Importing necessary libraries

In [ ]:

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

Loading the dataset

In [ ]:

```
import pandas as pd

# Load the dataset
file_path = '/content/Financial Analytics data.csv'
data = pd.read_csv(file_path)

# Display the first few rows
print("First few rows of the dataset:")
print(data.head())

# Get a summary of the DataFrame
print("\nDataFrame info:")
print(data.info())

# Check for missing values
print("\nMissing values in each column:")
print(data.isnull().sum())

# Display the columns to identify the empty column
print("\nColumns in the dataset:")
print(data.columns)

# Drop the empty column
# Assuming the empty column is named 'Unnamed: 4' as identified earlier
data = data.drop(columns=['Unnamed: 4'])

# Verify the column has been dropped
print("\nColumns after dropping the empty column:")
print(data.columns)

# Verify the changes
print("\nData types and non-null counts after cleaning:")
print(data.info())

print("\nSummary statistics after cleaning:")
print(data.describe())
```

```
First few rows of the dataset:
   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

DataFrame 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
None

Missing values in each column:
S.No.                  0
Name                   0
Mar Cap - Crore        9
Sales Qtr - Crore    123
Unnamed: 4           394
dtype: int64

Columns in the dataset:
Index(['S.No.', 'Name', 'Mar Cap - Crore', 'Sales Qtr - Crore', 'Unnamed: 4'], dtype='obj
ect')

Columns after dropping the empty column:
Index(['S.No.', 'Name', 'Mar Cap - Crore', 'Sales Qtr - Crore'], dtype='object')

Data types and non-null counts after cleaning:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 488 entries, 0 to 487
Data columns (total 4 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
dtypes: float64(2), int64(1), object(1)
memory usage: 15.4+ KB
None

Summary statistics after cleaning:
            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
```

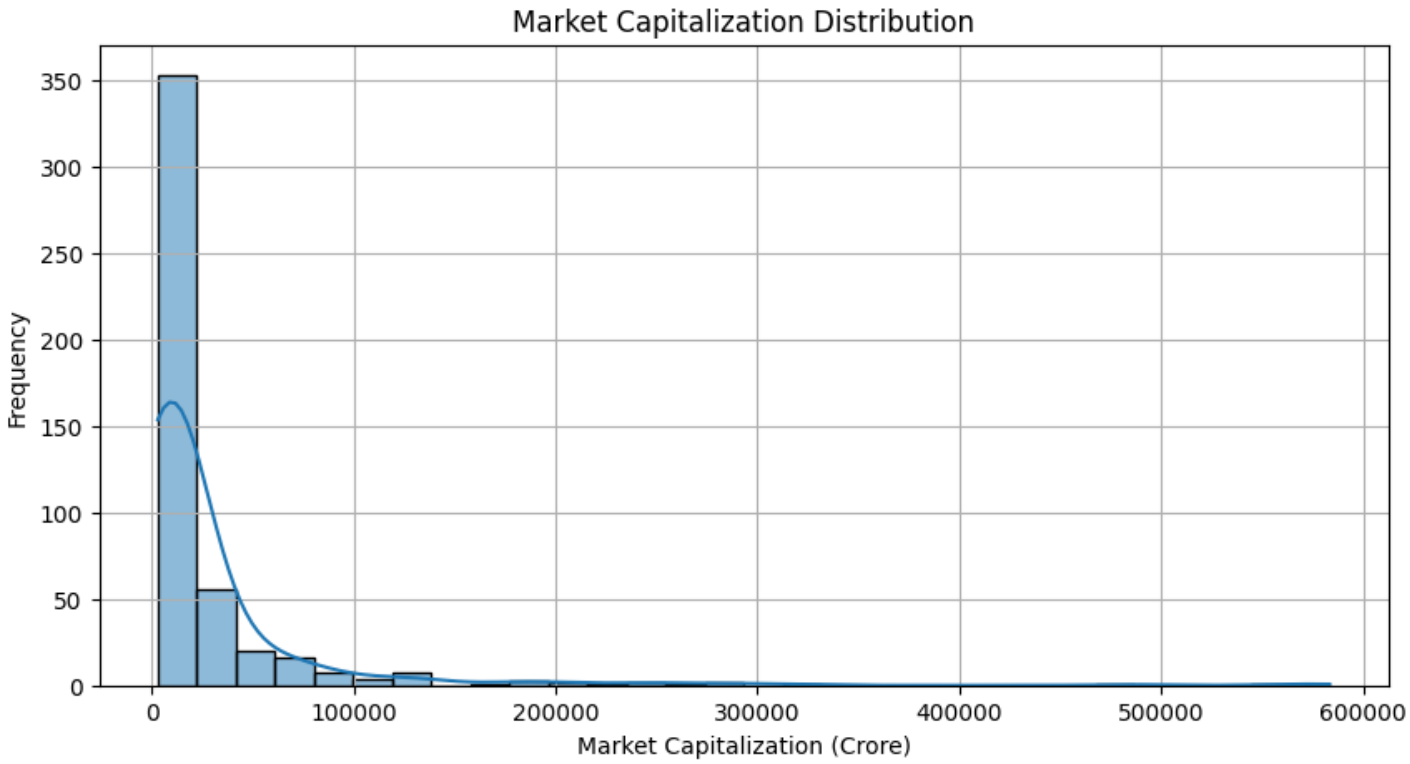
Exploratory Data Analysis (EDA)

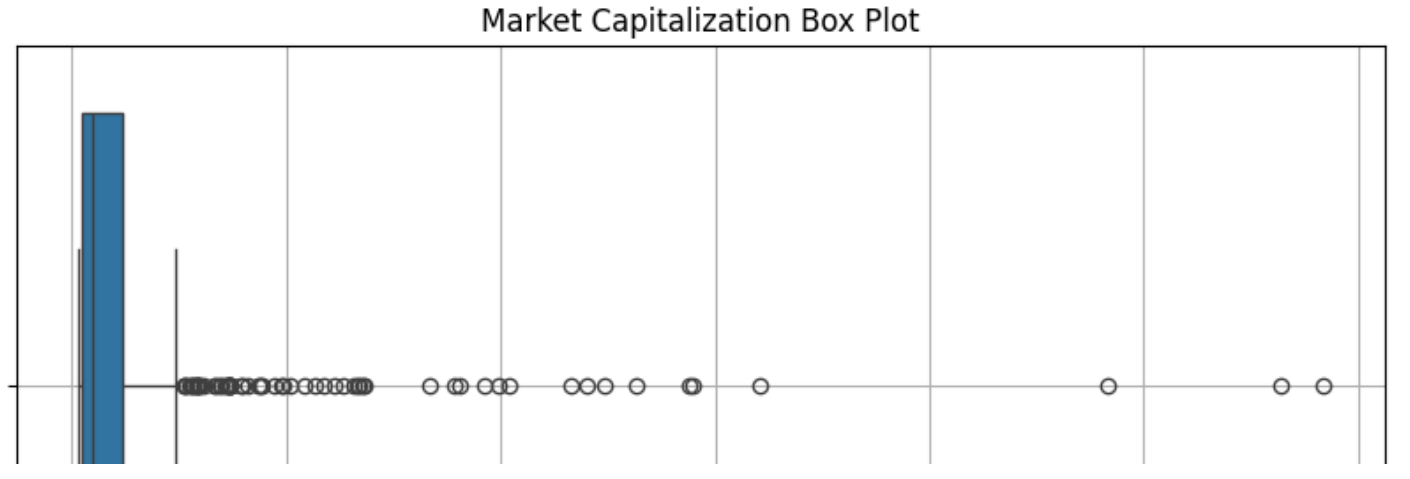
In [ ]:

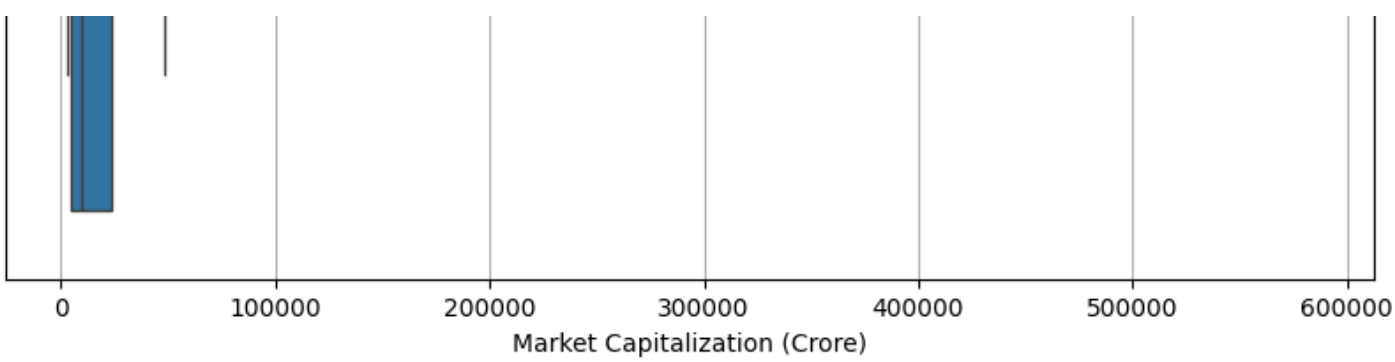
```
# Univariate Analysis: Market Capitalization
print("Summary Statistics for Market Capitalization")
print(data['Mar Cap - Crore'].describe())

# Histogram for Market Capitalization
plt.figure(figsize=(10, 5))
sns.histplot(data['Mar Cap - Crore'].dropna(), bins=30, kde=True)
plt.title('Market Capitalization Distribution')
plt.xlabel('Market Capitalization (Crore)')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()

# Boxplot for Market Capitalization
plt.figure(figsize=(10, 5))
sns.boxplot(x=data['Mar Cap - Crore'].dropna())
plt.title('Market Capitalization Box Plot')
plt.xlabel('Market Capitalization (Crore)')
plt.grid(True)
plt.show()

# Univariate Analysis: Quarterly Sales
```

```
print("\nSummary Statistics for Quarterly Sales")
print(data['Sales Qtr - Crore'].describe())

# Histogram for Quarterly Sales
plt.figure(figsize=(10, 5))
sns.histplot(data['Sales Qtr - Crore'].dropna(), bins=30, kde=True)
plt.title('Quarterly Sales Distribution')
plt.xlabel('Quarterly Sales (Crore)')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()

# Boxplot for Quarterly Sales
plt.figure(figsize=(10, 5))
sns.boxplot(x=data['Sales Qtr - Crore'].dropna())
plt.title('Quarterly Sales Box Plot')
plt.xlabel('Quarterly Sales (Crore)')
plt.grid(True)
plt.show()
```

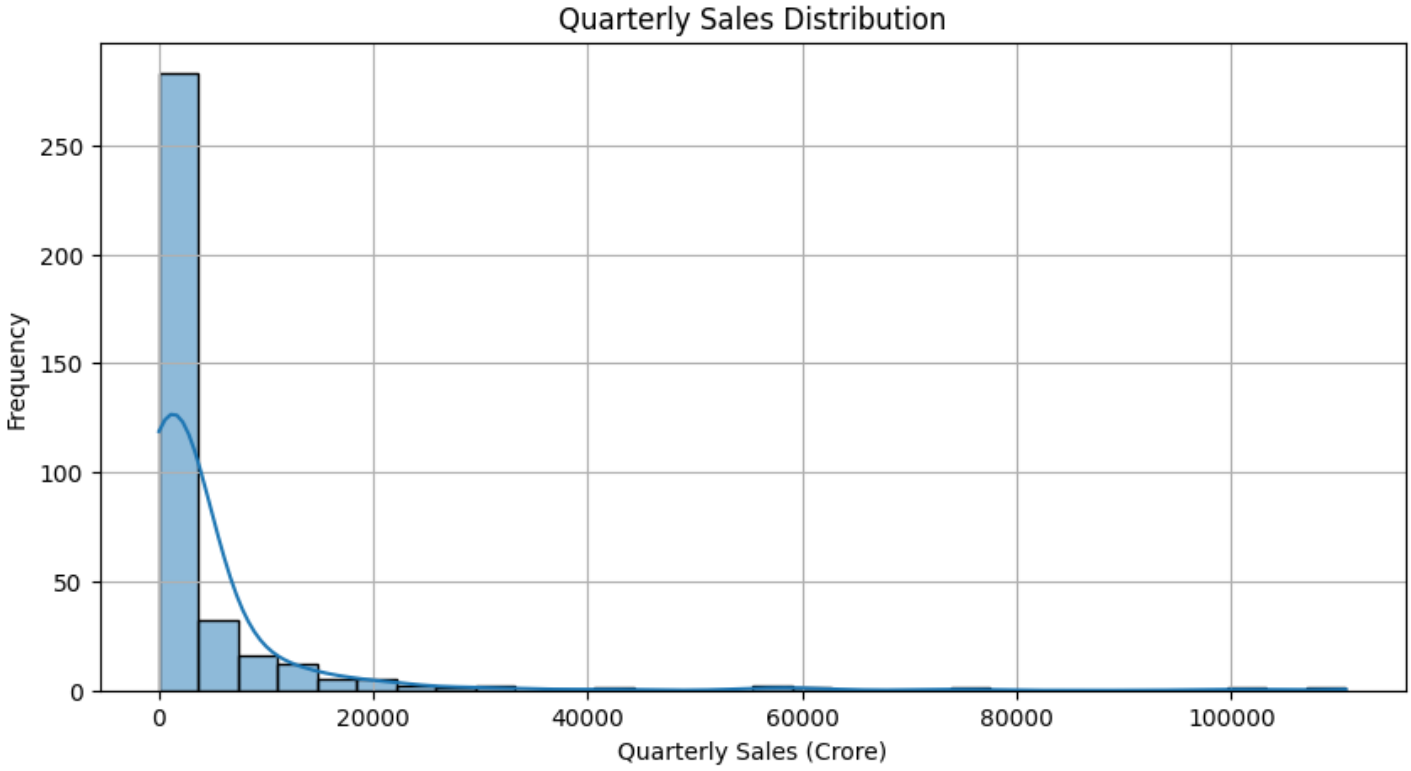
```
Summary Statistics for Market Capitalization
count       479.000000
mean      28043.857119
std       59464.615831
min        3017.070000
25%        4843.575000
50%        9885.050000
75%       23549.900000
max      583436.720000
Name: Mar Cap - Crore, dtype: float64
```

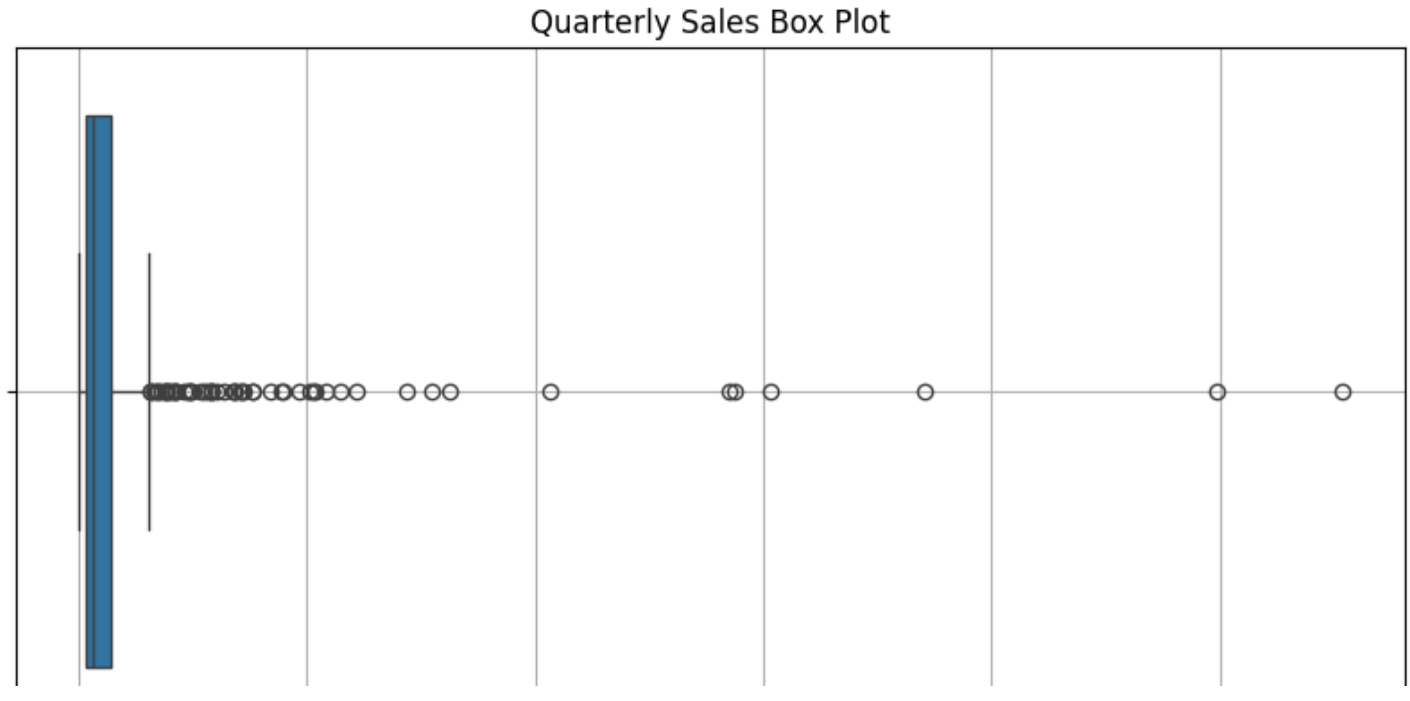







```
Summary Statistics for Quarterly Sales
count       365.000000
mean       4395.976849
std       11092.206185
min          47.240000
25%         593.740000
50%        1278.300000
75%        2840.750000
max      110666.930000
Name: Sales Qtr - Crore, dtype: float64
```



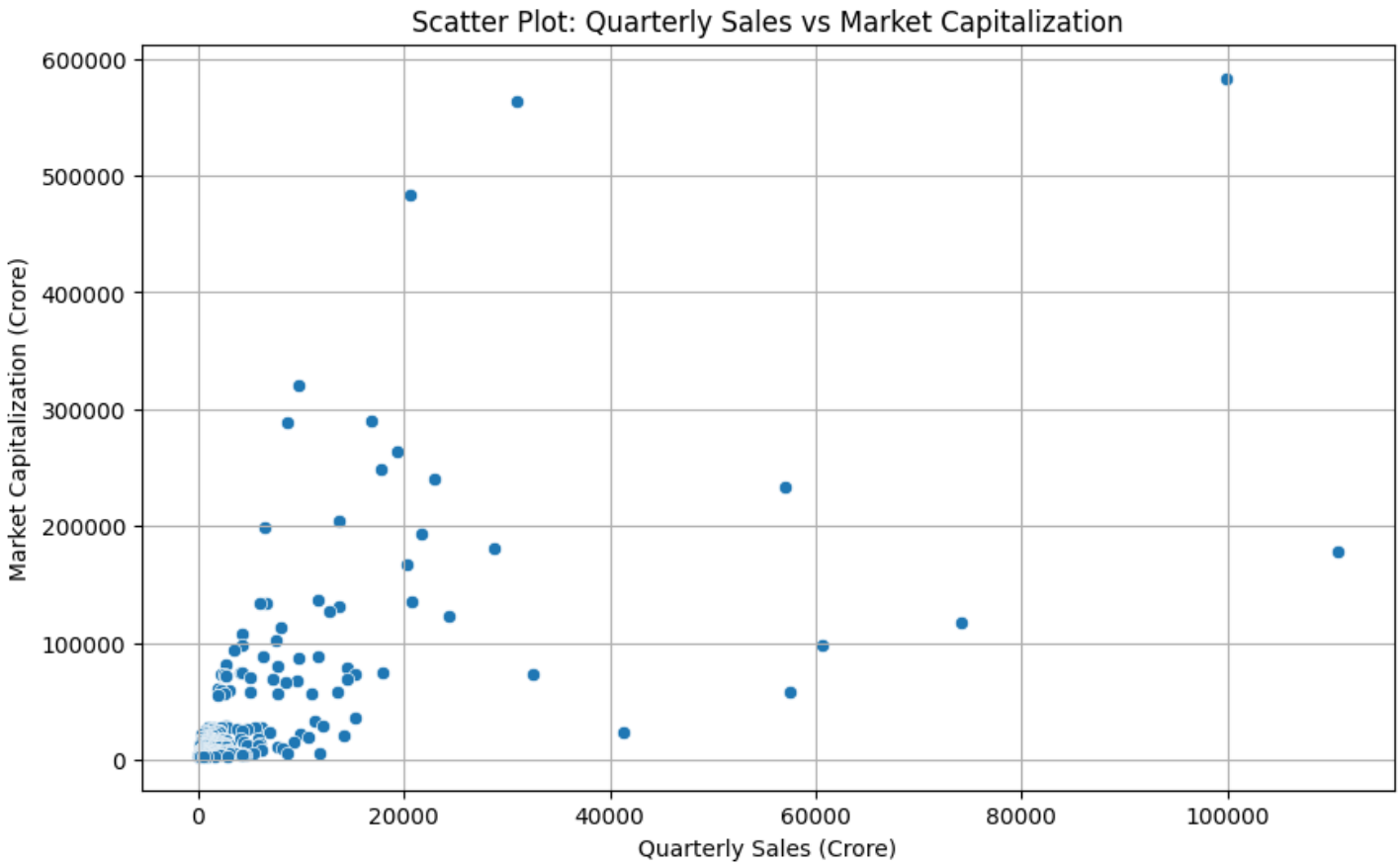


Bivariate

In [ ]:

```
# Bivariate Analysis
# 1. Correlation Analysis
correlation_matrix = data[['Mar Cap - Crore', 'Sales Qtr - Crore']].corr()
print("Correlation Matrix:")
print(correlation_matrix)

# 2. Scatter Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Sales Qtr - Crore', y='Mar Cap - Crore', data=data)
plt.title('Scatter Plot: Quarterly Sales vs Market Capitalization')
plt.xlabel('Quarterly Sales (Crore)')
plt.ylabel('Market Capitalization (Crore)')
plt.grid(True)
plt.show()
```

```
Correlation Matrix:
                  Mar Cap - Crore  Sales Qtr - Crore
Mar Cap - Crore          1.000000           0.620702
Sales Qtr - Crore        0.620702           1.000000
```



Calculating the Mean, Median and Standard deviation

In [ ]:

```
# Calculate the mean, median, and standard deviation for 'Mar Cap - Crore'
mean_mar_cap = data['Mar Cap - Crore'].mean()
median_mar_cap = data['Mar Cap - Crore'].median()
std_mar_cap = data['Mar Cap - Crore'].std()

# Calculate the mean, median, and standard deviation for 'Sales Qtr - Crore'
mean_sales_qtr = data['Sales Qtr - Crore'].mean()
```

```
median_sales_qtr = data['Sales Qtr - Crore'].median()
std_sales_qtr = data['Sales Qtr - Crore'].std()

# Print the results
print(f"Mean Market Cap: {mean_mar_cap}")
print(f"Median Market Cap: {median_mar_cap}")
print(f"Standard Deviation of Market Cap: {std_mar_cap}")

print(f"\nMean Quarterly Sales: {mean_sales_qtr}")
print(f"Median Quarterly Sales: {median_sales_qtr}")
print(f"Standard Deviation of Quarterly Sales: {std_sales_qtr}")
```

```
Mean Market Cap: 28043.857118997912
Median Market Cap: 9885.05
Standard Deviation of Market Cap: 59464.615831020186

Mean Quarterly Sales: 4395.976849315068
Median Quarterly Sales: 1278.3
Standard Deviation of Quarterly Sales: 11092.206185492805
```

OBSERVATION

Ultimately, the analysis gives you a big picture of your dataset by uncovering major metrics and their distributions. The correlation plots can show how market cap may be related to quarterly sales and visualizations help in the context of spread options, letting you see when something is just too out there. Analysis: This is very important for the trend and with this data, these companies can make an analysis on where they are standing in the market competition