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

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

In [3]:
```
data.head(20)
```

Out[3]:

	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
5	6	Hind. Unilever	288265.26	8590.00	NaN
6	7	Maruti Suzuki	263493.81	19283.20	NaN
7	8	Infosys	248320.35	17794.00	NaN
8	9	O N G C	239981.50	22995.88	NaN
9	10	St Bk of India	232763.33	57014.08	NaN
10	11	ICICI Bank	203802.35	13665.35	NaN
11	12	Kotak Mah. Bank	199253.77	6390.71	NaN
12	13	Coal India	192677.98	21643.28	NaN
13	14	Larsen & Toubro	180860.74	28747.45	NaN
14	15	I O C L	178017.48	110666.93	NaN
15	16	Bharti Airtel	167131.29	20318.60	NaN
16	17	Axis Bank	136380.76	11721.55	NaN
17	18	NTPC	135390.53	20774.37	NaN
18	19	Sun Pharma.Inds.	134241.36	6653.23	NaN
19	20	Hind.Zinc	133266.56	5922.00	NaN

Observation

Column 'Unnamed: 4' consists of majority of NAN values, dataset includes S.No, Name, Mar Cap - Crore, Sales Qtr - Crore and Unnamed: 4

In [4]:
```
data.iloc[50:100]
```

Out[4]:

	S.No.	Name	Mar Cap - Crore	Sales Qtr - Crore	Unnamed: 4
50	51	Dabur India	60015.00	1966.44	NaN
51	52	Bosch	59204.28	3071.92	NaN
52	53	Shree Cement	58987.08	2296.23	NaN
53	54	New India Assura	58108.48	5074.02	NaN

54	55	H P C L	58034.78	57474.25	NaN
55	56	ICICI Pru Life	57748.98	13555.32	NaN
56	57	Britannia Inds.	56837.20	2567.48	NaN
57	58	Tech Mahindra	56244.26	7775.96	NaN
58	59	Hindalco Inds.	55854.68	11022.81	NaN
59	60	Zee Entertainmen	54817.89	1838.07	NaN
60	61	Cairn India	53528.57	NaN	2149.36
61	62	Indiabulls Hous.	52781.67	NaN	3115.89
62	63	Ambuja Cem.	52361.46	NaN	6170.71
63	64	Interglobe Aviat	48621.37	NaN	6177.88
64	65	Cipla	48577.43	NaN	3913.82
65	66	Piramal Enterp.	47483.97	NaN	2858.36
66	67	United Spirits	46725.05	NaN	2263.30
67	68	Pidilite Inds.	45855.50	NaN	1542.90
68	69	Siemens	44239.04	NaN	2429.50
69	70	Cadila Health.	41876.19	NaN	3259.60
70	71	NMDC	41415.33	NaN	2469.03
71	72	DLF	40159.35	NaN	1693.72
72	73	Marico	39813.84	NaN	1337.59
73	74	Ashok Leyland	39047.57	NaN	7113.16
74	75	Bharat Electron	37776.23	NaN	2512.82
75	76	ICICI Lombard	37219.22	NaN	2110.99
76	77	Lupin	36878.85	NaN	3975.62
77	78	Petronet LNG	36615.00	NaN	7757.06
78	79	Aditya Birla Cap	36215.92	NaN	3325.02
79	80	Dr Reddy's Labs	35893.55	NaN	3834.10
80	81	Sun TV Network	35824.26	NaN	683.28
81	82	S A I L	35729.04	15323.65	NaN
82	83	UPL	35349.58	NaN	4194.00
83	84	Oracle Fin.Serv.	34620.19	NaN	1059.12
84	85	Bharat Forge	34397.69	NaN	1390.55
85	86	Biocon	34347.00	NaN	1057.90
86	87	B H E L	34162.38	NaN	6626.35
87	88	Aurobindo Pharma	33676.52	NaN	4336.11
88	89	Bank of Baroda	33364.23	11303.24	NaN
89	90	Idea Cellular	33047.33	NaN	6509.60
90	91	A B B	31983.33	NaN	2779.40
91	92	Havells India	31798.18	NaN	1965.77
92	93	Container Corpn.	31450.56	NaN	1639.55
93	94	TVS Motor Co.	30919.51	NaN	3684.95

94	95	ACC	30803.68	NaN	3494.24
95	96	Bajaj Holdings	30305.94	NaN	317.85
96	97	P & G Hygiene	30202.12	NaN	704.16
97	98	MRF	30030.01	NaN	3798.82
98	99	Shriram Trans.	29327.64	NaN	3087.67
99	100	Colgate-Palm.	NaN	NaN	NaN

Observation

There seems to be a error in dataset since values of the Sales column have been shifted to unnamed: 4 column

In [5]:
```
data.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
```

In [6]:
```
data.shape
```

Out[6]:
```
(488, 5)
```

Observation

There are in total 488 records and 5 columns

In [7]:
```
data.describe()
```

Out[7]:

	S.No.	Mar Cap - Crore	Sales Qtr - Crore	Unnamed: 4
count	488.000000	479.000000	365.000000	94.000000
mean	251.508197	28043.857119	4395.976849	1523.870106
std	145.884078	59464.615831	11092.206185	1800.008836
min	1.000000	3017.070000	47.240000	0.000000
25%	122.750000	4843.575000	593.740000	407.167500
50%	252.500000	9885.050000	1278.300000	702.325000
75%	378.250000	23549.900000	2840.750000	2234.815000
max	500.000000	583436.720000	110666.930000	7757.060000

Following are the NAN values in the dataset

In [8]:
```
data.isnull().sum()
```

Out[8]:
```
S.No.                  0
Name                   0
Mar Cap - Crore        9
Sales Qtr - Crore    123
Unnamed: 4           394
dtype: int64
```

Since there is an error in the dataset, many of the values in the Sales Qtr have been moved in another column 'unnamed 4'

Applying a function to make the corrections

In [9]:
```
data['Sales Qtr - Crore'].fillna(data['Unnamed: 4'], inplace=True)
```

In [10]:
```
data.isnull().sum()
```

Out[10]:
```
S.No.                  0
Name                   0
Mar Cap - Crore        9
Sales Qtr - Crore     29
Unnamed: 4           394
dtype: int64
```

Observation

The NAN values in the 'Sales Qtr - Crore' have been decreased

Dropping the 'Unnmaed: 4' since it is not required anymore

In [11]:
```
data.drop('Unnamed: 4', axis=1, inplace=True)
```

Applying a function to impute values in 'Mar Cap - Crore' Using the S.No.

In [12]:
```
def impute_mean_prev_next(column):
    if pd.isnull(column.iloc[0]):
        column.iloc[0] = column.iloc[1]

    if pd.isnull(column.iloc[-1]):
        column.iloc[-1] = column.iloc[-2]

    for i in range(1, len(column) - 1):
        if pd.isnull(column.iloc[i]):
            prev_value = column.iloc[i - 1]
            next_value = column.iloc[i + 1]
            if not pd.isnull(prev_value) and not pd.isnull(next_value):
                column.iloc[i] = (prev_value + next_value) / 2
            elif pd.isnull(prev_value) and not pd.isnull(next_value):
                column.iloc[i] = next_value
            elif not pd.isnull(prev_value) and pd.isnull(next_value):
                column.iloc[i] = prev_value
    return column
```

The above function can work in scenerios where there are not multiple NAN values at the same location in the column

In [13]:
```
# Applying the imputation function to the 'Mar Cap - Crore' column
data['Mar Cap - Crore'] = impute_mean_prev_next(data['Mar Cap - Crore'])
```

```
/tmp/ipykernel_358378/3774019202.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  column.iloc[-1] = column.iloc[-2]
/tmp/ipykernel_358378/3774019202.py:13: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  column.iloc[i] = (prev_value + next_value) / 2
```

In [14]:
```
data.iloc[99]
```

Out[14]:
```
S.No.                              100
Name                     Colgate-Palm.
Mar Cap - Crore              29130.035
Sales Qtr - Crore                  NaN
Name: 99, dtype: object
```

In [15]:
```
data.isnull().sum()
```

Out[15]:
```
S.No.                 0
Name                  0
Mar Cap - Crore       0
Sales Qtr - Crore    29
dtype: int64
```

Using the 'Mar Cap - Crore' to impute values in NAN records of 'Sales Qtr - Crore' as percentage of 'Mar Cap - Crore'

In [16]:
```
mean_percentage = ((data['Sales Qtr - Crore'] / data['Mar Cap - Crore']) * 100).mean()
mean_percentage
```

Out[16]:
```
16.496049620839532
```

Using the mean percentage of the values 'Sales Qtr - Crore' with respect to 'Mar Cap - Crore'

In [17]:
```
data['Sales Qtr - Crore'].fillna((data['Mar Cap - Crore']/100)*mean_percentage, inplace=
```

In [18]:
```
data.iloc[172]
```

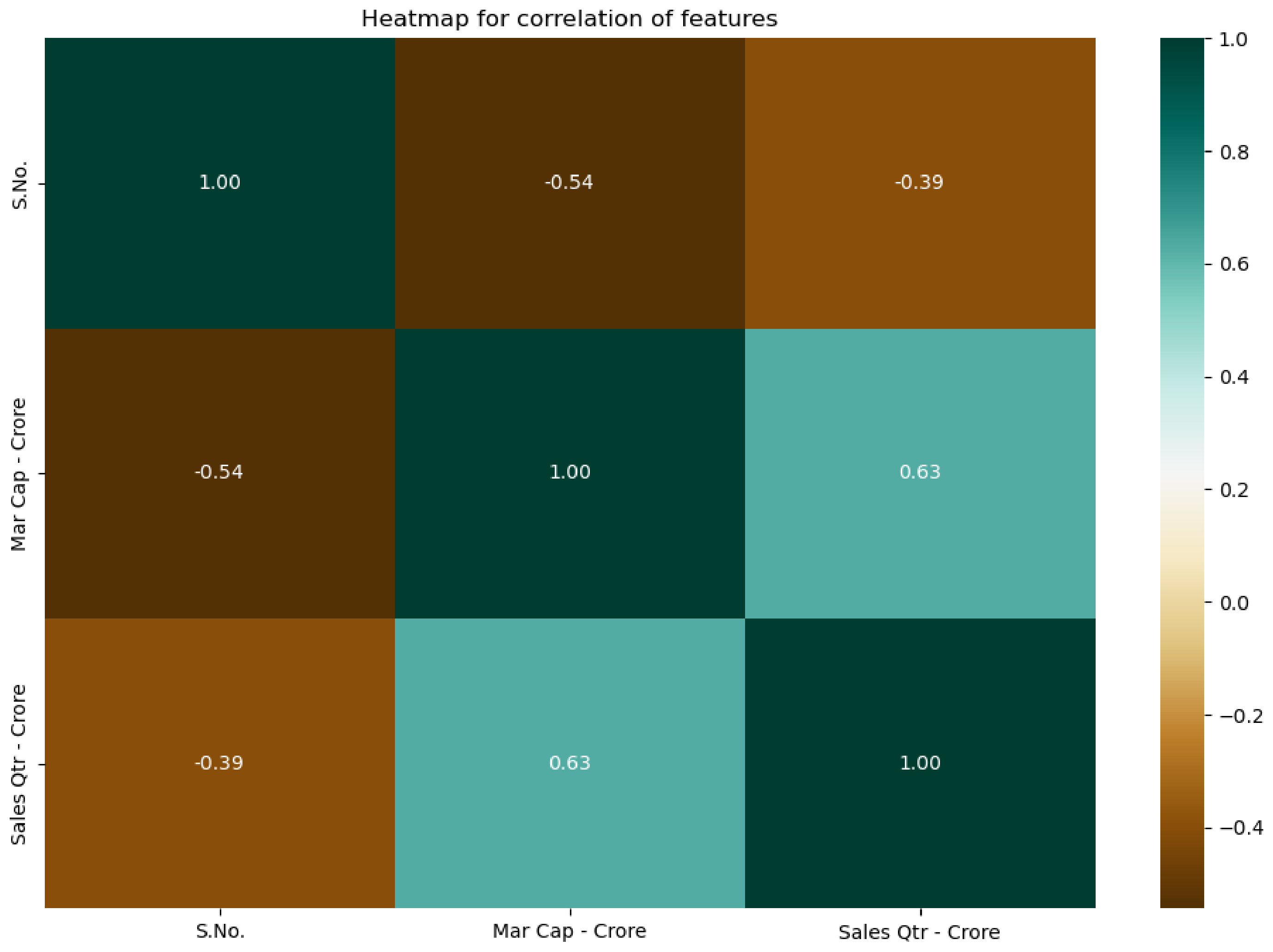
Out[18]:
```
S.No.                                177
Name                     Glenmark Pharma.
Mar Cap - Crore                 14785.53
Sales Qtr - Crore                2203.67
Name: 172, dtype: object
```

In [19]:
```python
data.isnull().sum()
```
Out[19]:
```
S.No.                0
Name                 0
Mar Cap - Crore      0
Sales Qtr - Crore    0
dtype: int64
```

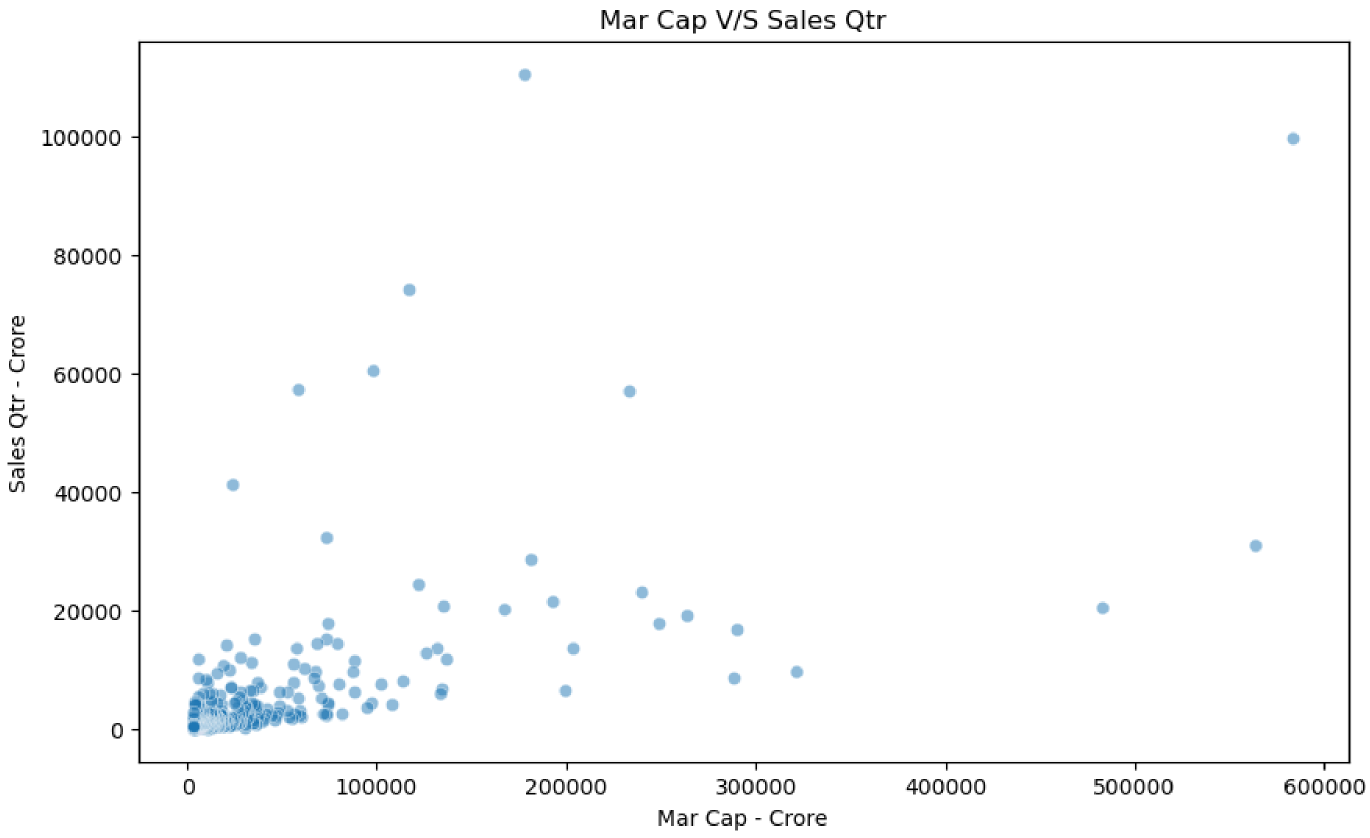
All the NAN values have been filled

In [20]:
```python
plt.figure(figsize=(12,8))
plt.title('Heatmap for correlation of features')

sns.heatmap(data = data.corr(), cmap='BrBG', annot=True, fmt='.2f')
plt.show()
```



In [22]:
```python
plt.figure(figsize=(10,6))
plt.title('Mar Cap V/S Sales Qtr')
sns.scatterplot(data = data, x = 'Mar Cap - Crore', y = 'Sales Qtr - Crore', alpha=0.5)
plt.show()
```



Observation

In the above plot we can see that majority of the comapanies have Mar Cap within 100000 Crore and Sales Qtr within 20000 Crore

Also we can see that Sales Qtr increses slightly as the Market Cap increases in huge amount

Conclusion

While the sales for the quarter increased slightly, the rise in market capitalization was disproportionately large.