Financial Analytics

December 5, 2023

1 Financial Analyst

it is difficult for a business to survive. I am tasked to analyse the competition for the management to provide better results. This data set has information on the market capitalization of the top 500 companies in India.

[1]:
```
#importing Nesesarry Libaries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
df=pd.read_csv('Financial Analytics data.csv')
df
```

[2]:
```
     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              NaN                NaN

[488 rows x 4 columns]
```

[3]:
```
# rows and columns
df.shape
```

[3]:
```
(488, 4)
```

[4]:
```
# Basis Statistics
df.describe()
```

[4]:

```
             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
```

[5]:

```
df.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
```

[6]:

```
# rows and columns
df.shape
```

[6]:

```
(488, 4)
```

[7]:

```
#converting the Mar cap and sales   Qtr from float data type to numeric
df['Mar Cap - Crore'] = pd.to_numeric(df['Mar Cap - Crore'], errors='coerce')
df['Sales Qtr - Crore'] = pd.to_numeric(df['Sales Qtr - Crore'],␣
  ↪errors='coerce')
df.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
```

2 Exploratory Data Analysis

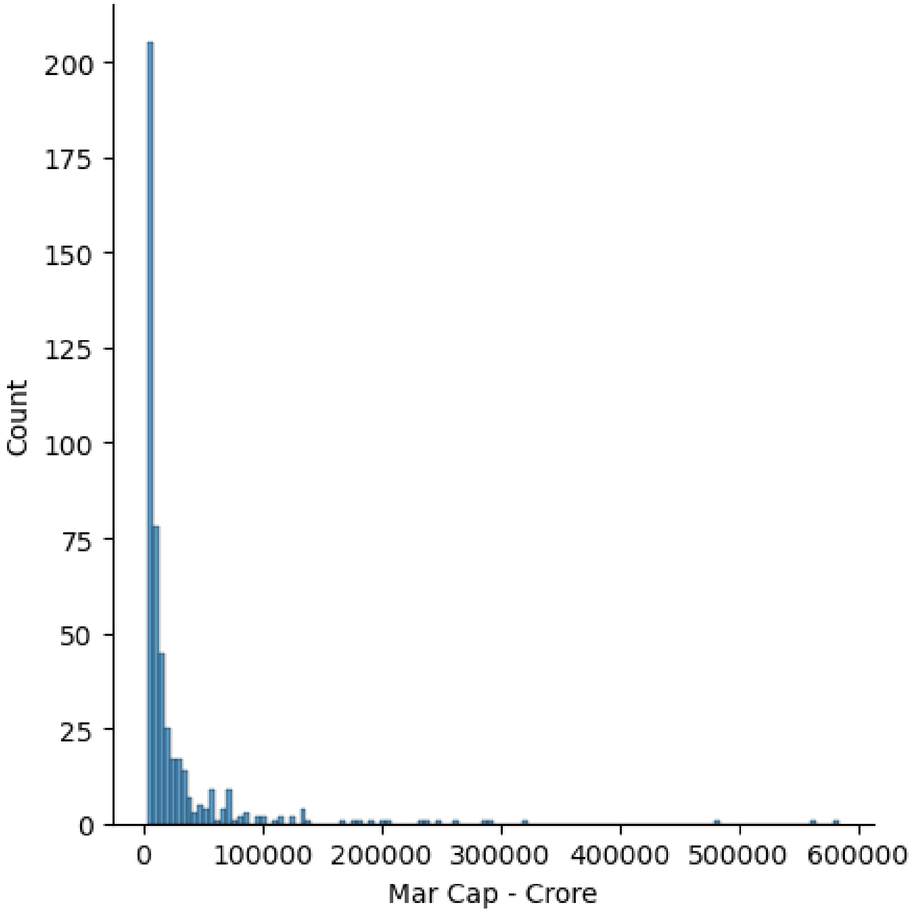
```
[8]: df.isnull().sum()#Checking for null values
```

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

So the Mar cap and sales column contain 9 and 29 null values so we will fill the nan by seeing the skewness of both the data in a graph to fill the values

```
[9]: #using Distribution plot to choose which method to choose the fill the null␣
      ↪values for sale scolumn
     sns.displot(df['Mar Cap - Crore'])
```

```
[9]: <seaborn.axisgrid.FacetGrid at 0x220f7eaebf0>
```



So the mar cap is skewed to the extreme left so we will use the median tecnique to fill the null data

[10]:

```
# Filling the null values using Median
col = df['Mar Cap - Crore']
col = col.fillna(col.median(),inplace=True)
```

[11]:

```
# After Fill na Method
df.isnull().sum()
```

[11]:

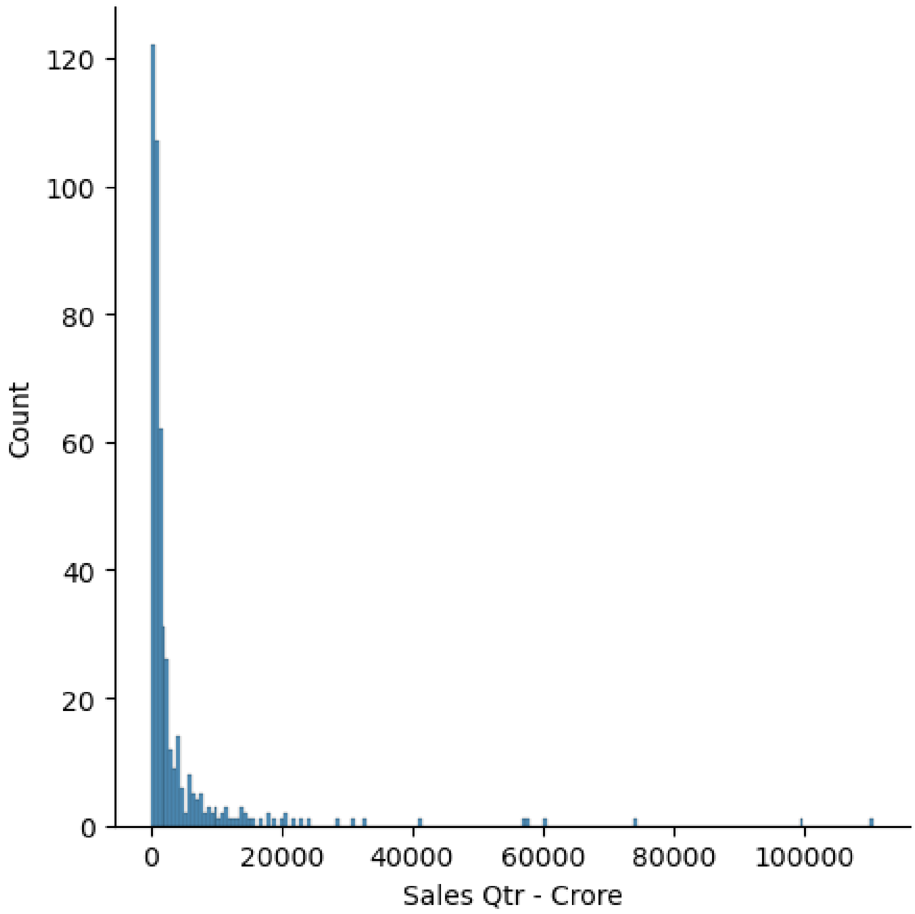
```
S.No.               0
Name                0
Mar Cap - Crore     0
Sales Qtr - Crore  29
dtype: int64
```

[12]:

```
#using Distribution plot to choose which method to choose the fill the null␣
 ↪values for sale scolumn
sns.displot(df['Sales Qtr - Crore'])
```

[12]:

```
<seaborn.axisgrid.FacetGrid at 0x220f7eaef50>
```



Since the skewness of the graph is extreme left so we use median technique to fill the null values

[13]:

```
# Filling the null values using Median
col = df['Sales Qtr - Crore']
col = col.fillna(col.median(),inplace=True)
```

[14]:

```
# After Fill na Method
df.isnull().sum()
```

[14]:

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

Null values have been filled as we can see in the data

[15]:

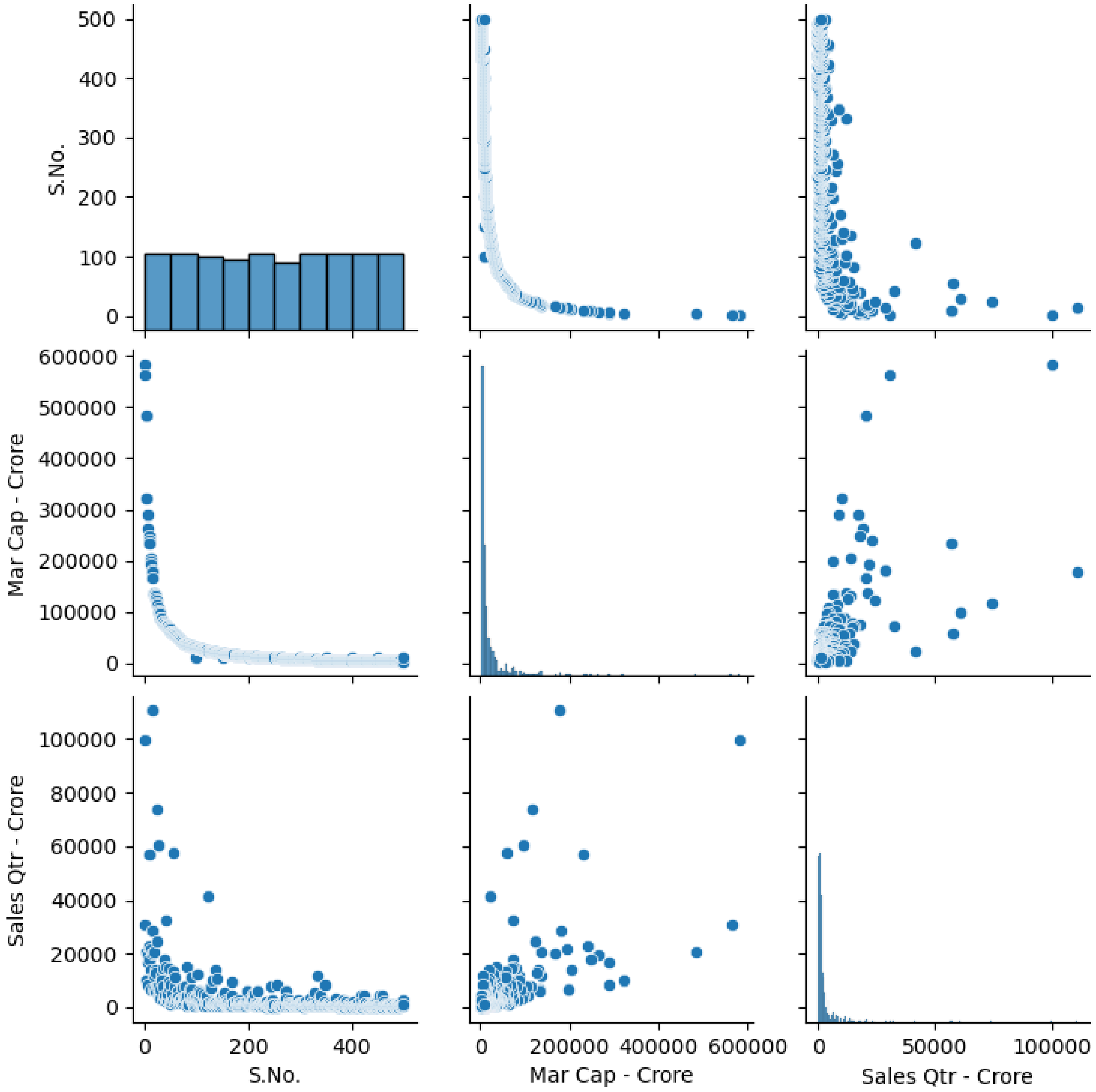
```
# Summary statistics
print(df.describe())
```

```
            S.No.  Mar Cap - Crore  Sales Qtr - Crore
count  488.000000       488.000000         488.000000
mean   251.508197     27708.961086        3649.084570
std    145.884078     58963.329098        9708.054143
min      1.000000      3017.070000           0.000000
25%    122.750000      4879.612500         570.035000
50%    252.500000      9885.050000        1137.170000
75%    378.250000     23400.815000        2580.797500
max    500.000000    583436.720000      110666.930000
```

3 Visualizations

[25]:

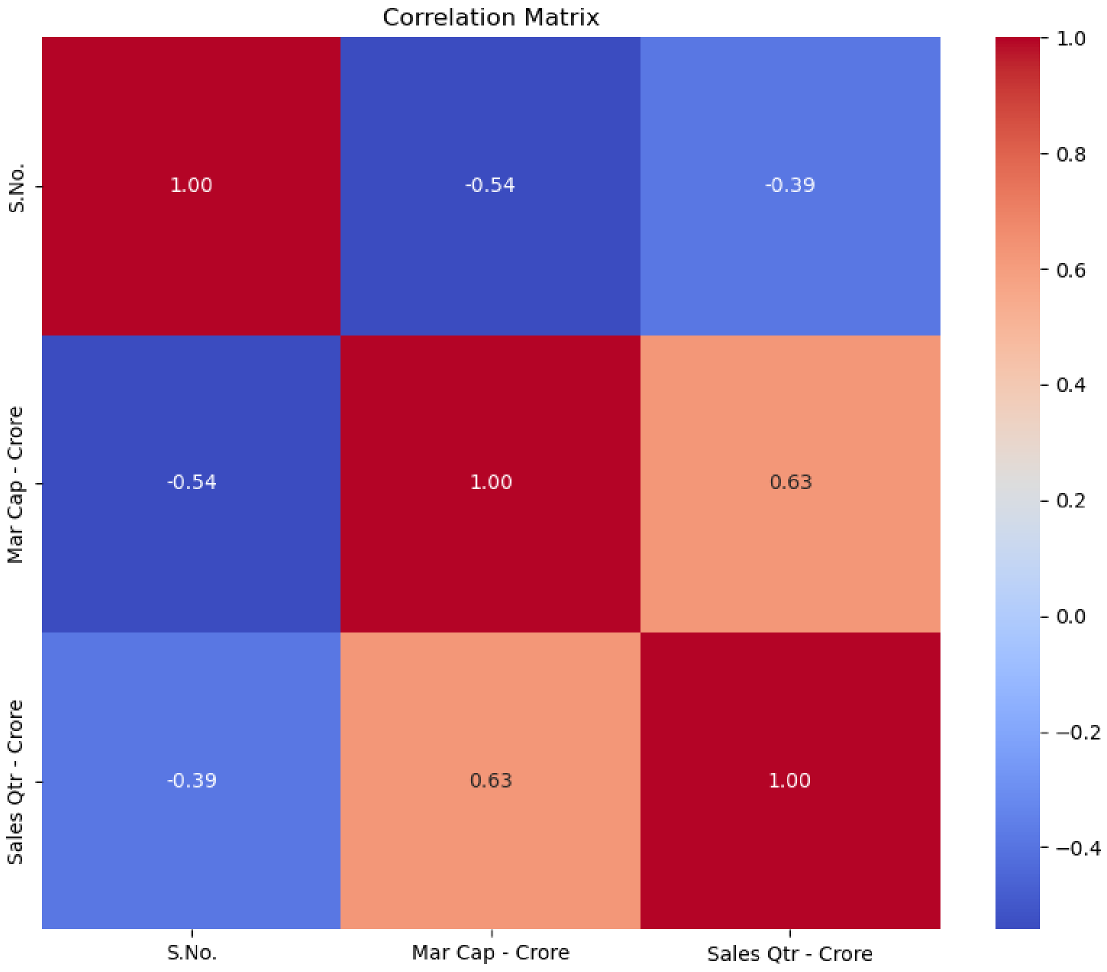
```
# Pairplot for numerical variables
sns.pairplot(df)
plt.show()
```



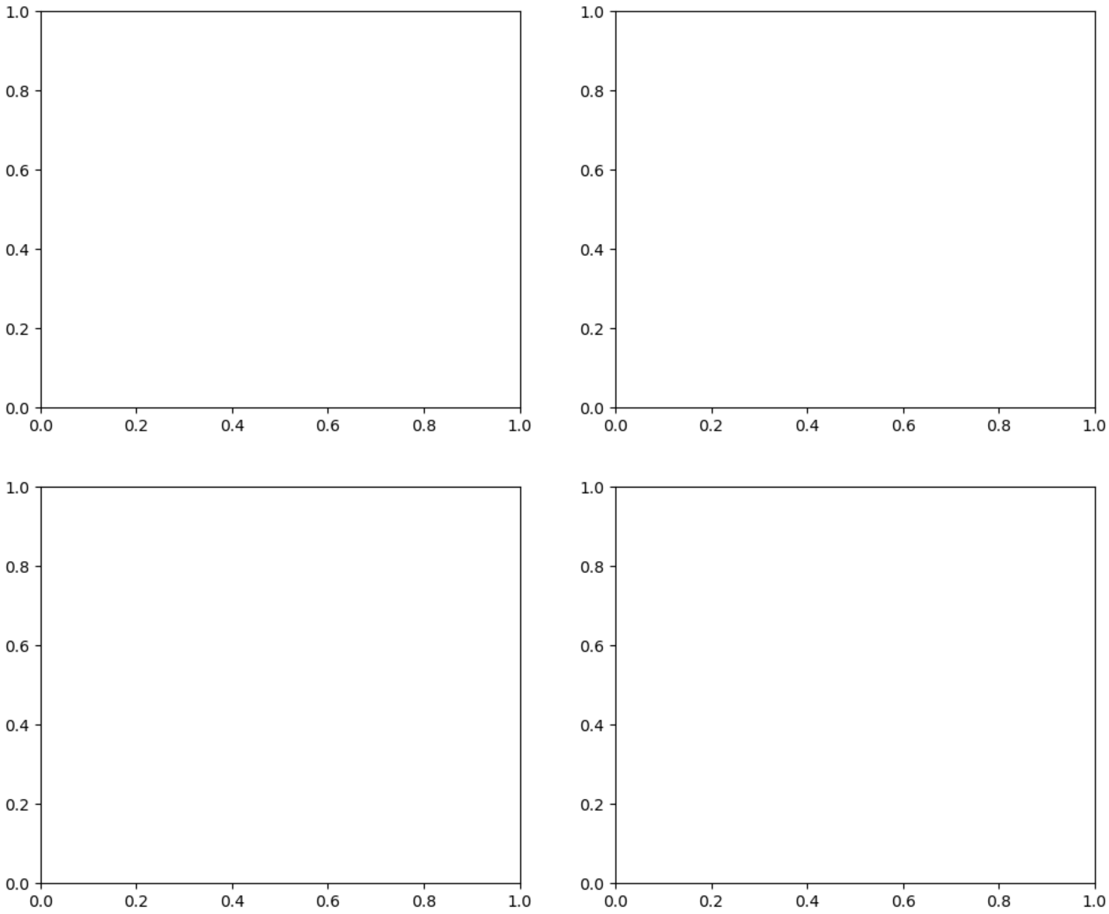
[26]:

```
# Correlation matrix heatmap
correlation_matrix = df.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
```

```
C:\Users\burag\AppData\Local\Temp\ipykernel_10788\3725550307.py:2:
FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only valid
columns or specify the value of numeric_only to silence this warning.
  correlation_matrix = df.corr()
```



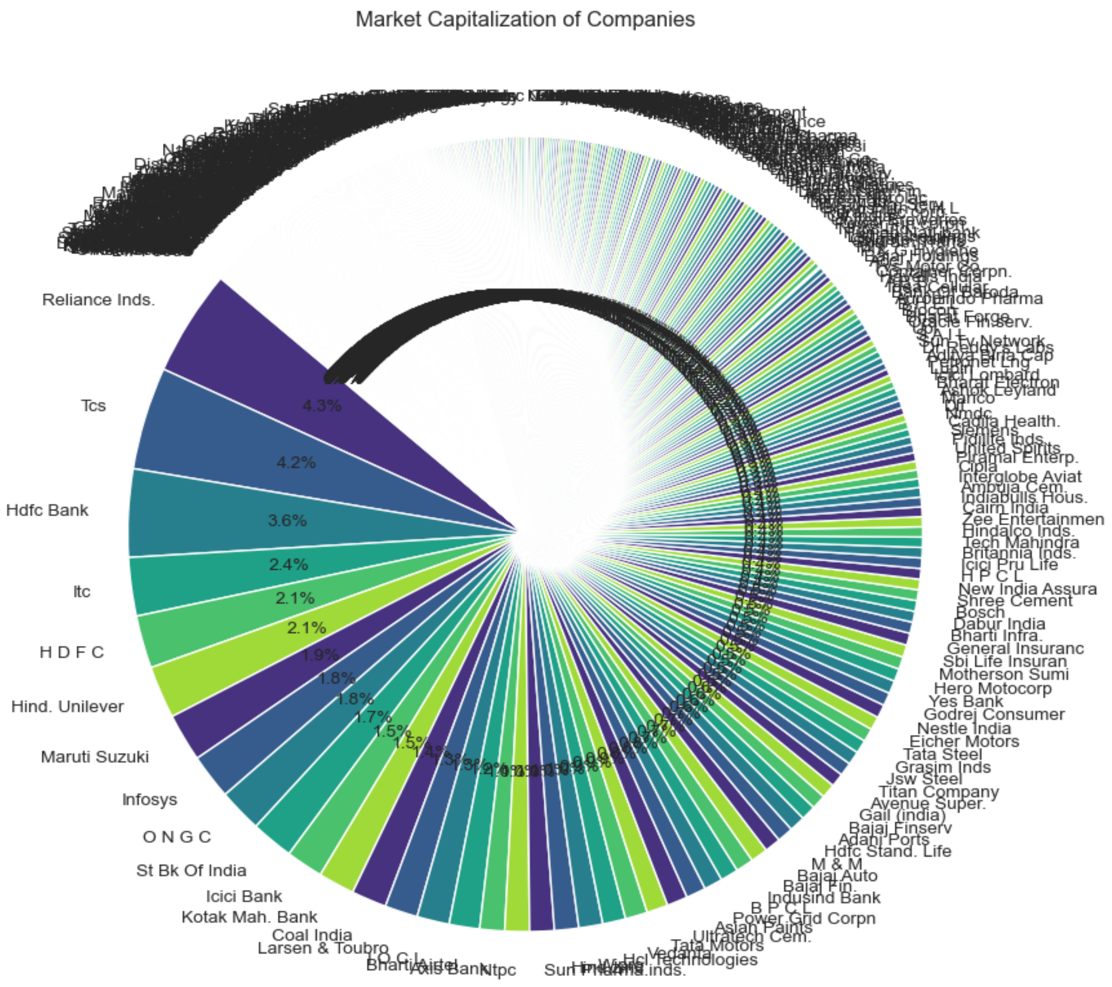
[27]:
```
# Distribution plots for key metrics
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12, 10))
```



[68]:
```
# Assuming df is your DataFrame
sns.set_theme()

# Plot a pie chart
plt.figure(figsize=(10, 10))
plt.pie(df['Mar Cap - Crore'], labels=df['Name'], autopct='%1.1f%%', startangle=140, colors=sns.color_palette('viridis'), textprops={'fontsize': 10})
plt.title('Market Capitalization of Companies')

# Show the pie chart
plt.show()
```



Market Capitalization of Companies

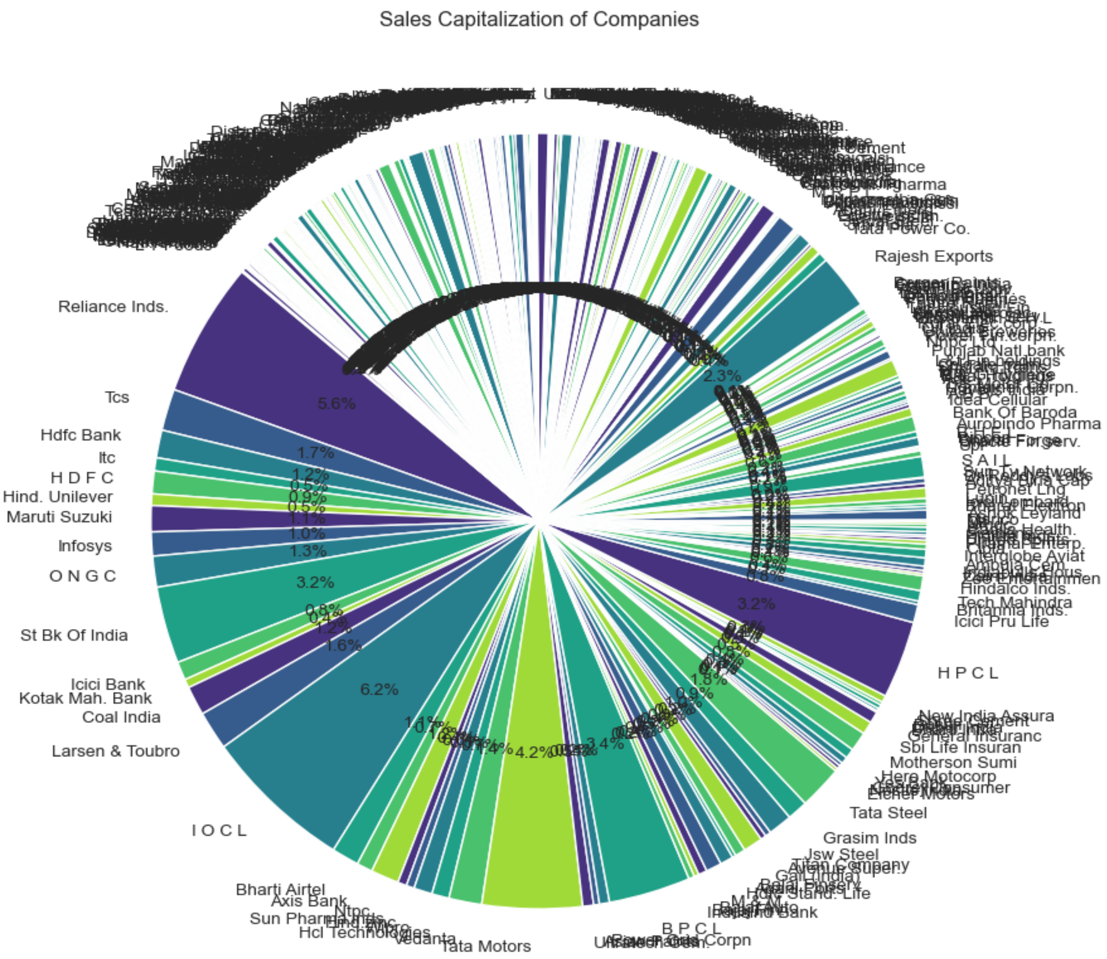
In this data we can analyze that the Relince industry have the higest Market Cap follow along TCS

[66]:

```
# Assuming df is your DataFrame
sns.set_theme()

# Plot a pie chart
plt.figure(figsize=(10, 10))
plt.pie(df['Sales Qtr - Crore'], labels=df['Name'], autopct='%1.1f%%', startangle=140, colors=sns.color_palette('viridis'), textprops={'fontsize': 10})
plt.title('Sales Capitalization of Companies')

# Show the pie chart
plt.show()
```



In this data we can Analyze that the Iocl has the higest Quaterly sales then the reliance Ind

[ ]:

[ ]:

[ ]:

[ ]: