In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
```

In [3]:
```python
df = pd.read_csv("D:/Unified_Internship/Project 2 - Financial Analytics/Financial Analyt
```

In [4]:
```python
df.head()
```

Out[4]:

	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

In [7]:
```python
fdf=df
```

In [8]:
```python
fdf.head()
```

Out[8]:

	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

Data Exploration and Understanding

In [10]:
```python
fdf.info()
```

```
<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  459 non-null    float64
dtypes: float64(2), int64(1), object(1)
memory usage: 15.4+ KB
```

In [11]:
```python
fdf.describe()
```

Out[11]:

	S.No.	Mar Cap - Crore	Sales Qtr - Crore
count	488.000000	479.000000	459.000000
mean	251.508197	28043.857119	3807.789412
std	145.884078	59464.615831	9989.449987
min	1.000000	3017.070000	0.000000
25%	122.750000	4843.575000	534.910000
50%	252.500000	9885.050000	1137.170000
75%	378.250000	23549.900000	2730.195000
max	500.000000	583436.720000	110666.930000

In [13]:

```
fdf.duplicated().sum()
```

Out[13]:

```
0
```

In [14]:

```
fdf.isnull().sum ()
```

Out[14]:

```
S.No.                 0
Name                  0
Mar Cap - Crore       9
Sales Qtr - Crore    29
dtype: int64
```

In [16]:

```
nan_mask = fdf['Sales Qtr - Crore'].isna()

# Filter rows where 'Unnamed: 4' has numerical values corresponding to NaN in 'Sales Qtr
#verification_df = fdf[nan_mask & pd.to_numeric(fdf['Unnamed: 4'], errors='coerce').notn

# Display the verification DataFrame
#print(verification_df)
```

Data Cleaning and Preprocessing

In [17]:

```
fdf.fillna(0,inplace=True)
fdf
```

Out[17]:

	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
...	...	...	...	...
483	496	Lak. Vilas Bank	3029.57	790.17
484	497	NOCIL	3026.26	249.27
485	498	Orient Cement	3024.32	511.53
486	499	Natl.Fertilizer	3017.07	2840.75
487	500	L T Foods	0.00	0.00

488 rows × 4 columns

In [21]:

```
#Indexes with both zero Mar Cap - Crore and Sale Qtr (in Cr)
fdf[(fdf['Mar Cap - Crore'] == 0) & (fdf['Sales Qtr - Crore'] == 0)].index
```

Out[21]:

```
Int64Index([99, 147, 193, 243, 287, 337, 387, 437, 487], dtype='int64')
```

In [22]:

```
#dropping Company data having both zero Mar Cap - Crore and Sale Qtr (in Cr)
fdf = fdf.drop(fdf[(fdf['Mar Cap - Crore'] == 0) & (fdf['Sales Qtr - Crore'] == 0)].inde
fdf
```

Out[22]:

	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
...	...	...	...	...
482	495	Prime Focus	3031.50	609.61
483	496	Lak. Vilas Bank	3029.57	790.17
484	497	NOCIL	3026.26	249.27
485	498	Orient Cement	3024.32	511.53
486	499	Natl.Fertilizer	3017.07	2840.75

479 rows × 4 columns

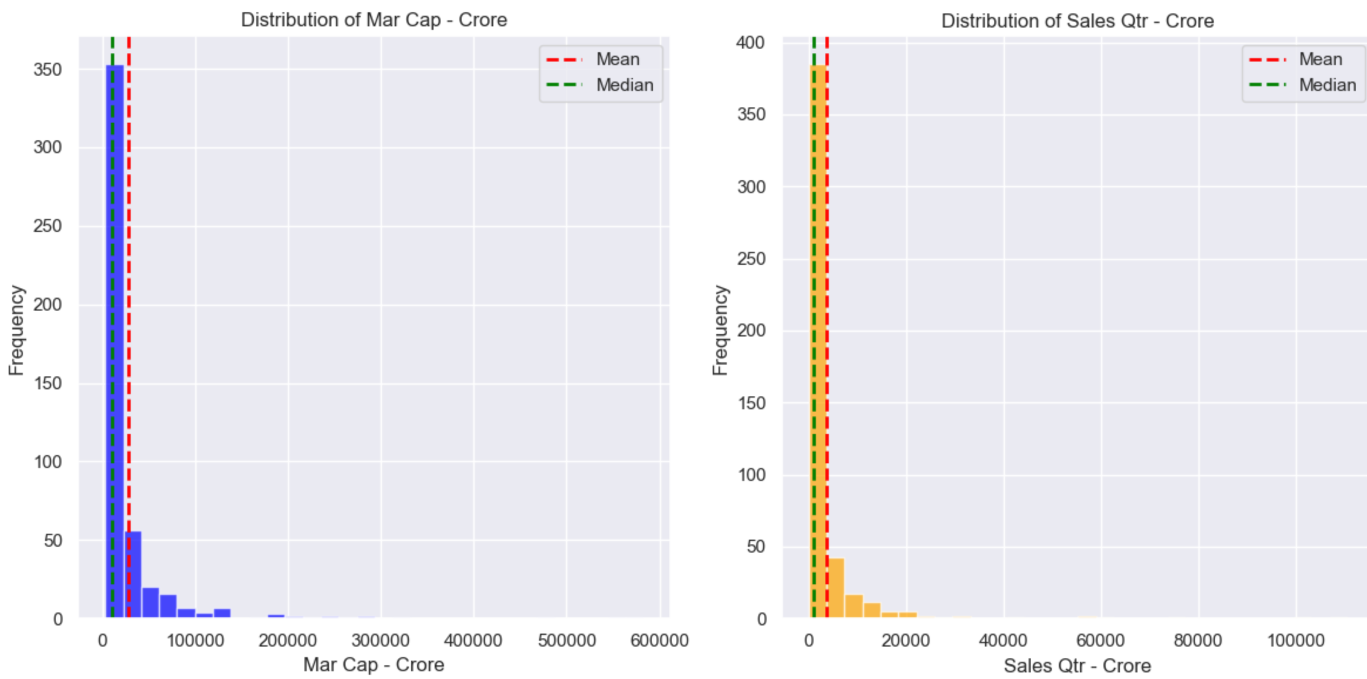
In [23]:

```
feature1 = 'Mar Cap - Crore'
feature2 = 'Sales Qtr - Crore'

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

plt.subplot(1, 2, 1)
plt.hist(fdf[feature1].dropna(), bins=30, color='blue', alpha=0.7)
plt.title(f'Distribution of {feature1}')
```



```
eature1)
```

```
plt.ylabel('Frequency')
plt.axvline(fdf[feature1].mean(), color='red', linestyle='dashed', linewidth=2, label='M
plt.axvline(fdf[feature1].median(), color='green', linestyle='dashed', linewidth=2, labe
plt.legend()

plt.subplot(1, 2, 2)
plt.hist(fdf[feature2].dropna(), bins=30, color='orange', alpha=0.7)
plt.title(f'Distribution of {feature2}')
plt.xlabel(feature2)
plt.ylabel('Frequency')
plt.axvline(fdf[feature2].mean(), color='red', linestyle='dashed', linewidth=2, label='M
plt.axvline(fdf[feature2].median(), color='green', linestyle='dashed', linewidth=2, labe
plt.legend()

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



In [24]:
```
fdf_d = fdf[(fdf['Mar Cap - Crore'] == 0) & (fdf['Sales Qtr - Crore'] == 0)]
fdf_d
```

Out[24]:

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

In [25]:
```
fdf
```

Loading [MathJax]/extensions/Safe.js

Out[25]:

	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
...	...	...	...	...
482	495	Prime Focus	3031.50	609.61
483	496	Lak. Vilas Bank	3029.57	790.17
484	497	NOCIL	3026.26	249.27
485	498	Orient Cement	3024.32	511.53
486	499	Natl.Fertilizer	3017.07	2840.75

479 rows × 4 columns

In [26]:

```
# Count zero entries in each column
zero_counts = (fdf == 0).sum()
print(zero_counts)
```

```
S.No.                 0
Name                  0
Mar Cap - Crore       0
Sales Qtr - Crore    21
dtype: int64
```

In [27]:

```
fdf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 479 entries, 0 to 486
Data columns (total 4 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   S.No.              479 non-null    int64  
 1   Name               479 non-null    object 
 2   Mar Cap - Crore    479 non-null    float64
 3   Sales Qtr - Crore  479 non-null    float64
dtypes: float64(2), int64(1), object(1)
memory usage: 18.7+ KB
```

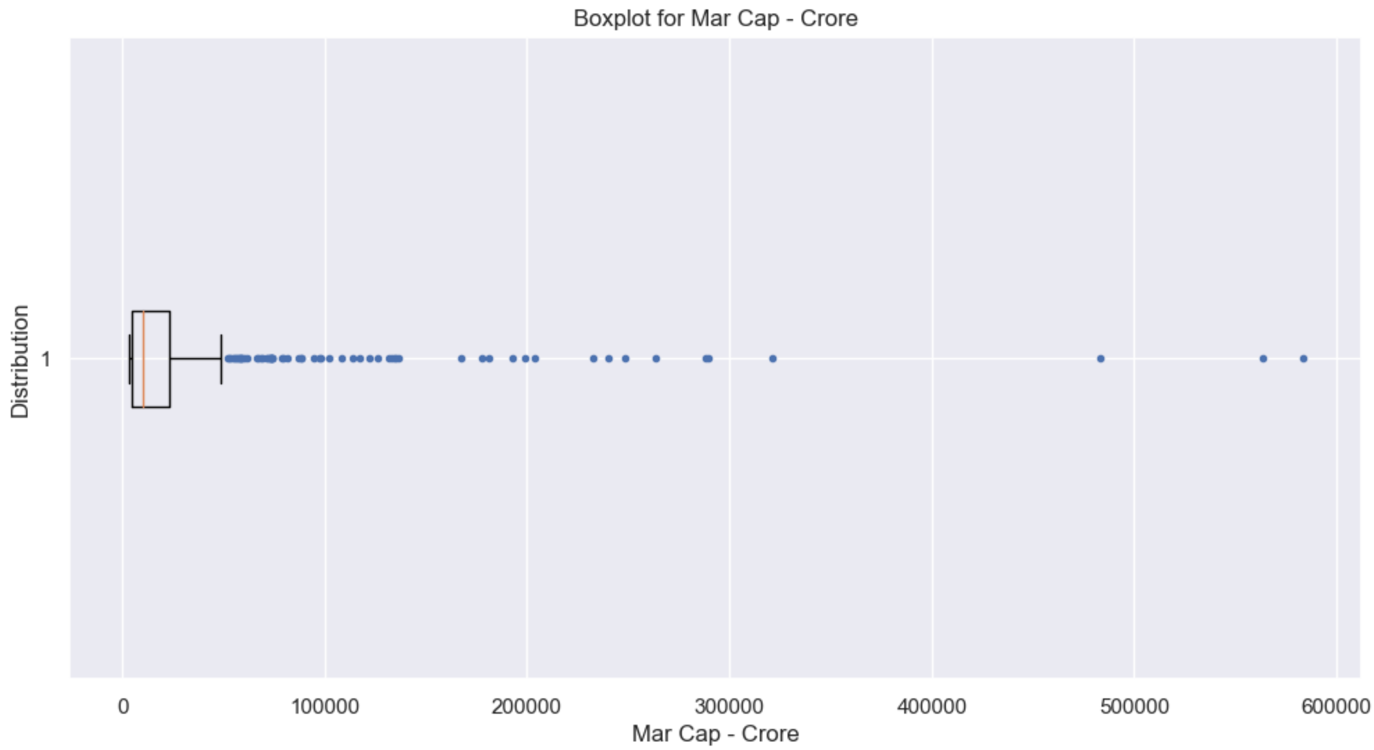
In [28]:

```
fdf.describe()
```

Out[28]:

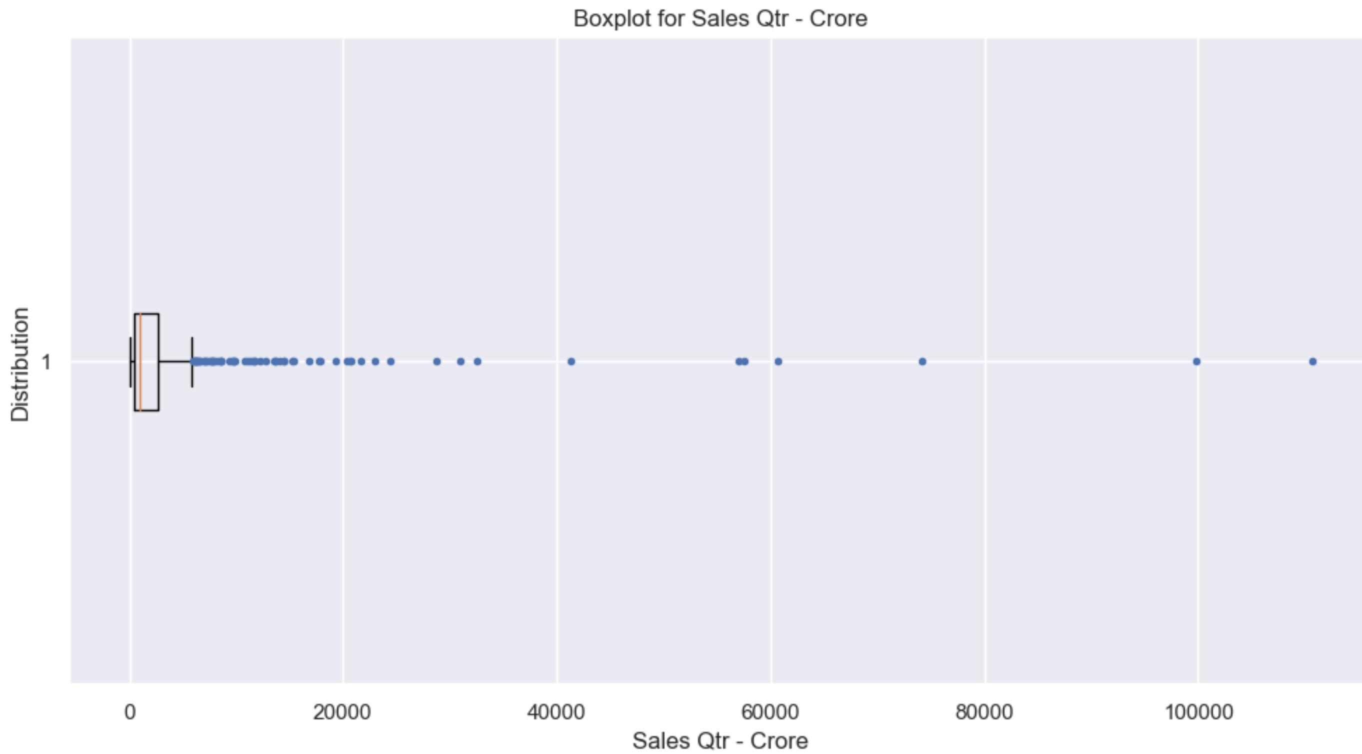
	S.No.	Mar Cap - Crore	Sales Qtr - Crore
count	479.000000	479.000000	479.000000
mean	250.597077	28043.857119	3648.800292
std	146.027260	59464.615831	9807.913520
min	1.000000	3017.070000	0.000000
25%	121.500000	4843.575000	473.595000
50%	249.000000	9885.050000	1012.940000
75%	377.500000	23549.900000	2630.235000
max	499.000000	583436.720000	110666.930000

In [29]:
```python
# Create a boxplot for Mar Cap - Crore
column_name = 'Mar Cap - Crore'
plt.figure(figsize=(12, 6))
plt.boxplot(fdf[column_name], vert=False, sym='b.')
plt.title(f'Boxplot for {column_name}')
plt.xlabel(column_name)
plt.ylabel('Distribution')
plt.show()
```



In [30]:
```python
# Create a boxplot for Sale Qtr (in Cr)
column_name = 'Sales Qtr - Crore'
plt.figure(figsize=(12, 6))
plt.boxplot(fdf[column_name], vert=False, sym='b.')
plt.title(f'Boxplot for {column_name}')
plt.xlabel(column_name)
plt.ylabel('Distribution')
plt.show()
```

Loading [MathJax]/extensions/Safe.js



In [31]:
```
fdf_d = fdf[(fdf['Mar Cap - Crore'] != 0) & (fdf['Sales Qtr - Crore'] == 0)]
fdf_d
```

Out[31]:

	S.No.	Name	Mar Cap - Crore	Sales Qtr - Crore
49	50	Bharti Infra.	61776.92	0.0
171	176	Info Edg.(India)	14845.05	0.0
185	192	Max Financial	13401.76	0.0
224	231	Bombay Burmah	10864.53	0.0
241	248	Sundaram Clayton	10074.36	0.0
258	271	Mahindra CIE	8587.04	0.0
314	327	Prism Cement	6176.23	0.0
332	345	GE Power	5497.40	0.0
338	351	MMTC	5300.00	0.0
370	383	Swan Energy	4721.49	0.0
374	387	Shoppers St.	4558.06	0.0
379	392	Stand.Chart.PLC	4487.31	0.0
393	406	Ujjivan Fin.Ser.	4293.42	0.0
396	409	Jindal Saw	4278.31	0.0
398	411	Linde India	4198.33	0.0
409	422	JP Associates	4074.37	0.0
418	431	HMT	3973.50	0.0
424	437	Gayatri Projects	3835.73	0.0
446	459	JP Power Ven.	3597.60	0.0
451	464	Amber Enterp.	3529.87	0.0
459	472	Hind.Construct.	3452.57	0.0

Feature Engineering

In [32]:

```
# Define a function to calculate Market Cap-to-Sales Ratio
def market_cap_to_sales_ratio(row):
    if row['Sales Qtr - Crore'] != 0:
        return row['Mar Cap - Crore'] / row['Sales Qtr - Crore']
    else:
        return 0  # Handle division by zero scenario

# Apply the function to create the new column
fdf['Market Cap-to-Sales Ratio'] = fdf.apply(market_cap_to_sales_ratio, axis=1)
fdf
```

Out[32]:

	S.No.	Name	Mar Cap - Crore	Sales Qtr - Crore	Market Cap-to-Sales Ratio
0	1	Reliance Inds.	583436.72	99810.00	5.845474
1	2	TCS	563709.84	30904.00	18.240676
2	3	HDFC Bank	482953.59	20581.27	23.465685
3	4	ITC	320985.27	9772.02	32.847382
4	5	H D F C	289497.37	16840.51	17.190535
...	...	...	...	...	...
482	495	Prime Focus	3031.50	609.61	4.972851
483	496	Lak. Vilas Bank	3029.57	790.17	3.834074
484	497	NOCIL	3026.26	249.27	12.140490
485	498	Orient Cement	3024.32	511.53	5.912302
486	499	Natl.Fertilizer	3017.07	2840.75	1.062068

479 rows × 5 columns

In [33]:

```
fdf_d = fdf[(fdf['Sales Qtr - Crore'] == 0)]
fdf_d
```

Out[33]:

	S.No.	Name	Mar Cap - Crore	Sales Qtr - Crore	Market Cap-to-Sales Ratio
49	50	Bharti Infra.	61776.92	0.0	0.0
171	176	Info Edg.(India)	14845.05	0.0	0.0
185	192	Max Financial	13401.76	0.0	0.0
224	231	Bombay Burmah	10864.53	0.0	0.0
241	248	Sundaram Clayton	10074.36	0.0	0.0
258	271	Mahindra CIE	8587.04	0.0	0.0
314	327	Prism Cement	6176.23	0.0	0.0
332	345	GE Power	5497.40	0.0	0.0
338	351	MMTC	5300.00	0.0	0.0
370	383	Swan Energy	4721.49	0.0	0.0
374	387	Shoppers St.	4558.06	0.0	0.0
379	392	Stand.Chart.PLC	4487.31	0.0	0.0
393	406	Ujjivan Fin.Ser.	4293.42	0.0	0.0
396	409	Jindal Saw	4278.31	0.0	0.0
398	411	Linde India	4198.33	0.0	0.0
409	422	JP Associates	4074.37	0.0	0.0
418	431	HMT	3973.50	0.0	0.0
424	437	Gayatri Projects	3835.73	0.0	0.0
446	459	JP Power Ven.	3597.60	0.0	0.0
451	464	Amber Enterp.	3529.87	0.0	0.0
459	472	Hind.Construct.	3452.57	0.0	0.0

In [34]:

```
fdf.sort_values('Market Cap-to-Sales Ratio',ascending=False)
fdf
```

Out[34]:

	S.No.	Name	Mar Cap - Crore	Sales Qtr - Crore	Market Cap-to-Sales Ratio
0	1	Reliance Inds.	583436.72	99810.00	5.845474
1	2	TCS	563709.84	30904.00	18.240676
2	3	HDFC Bank	482953.59	20581.27	23.465685
3	4	ITC	320985.27	9772.02	32.847382
4	5	H D F C	289497.37	16840.51	17.190535
...	...	...	...	...	...
482	495	Prime Focus	3031.50	609.61	4.972851
483	496	Lak. Vilas Bank	3029.57	790.17	3.834074
484	497	NOCIL	3026.26	249.27	12.140490
485	498	Orient Cement	3024.32	511.53	5.912302
486	499	Natl.Fertilizer	3017.07	2840.75	1.062068

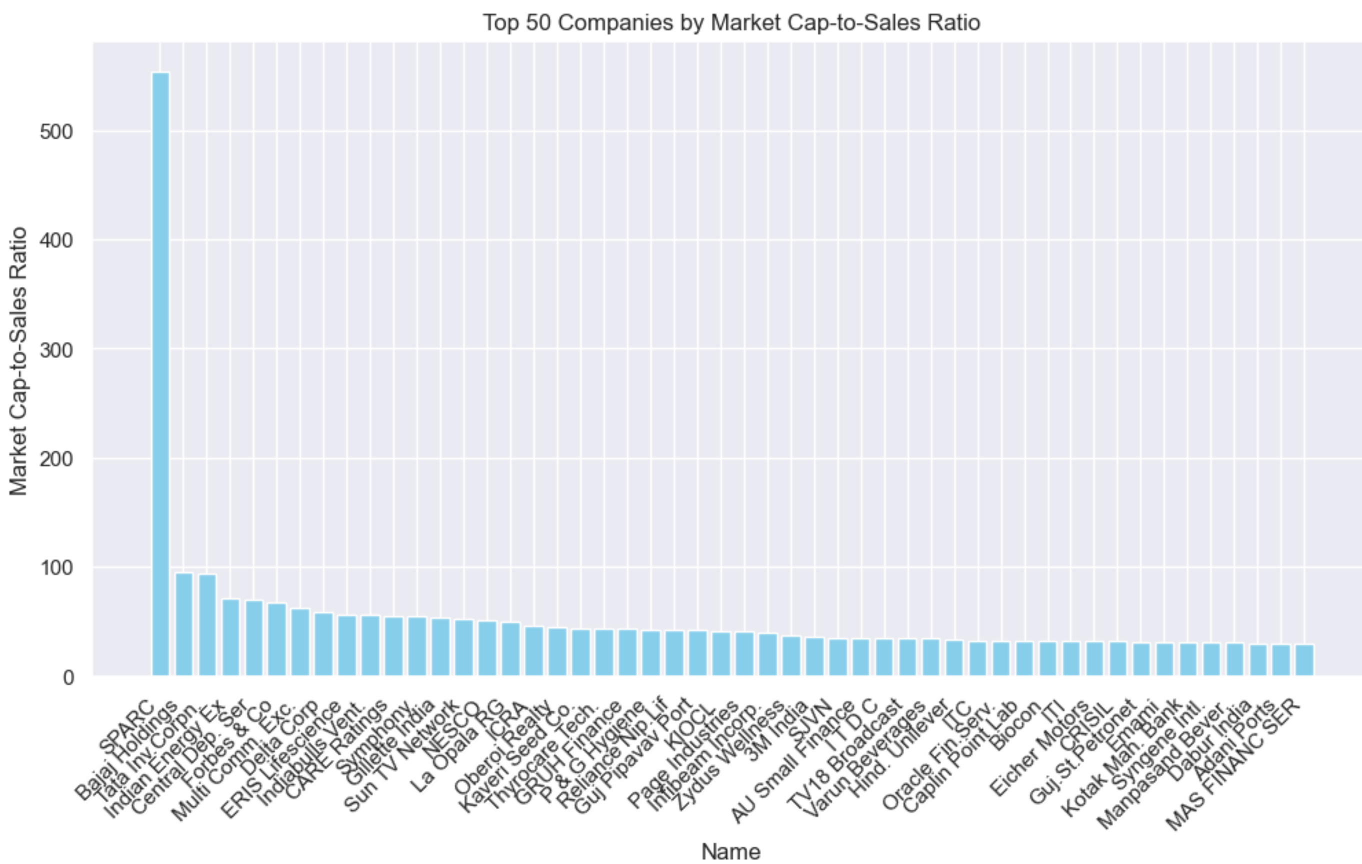
479 rows × 5 columns

In [35]:

```
# Specify the top N companies
```

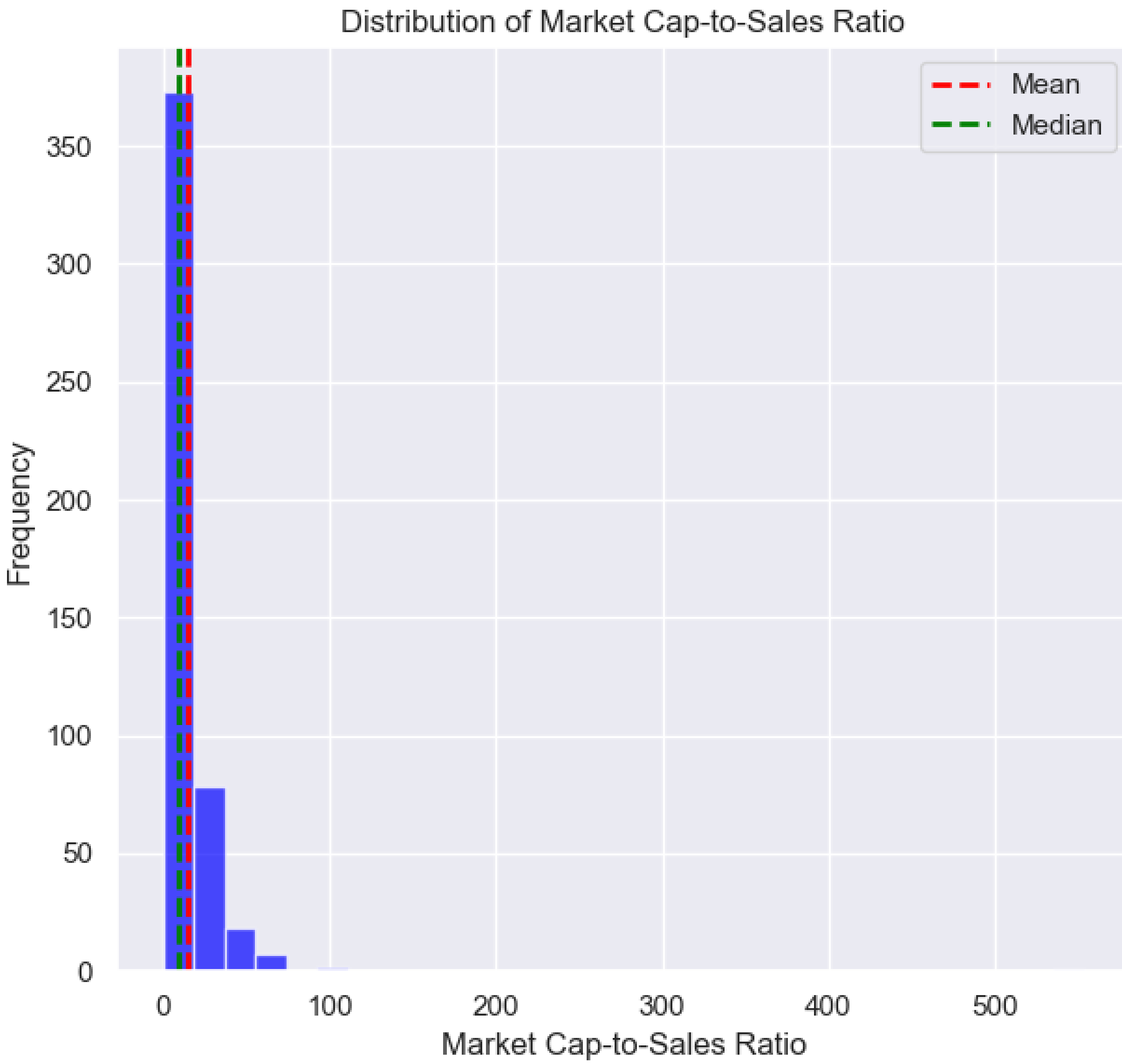
Loading [MathJax]/extensions/Safe.js

```python
# Select the top N companies based on Market Cap-to-Sales Ratio
top_companies = fdf.nlargest(top_n, 'Market Cap-to-Sales Ratio')

# Create a bar plot
plt.figure(figsize=(12, 6))
plt.bar(top_companies['Name'], top_companies['Market Cap-to-Sales Ratio'], color='skyblu
plt.title(f'Top {top_n} Companies by Market Cap-to-Sales Ratio')
plt.xlabel('Name')
plt.ylabel('Market Cap-to-Sales Ratio')
plt.xticks(rotation=45, ha='right')
plt.show()
```



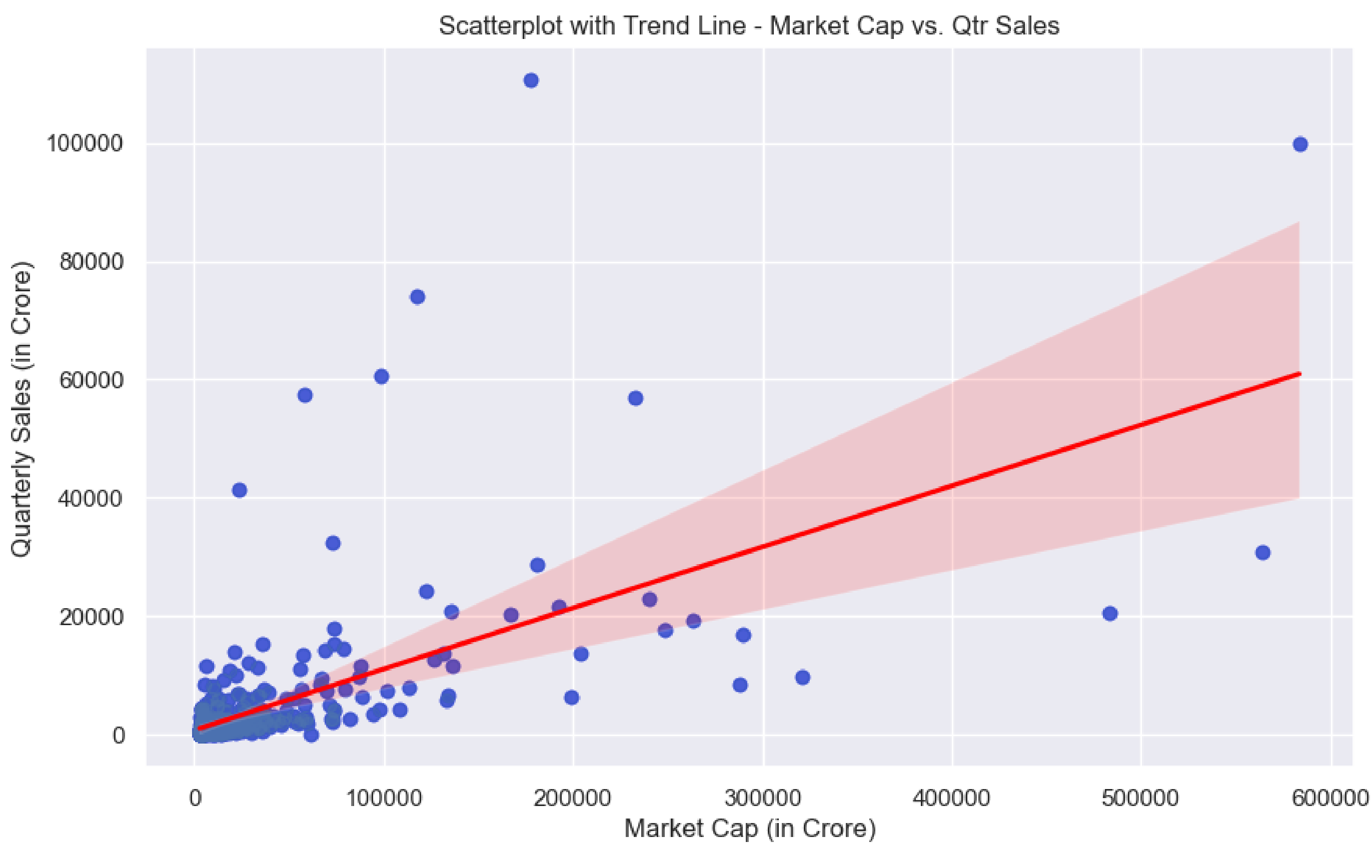
In [36]:

```python
feature = 'Market Cap-to-Sales Ratio'

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

plt.subplot(1, 2, 1)
plt.hist(fdf[feature].dropna(), bins=30, color='blue', alpha=0.7)
plt.title(f'Distribution of {feature}')
plt.xlabel(feature)
plt.ylabel('Frequency')
plt.axvline(fdf[feature].mean(), color='red', linestyle='dashed', linewidth=2, label='Me
plt.axvline(fdf[feature].median(), color='green', linestyle='dashed', linewidth=2, label
plt.legend()

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

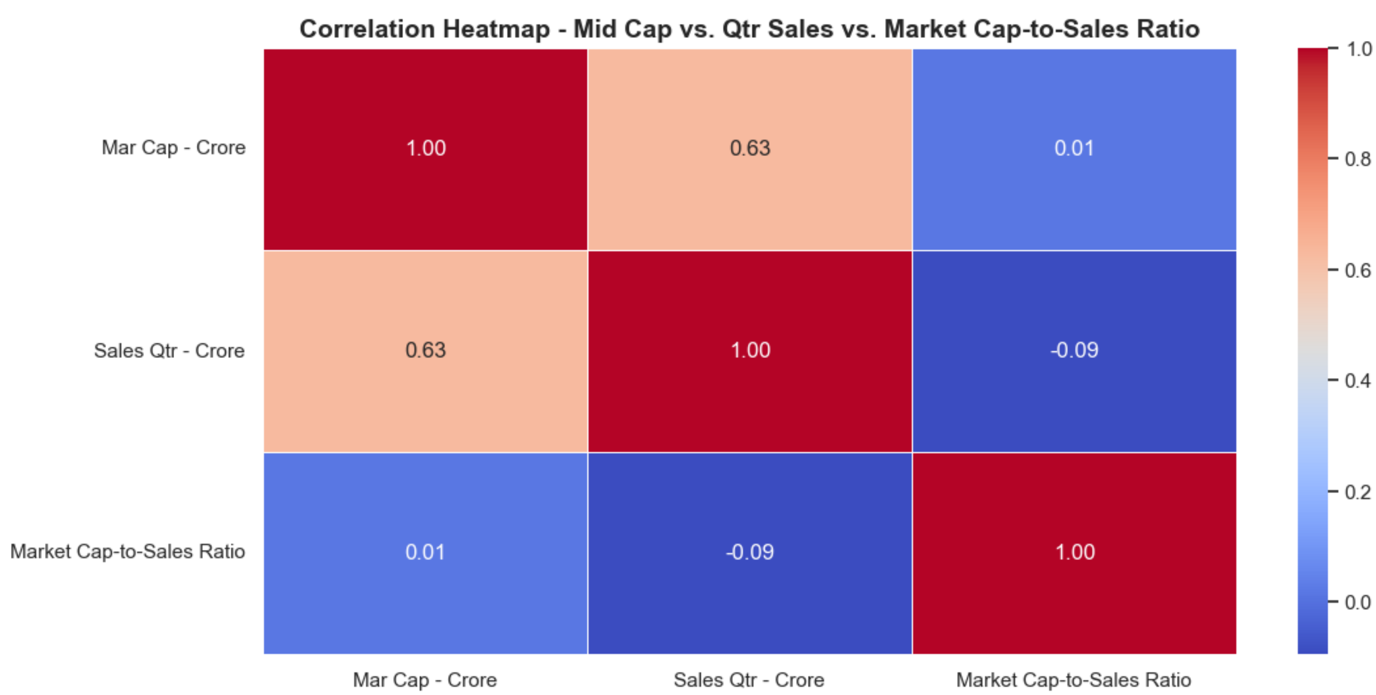


In [37]:

```
# Create a scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(fdf['Mar Cap - Crore'], fdf['Sales Qtr - Crore'], color='blue', alpha=0.7)
sns.regplot(x='Mar Cap - Crore', y='Sales Qtr - Crore', data=fdf, scatter_kws={'alpha':0

# Set plot title and labels
plt.title('Scatterplot with Trend Line - Market Cap vs. Qtr Sales')
plt.xlabel('Market Cap (in Crore)')
plt.ylabel('Quarterly Sales (in Crore)')
plt.grid(True)
plt.show()
```



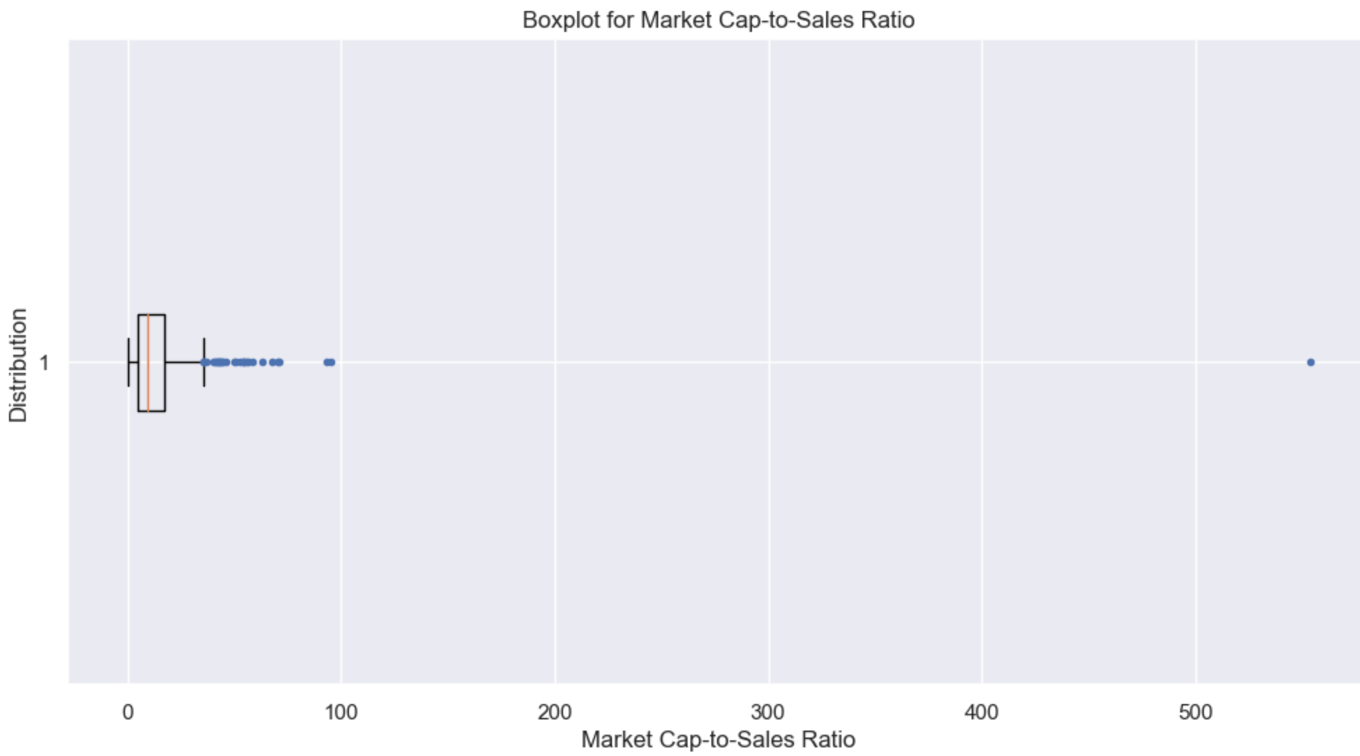
In [38]:
```
correlation_data = fdf[['Mar Cap - Crore', 'Sales Qtr - Crore','Market Cap-to-Sales Rati

correlation_matrix = correlation_data.corr()

plt.figure(figsize=(12, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)

plt.title('Correlation Heatmap - Mid Cap vs. Qtr Sales vs. Market Cap-to-Sales Ratio',fo
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```



Loading [MathJax]/extensions/Safe.js

Observation: The correlation coefficient of 0.63 indicates a positive relationship between Market Capitalization and Quarterly Sales. However, there is minimal or negative correlation between these parameters and the ratio.

In [39]:
```
# Create a boxplot for  Market Cap-to-Sales Ratio
column_name = 'Market Cap-to-Sales Ratio'
plt.figure(figsize=(12, 6))
plt.boxplot(fdf[column_name], vert=False, sym='b.')
plt.title(f'Boxplot for {column_name}')
plt.xlabel(column_name)
plt.ylabel('Distribution')
plt.show()
```



In [40]:
```
# Step 1: Calculate IQR
q1 = fdf['Market Cap-to-Sales Ratio'].quantile(0.25)
q3 = fdf['Market Cap-to-Sales Ratio'].quantile(0.75)
iqr = q3 - q1

# Step 2: Define lower and upper bounds
lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr

# Step 3: Remove outliers
fdf_no_outliers = fdf[(fdf['Market Cap-to-Sales Ratio'] >= lower_bound) & (fdf['Market C

# Display the updated DataFrame
print("DataFrame after removing outliers:")
print(fdf_no_outliers)
```

```
DataFrame after removing outliers:
     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   
..     ...               ...              ...                ...   
482    495       Prime Focus          3031.50             609.61   
483    496  Lak. Vilas Bank          3029.57             790.17   
484    497             NOCIL          3026.26             249.27   
485    498    Orient Cement           3024.32             511.53   
486    499  Natl.Fertilizer          3017.07            2840.75   

     Market Cap-to-Sales Ratio  
0                     5.845474  
1                    18.240676  
2                    23.465685  
3                    32.847382  
4                    17.190535  
..                         ...  
482                   4.972851  
483                   3.834074  
484                  12.140490  
485                   5.912302  
486                   1.062068  

[449 rows x 5 columns]
```

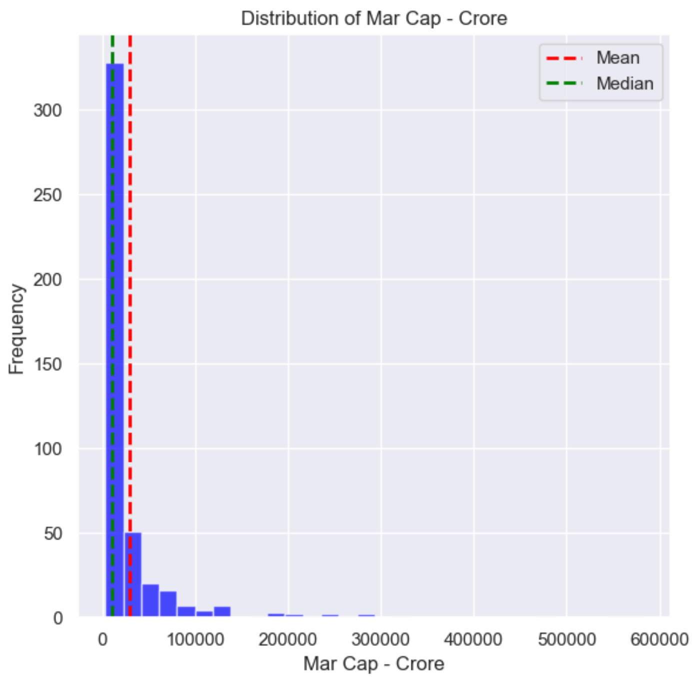
In [41]:
```python
fdf_no_outliers.describe()
```

Out[41]:

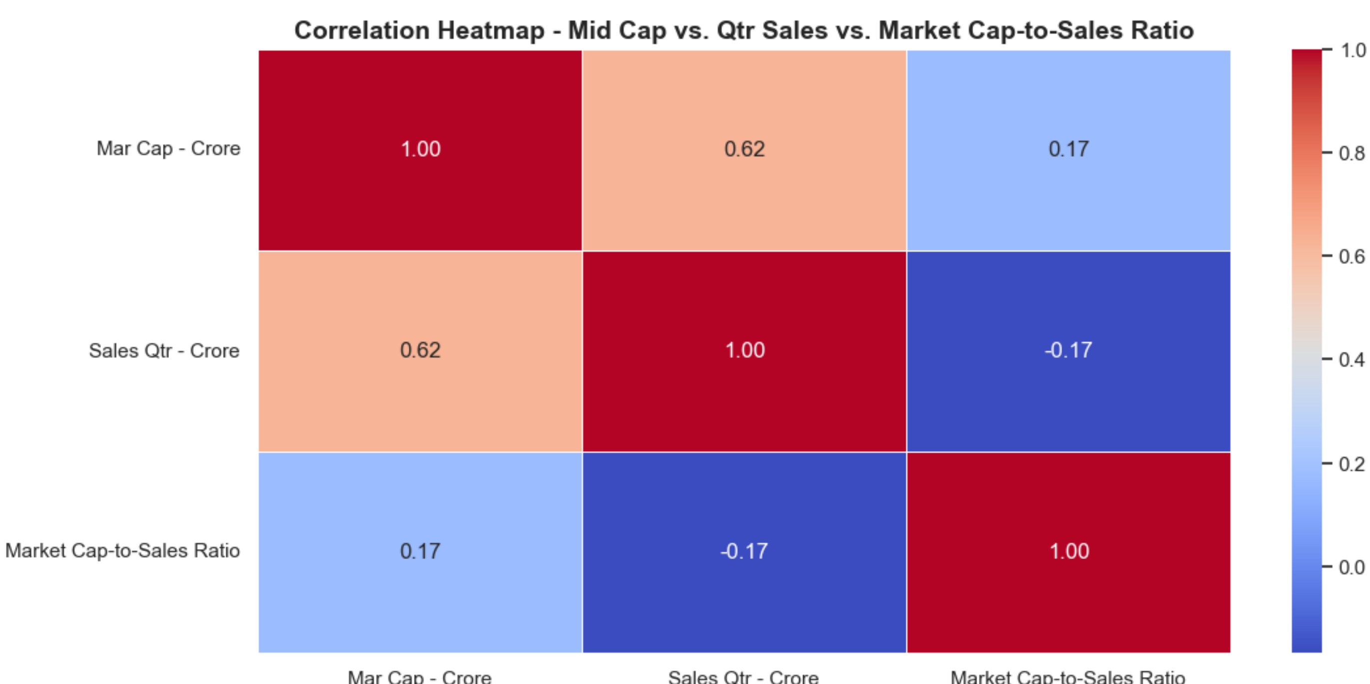
	S.No.	Mar Cap - Crore	Sales Qtr - Crore	Market Cap-to-Sales Ratio
count	449.000000	449.000000	449.000000	449.000000
mean	248.775056	29091.643697	3875.877016	11.000772
std	146.386202	61232.836412	10090.031729	8.318651
min	1.000000	3017.070000	0.000000	0.000000
25%	118.000000	4886.090000	553.840000	4.868833
50%	249.000000	9885.050000	1156.610000	8.902439
75%	374.000000	24626.100000	2780.260000	15.064476
max	499.000000	583436.720000	110666.930000	35.447508

In [42]:
```python
feature1 = 'Mar Cap - Crore'
feature2 = 'Sales Qtr - Crore'

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

plt.subplot(1, 2, 1)
plt.hist(fdf_no_outliers[feature1].dropna(), bins=30, color='blue', alpha=0.7)
plt.title(f'Distribution of {feature1}')
plt.xlabel(feature1)
plt.ylabel('Frequency')
plt.axvline(fdf_no_outliers[feature1].mean(), color='red', linestyle='dashed', linewidth
plt.axvline(fdf_no_outliers[feature1].median(), color='green', linestyle='dashed', linew
plt.legend()

plt.subplot(1, 2, 2)
plt.hist(fdf_no_outliers[feature2].dropna(), bins=30, color='orange', alpha=0.7)
plt.title(f'Distribution of {feature2}')
plt.xlabel(feature2)
```

```
plt.ylabel('Frequency')
plt.axvline(fdf_no_outliers[feature2].mean(), color='red', linestyle='dashed', linewidth
plt.axvline(fdf_no_outliers[feature2].median(), color='green', linestyle='dashed', linew
plt.legend()

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



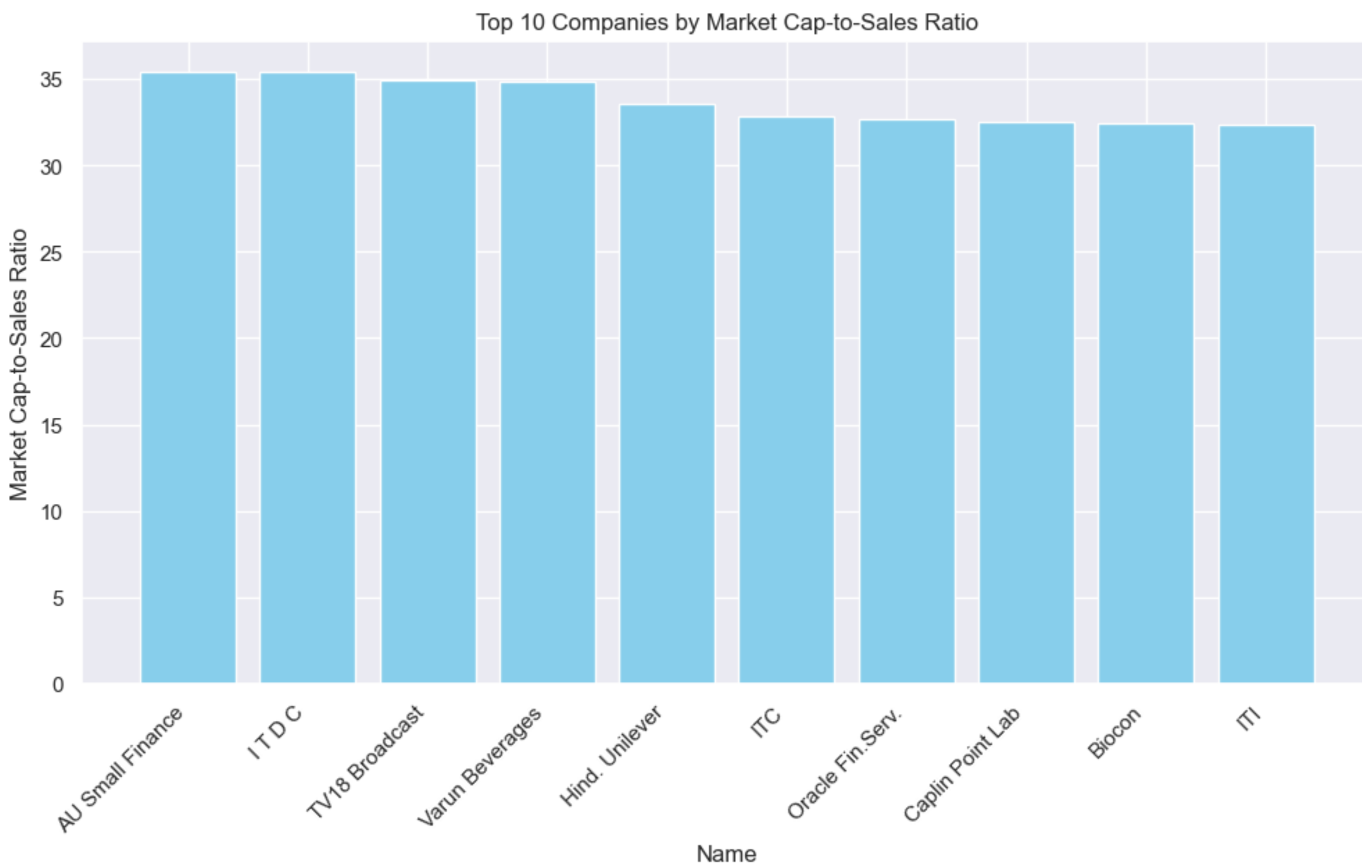
In [43]:
```
correlation_data = fdf_no_outliers[['Mar Cap - Crore', 'Sales Qtr - Crore','Market Cap-t

correlation_matrix = correlation_data.corr()

plt.figure(figsize=(12, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)

plt.title('Correlation Heatmap - Mid Cap vs. Qtr Sales vs. Market Cap-to-Sales Ratio',fo
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```



Loading [MathJax]/extensions/Safe.js

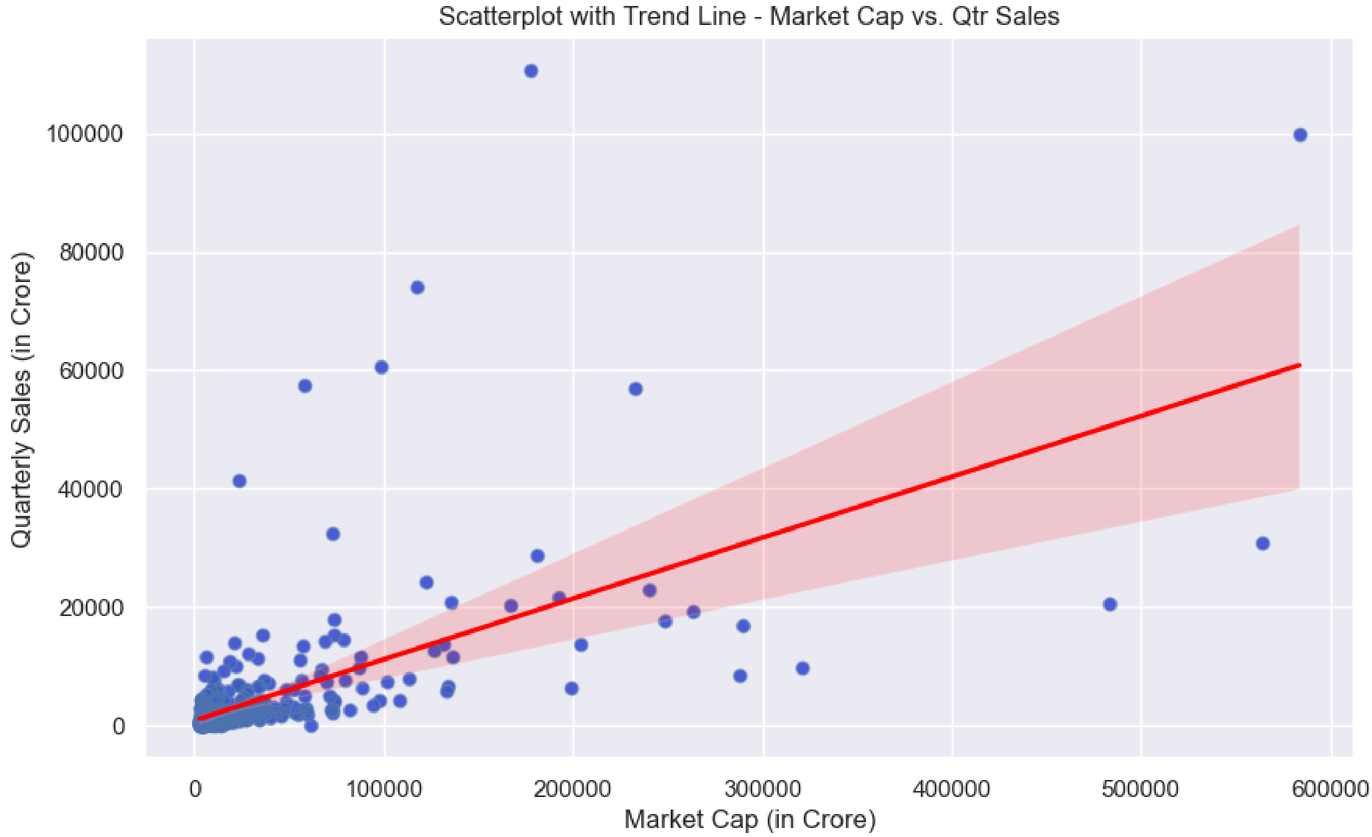
Observation: The correlation coefficient of 0.62 indicates a positive relationship between Market Capitalization and Quarterly Sales. However, there is minimal or negative correlation between these parameters and the ratio.

In [44]:
```
top_n = 10

top_companies = fdf_no_outliers.nlargest(top_n, 'Market Cap-to-Sales Ratio')

plt.figure(figsize=(12, 6))
plt.bar(top_companies['Name'], top_companies['Market Cap-to-Sales Ratio'], color='skyblu
plt.title(f'Top {top_n} Companies by Market Cap-to-Sales Ratio')
plt.xlabel('Name')
plt.ylabel('Market Cap-to-Sales Ratio')
plt.xticks(rotation=45, ha='right')
plt.show()
```

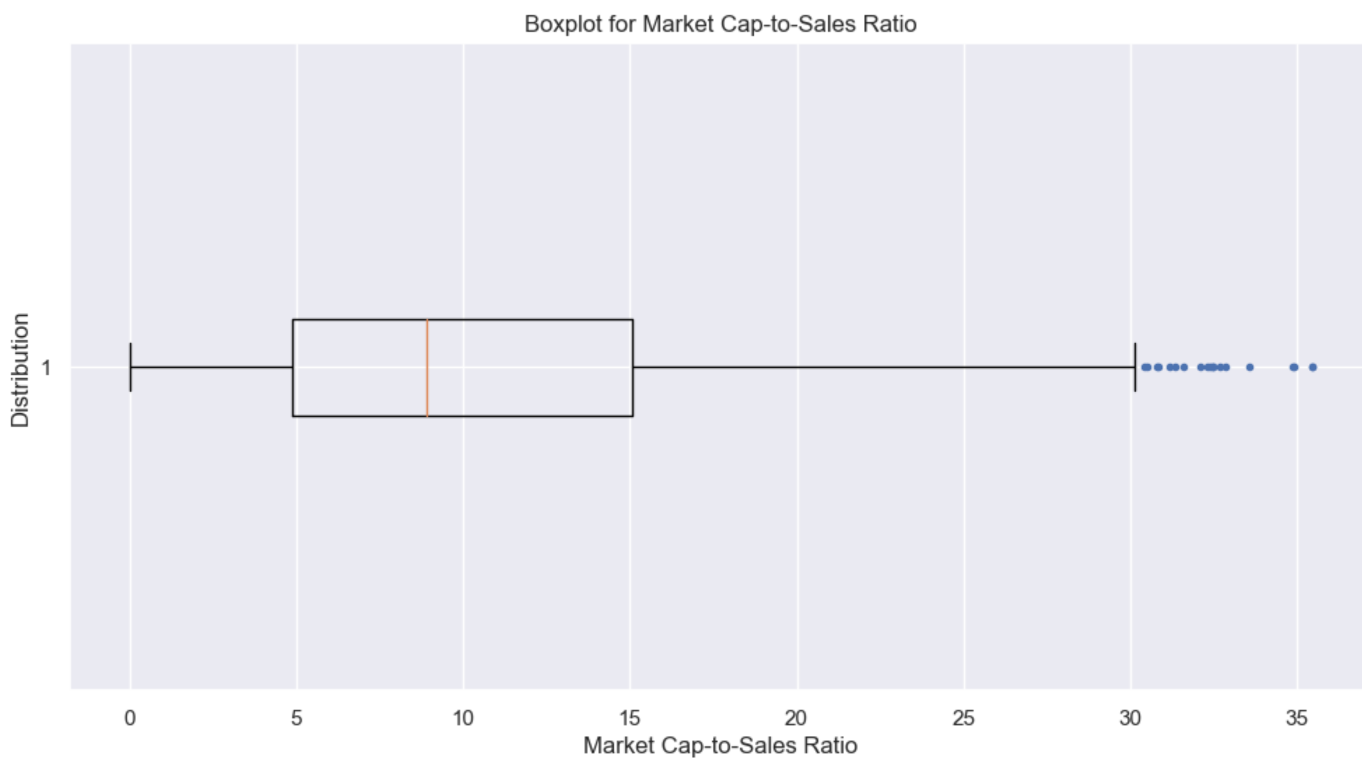


In [45]:
```
# Create a scatter plot with a trend line
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Mar Cap - Crore', y='Sales Qtr - Crore', data=fdf_no_outliers, color=
sns.regplot(x='Mar Cap - Crore', y='Sales Qtr - Crore', data=fdf_no_outliers, scatter_kw

# Set plot title and labels
plt.title('Scatterplot with Trend Line - Market Cap vs. Qtr Sales')
plt.xlabel('Market Cap (in Crore)')
plt.ylabel('Quarterly Sales (in Crore)')
plt.grid(True)
plt.show()
```

Loading [MathJax]/extensions/Safe.js



In [46]:
```python
# Create a boxplot for Mar Cap - Crore
column_name = 'Market Cap-to-Sales Ratio'
plt.figure(figsize=(12, 6))
plt.boxplot(fdf_no_outliers[column_name], vert=False, sym='b.')
plt.title(f'Boxplot for {column_name}')
plt.xlabel(column_name)
plt.ylabel('Distribution')
plt.show()
```



In [47]:
```python
fdf_no_outliers
```

Loading [MathJax]/extensions/Safe.js

Out[47]:

	S.No.	Name	Mar Cap - Crore	Sales Qtr - Crore	Market Cap-to-Sales Ratio
0	1	Reliance Inds.	583436.72	99810.00	5.845474
1	2	TCS	563709.84	30904.00	18.240676
2	3	HDFC Bank	482953.59	20581.27	23.465685
3	4	ITC	320985.27	9772.02	32.847382
4	5	H D F C	289497.37	16840.51	17.190535
...	...	...	...	...	...
482	495	Prime Focus	3031.50	609.61	4.972851
483	496	Lak. Vilas Bank	3029.57	790.17	3.834074
484	497	NOCIL	3026.26	249.27	12.140490
485	498	Orient Cement	3024.32	511.53	5.912302
486	499	Natl.Fertilizer	3017.07	2840.75	1.062068

449 rows × 5 columns

Exporting cleaned and preprocessed data

In [48]:

```
fdf_no_outliers.to_csv('D:\\Unified_Internship\\My_projectWork\\Project4(finance)\\Finan
```

In [ ]: