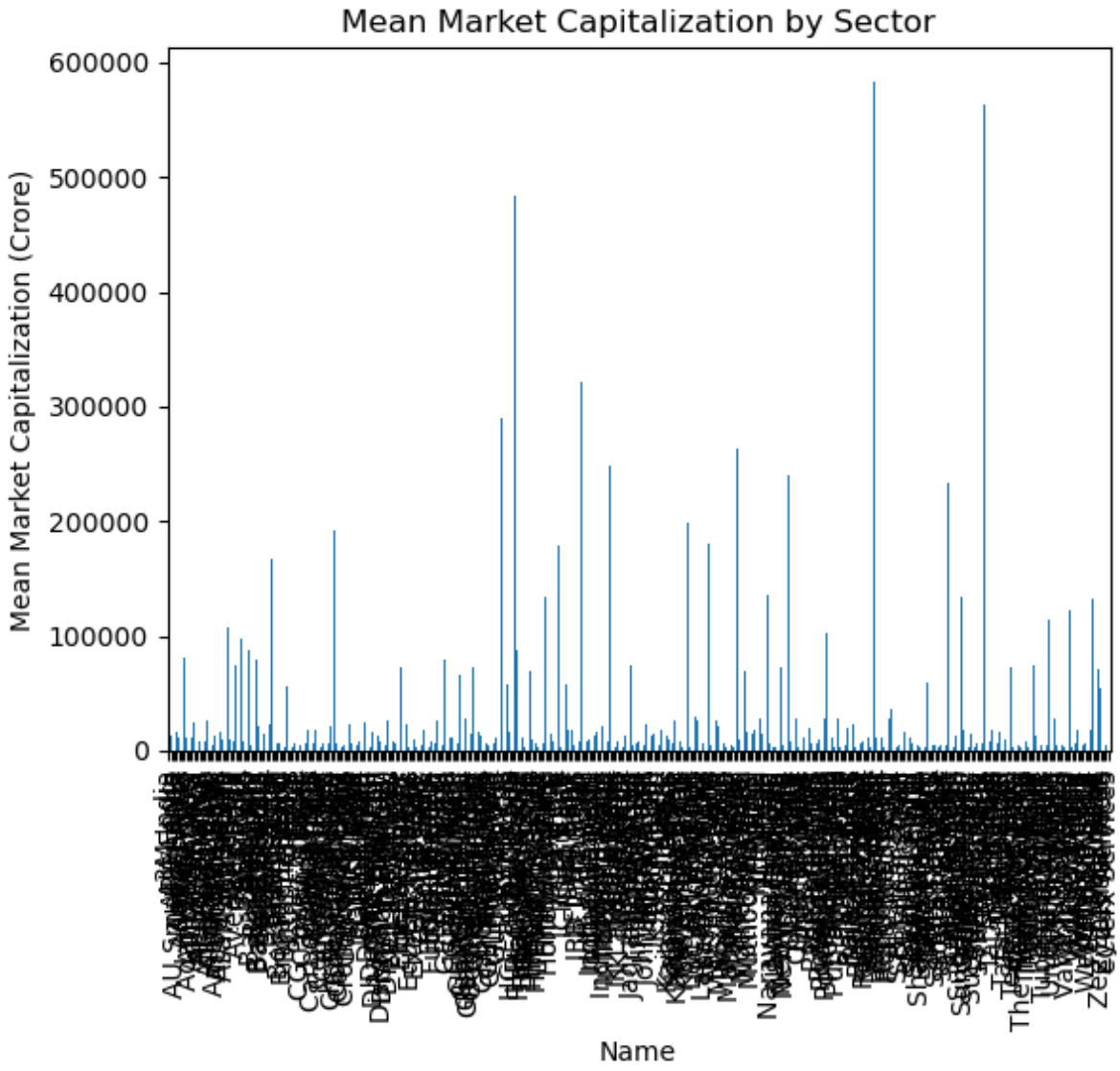
Out[12]:

	Mar Cap - Crore	Sales Qtr - Crore
count	365.000000	365.000000
mean	31300.970301	4395.976849
std	67224.641338	11092.206185
min	3017.070000	47.240000
25%	5089.870000	593.740000
50%	9097.330000	1278.300000
75%	21372.180000	2840.750000
max	583436.720000	110666.930000

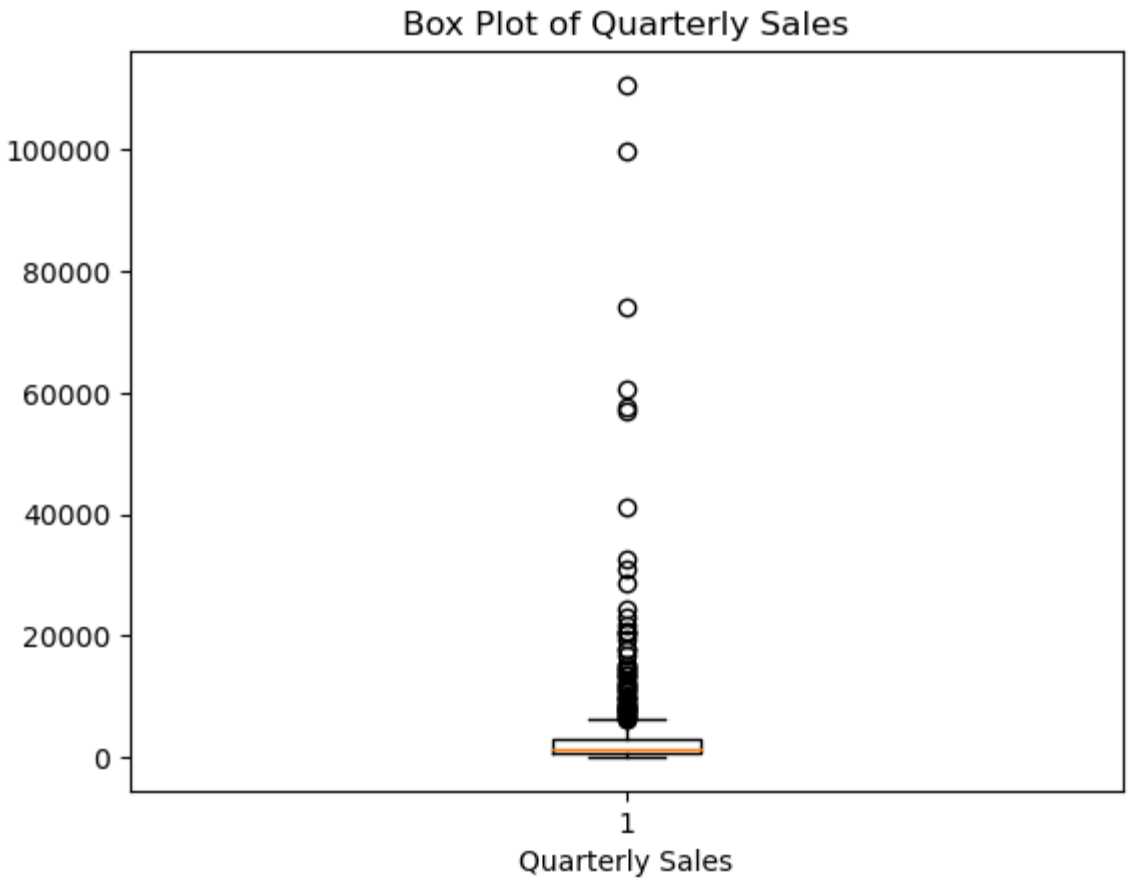
In [44]:
```
# Scatter plot of market capitalization vs. quarterly sales (contains market capita

plt.scatter(ef['Mar Cap - Crore'], ef ['Sales Qtr - Crore'])
plt.xlabel('Market Capitalization (Crore)')
plt.ylabel('Quarterly Sales (Crore)')
plt.title('Scatter Plot: Market Cap vs. Quarterly Sales')
plt.show()
```



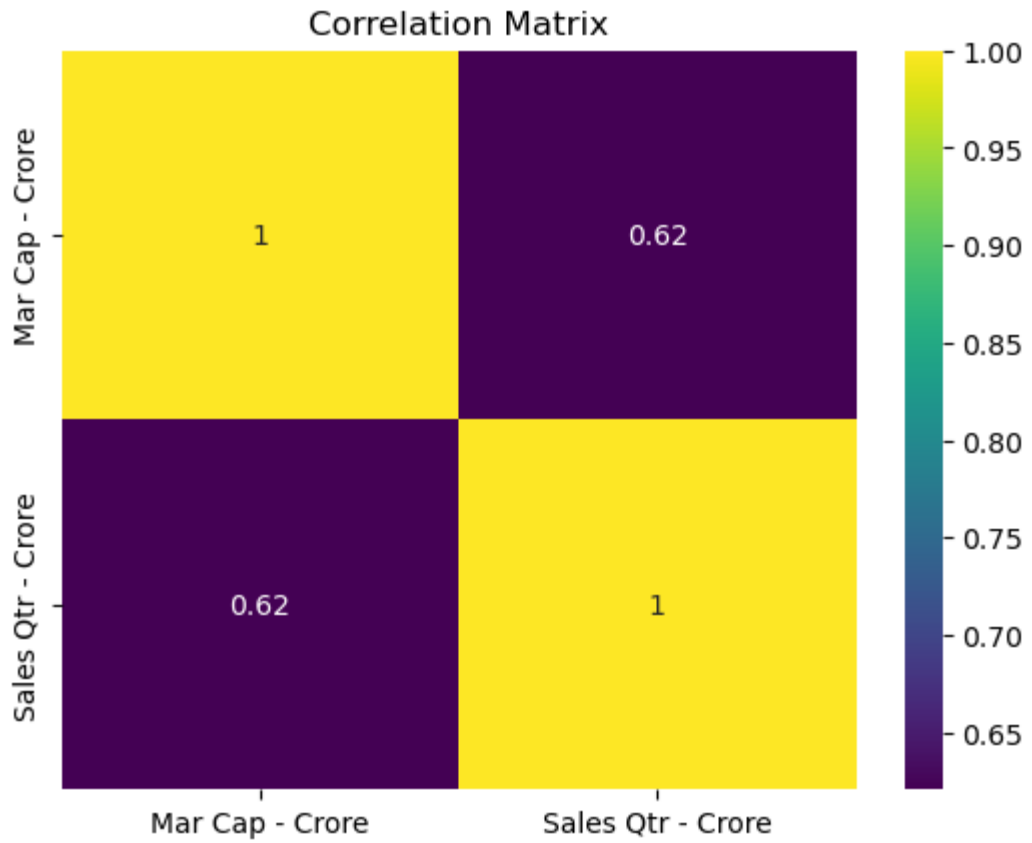
In [51]:
```
# Assuming 'data' contains sector information and market capitalization

sector_mean_cap = ef.groupby('Name')['Mar Cap - Crore'].mean()
sector_mean_cap.plot(kind='bar')
plt.xlabel('Name')
plt.ylabel('Mean Market Capitalization (Crore)')
plt.title('Mean Market Capitalization by Sector')
plt.show()
```



In [53]:
```
plt.boxplot(ef['Sales Qtr - Crore'])
plt.xlabel('Quarterly Sales')
plt.title('Box Plot of Quarterly Sales')
plt.show()
```



In [27]:
```
# Correlation matrix

correlation_matrix = ef.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='viridis')
plt.title('Correlation Matrix')
plt.show()
```



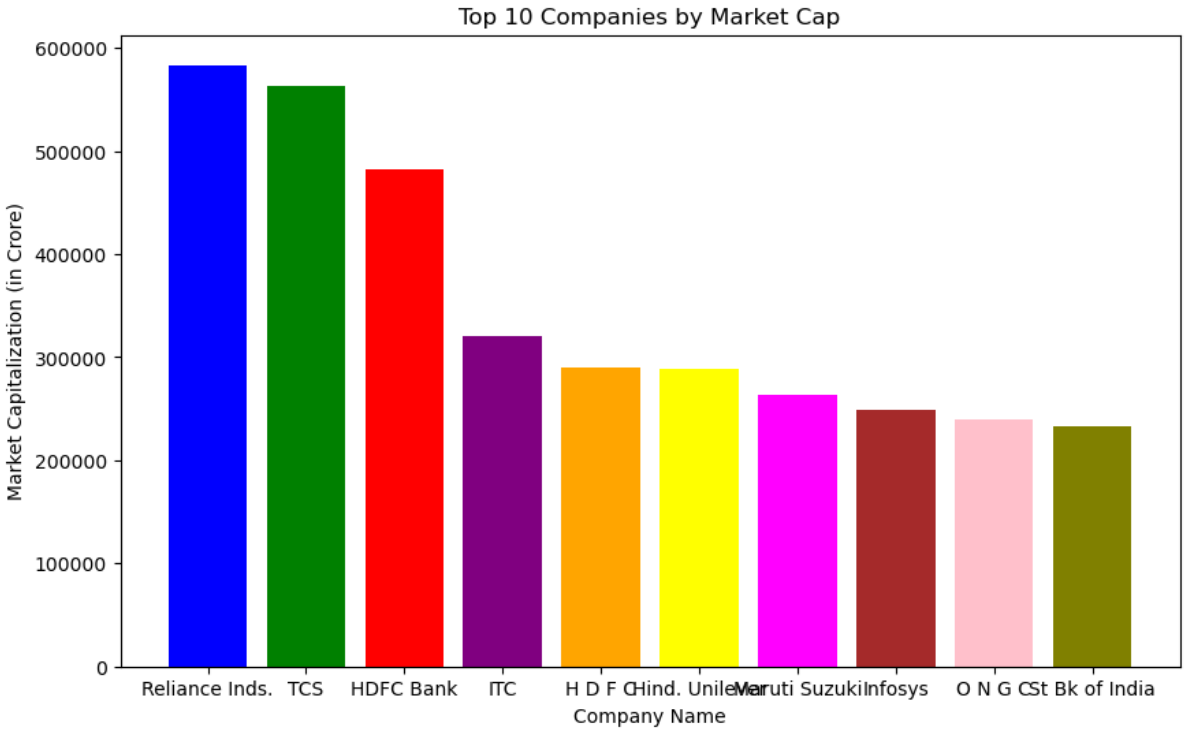
In [42]:
```
# Top companies based on market capitalization
top_market_cap = ef.nlargest(10, 'Mar Cap - Crore')
print("Top Companies by Market Capitalization:\n", top_market_cap)
```

```
Top Companies by Market Capitalization:
              Name  Mar Cap - Crore  Sales Qtr - Crore
0  Reliance Inds.         583436.72           99810.00
1             TCS         563709.84           30904.00
2       HDFC Bank         482953.59           20581.27
3             ITC         320985.27            9772.02
4         H D F C         289497.37           16840.51
5  Hind. Unilever         288265.26            8590.00
6   Maruti Suzuki         263493.81           19283.20
7         Infosys         248320.35           17794.00
8         O N G C         239981.50           22995.88
9  St Bk of India         232763.33           57014.08
```

In [43]:
```
# Top companies based on quarterly sales
top_sales = ef.nlargest(10, 'Sales Qtr - Crore')
print("Top Companies by Quarterly Sales:\n", top_sales)
```

```
Top Companies by Quarterly Sales:
                Name  Mar Cap - Crore  Sales Qtr - Crore
14           I O C L        178017.48          110666.93
0     Reliance Inds.        583436.72           99810.00
23       Tata Motors        117071.87           74156.07
27           B P C L         98278.00           60616.36
54           H P C L         58034.78           57474.25
9     St Bk of India        232763.33           57014.08
122   Rajesh Exports         23495.54           41304.84
40        Tata Steel         73376.14           32464.14
1                TCS        563709.84           30904.00
13   Larsen & Toubro        180860.74           28747.45
```

In [31]:
```
import matplotlib.pyplot as plt

Name = ef['Name'].head(10)
Market_Capital = ef['Mar Cap - Crore'].head(10)

# Define colors for the bars
colors = ['blue', 'green', 'red', 'purple', 'orange', 'yellow', 'magenta', 'brown',

# Figure Size
fig = plt.figure(figsize=(10, 6))

# Bar Plot with specified colors
plt.bar(Name, Market_Capital, color=colors)

# Labeling axes and title
plt.xlabel('Company Name')
plt.ylabel('Market Capitalization (in Crore)')
plt.title('Top 10 Companies by Market Cap')

# Show Plot
plt.show()
```

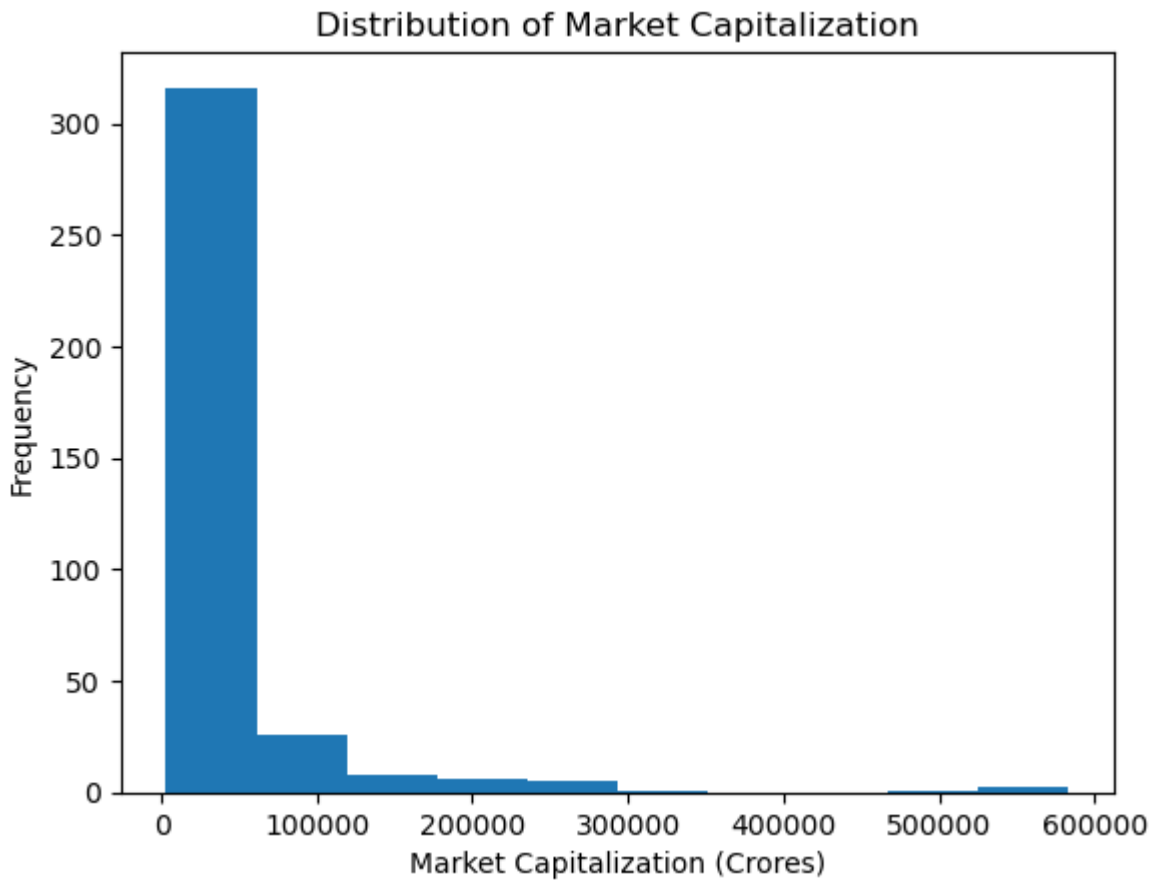


In [21]:
```
import numpy as np
import matplotlib.pyplot as plt

top_market_cap = ef.nlargest(10, 'Mar Cap - Crore')
top_Sales_Qtr = ef.nlargest(10, 'Sales Qtr - Crore')

X = top_market_cap['Name']
Y = top_market_cap['Mar Cap - Crore']
Z = top_Sales_Qtr['Sales Qtr - Crore']

X_axis = np.arange(len(X))

plt.bar(X_axis - 0.2, Y, 0.4, label='Mar Cap')
plt.bar(X_axis + 0.2, Z, 0.4, label='Sales Qtr')

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

plt.title("Top 10 Companies by Market Cap")
plt.xlabel("Company")
plt.ylabel("Amount (in Crore)")
plt.legend()
plt.tight_layout() # Adjust layout to prevent overlapping labels
plt.show()
```

In [36]:
```
summary_stats = ef.describe()
print("Summary Statistics:\n", summary_stats)
```

```
Summary Statistics:
        Mar Cap - Crore  Sales Qtr - Crore
count       365.000000         365.000000
mean      31300.970301        4395.976849
std       67224.641338       11092.206185
min        3017.070000          47.240000
25%        5089.870000         593.740000
50%        9097.330000        1278.300000
75%       21372.180000        2840.750000
max      583436.720000      110666.930000
```

In [35]:
```
plt.hist(ef['Mar Cap - Crore'], bins=10)
plt.xlabel('Market Capitalization (Crores)')
plt.ylabel('Frequency')
plt.title('Distribution of Market Capitalization')
plt.show()
```



In [40]:
```
# Correlation between market capitalization and quarterly sales
correlation = ef['Mar Cap - Crore'].corr(ef['Sales Qtr - Crore'])
print("Correlation between Market Cap and Quarterly Sales:", correlation)
```

```
Correlation between Market Cap and Quarterly Sales: 0.6207020390075659
```

In [41]:
```
# Regression analysis
import statsmodels.api as sm

X = ef['Mar Cap - Crore']
y = ef['Sales Qtr - Crore']

X = sm.add_constant(X)
model = sm.OLS(y, X).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:      Sales Qtr - Crore   R-squared:                       0.385
Model:                            OLS   Adj. R-squared:                  0.384
Method:                 Least Squares   F-statistic:                     227.5
Date:                Sun, 12 May 2024   Prob (F-statistic):           2.97e-40
Time:                        12:57:04   Log-Likelihood:                -3828.2
No. Observations:                 365   AIC:                             7660.
Df Residuals:                     363   BIC:                             7668.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1190.2212    502.952      2.366      0.018     201.156    2179.287
Mar Cap - Crore     0.1024      0.007     15.083      0.000       0.089       0.116
==============================================================================
Omnibus:                      440.425   Durbin-Watson:                   1.760
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            35153.388
Skew:                           5.499   Prob(JB):                         0.00
Kurtosis:                      49.803   Cond. No.                     8.17e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly sp
ecified.
[2] The condition number is large, 8.17e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [ ]:
```
```