Other Insights

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

file_path = 'C:\\Users\\DELL\\Desktop\\Data Analytics\\DataSets\\Cleaned - Assignment.xlsx'
Data = pd.read_excel(file_path)
print(Data)
```

```
    Company Name                        Model  Battery Power (Mah)  \
0        Micromax  BOLT S301 3G Without Charger                 1200
1         Karbonn                      A52 plus                 1300
2         Karbonn                      A52 plus                 1300
3         Karbonn                   A40 Indian                  1400
4            Lava                 Iris Atom X                  1400
..            ...                         ...                   ...
979       Samsung                  Galaxy M21                  6000
980       Samsung                  Galaxy M31                  6000
981       Samsung                  Galaxy M21                  6000
982          Asus                 ROG Phone II                 6000
983          Asus                  ROG Phone 3                 6000

       Battery Type Battery Performance  Rear Camera Type Front Camera Type  \
0            Normal                Low     Single Camera     Not Available
1         lithium-ion              Low     Single Camera     Not Available
2         lithium-ion              Low     Single Camera     Not Available
3            Normal                Low     Single Camera     Single Camera
4         lithium-ion              Low     Single Camera     Single Camera
..            ...                 ...               ...               ...
979          Normal      Extremely High  Multiple Cameras     Single Camera
980          Normal      Extremely High  Multiple Cameras     Single Camera
981          Normal      Extremely High  Multiple Cameras     Single Camera
982          Normal      Extremely High  Multiple Cameras     Single Camera
983  lithium-polymer     Extremely High  Multiple Cameras     Single Camera

     Display in Cms. Display Type  RAM (GB)  ROM (GB)  Expandable Upto (GB)  \
0               8.89        FWVGA      0.50       4.0                   0.0
1               8.89         HVGA      0.50       4.0                  32.0
2               8.89         HVGA      0.50       4.0                  32.0
3              10.16       NORMAL      1.00       8.0                  32.0
4              10.16         WVGA      0.25       0.5                  32.0
..               ...          ...       ...       ...                   ...
979            16.26      FULL HD+      6.00     128.0                 512.0
```

[7]:
```
import pandas as pd

company_summary = Data.groupby('Company Name').agg({
    'Price': 'mean',
    'Ratings': 'mean',
    'Battery Power (Mah)': 'mean',
    'RAM (GB)': 'mean',
    'ROM (GB)': 'mean',
    'Expandable Upto (GB)': 'mean'
}).reset_index()

print(company_summary)
```

```
   Company Name         Price   Ratings  Battery Power (Mah)  RAM (GB)  \
0        Alcatel   7019.400000  3.760000          3000.000000  2.200000
1           Asus  12973.550000  4.255000          4453.500000  4.150000
2          Black  31999.000000  4.400000          4000.000000  6.000000
3      Blackbear   4165.666667  3.033333          2433.333333  1.333333
4         Celkon   6642.000000  3.400000          2750.000000  2.500000
5          Comio   4027.000000  3.850000          2350.000000  1.250000
6        Coolpad   5815.800000  3.660000          2800.000000  2.400000
7          Forme   4999.000000  3.600000          2500.000000  3.000000
8         Gionee   8658.230769  3.992308          3456.538462  3.038462
9           Gome   6374.714286  4.071429          3242.857143  3.285714
10        Google  32999.000000  4.500000          3350.000000  4.000000
11        Homtom   5049.000000  3.600000          3500.000000  3.000000
12         Honor  15633.433333  4.310000          3459.000000  4.366667
13        Huawei  18449.000000  4.150000          3670.000000  5.000000
14       Infinix   9491.000000  4.288000          4368.000000  3.480000
15       Infocus   6194.500000  3.800000          2750.000000  2.500000
16         Intex   4332.333333  3.700000          2933.333333  1.333333
17          Iqoo  38790.000000  4.480000          4440.000000  8.800000
18          Itel   4243.263158  4.110526          2629.473684  1.315789
19       Karbonn   3350.461538  3.684615          2065.384615  0.966346
20      Kenxinda   3449.000000  3.500000          4500.000000  1.000000
21          Lava   4822.818182  3.839394          2603.030303  1.409091
22        Lenovo  10029.066667  4.006667          3494.666667  2.933333
23            Lg  19052.384615  4.038462          3303.846154  3.538462
24           Lyf   4565.666667  3.566667          2406.666667  2.000000
25         Meizu   7994.500000  3.950000          3035.000000  3.000000
26            Mi  17005.333333  4.208333          3667.500000  5.333333
27           Mi3  13999.000000  4.400000          3050.000000  2.000000
```

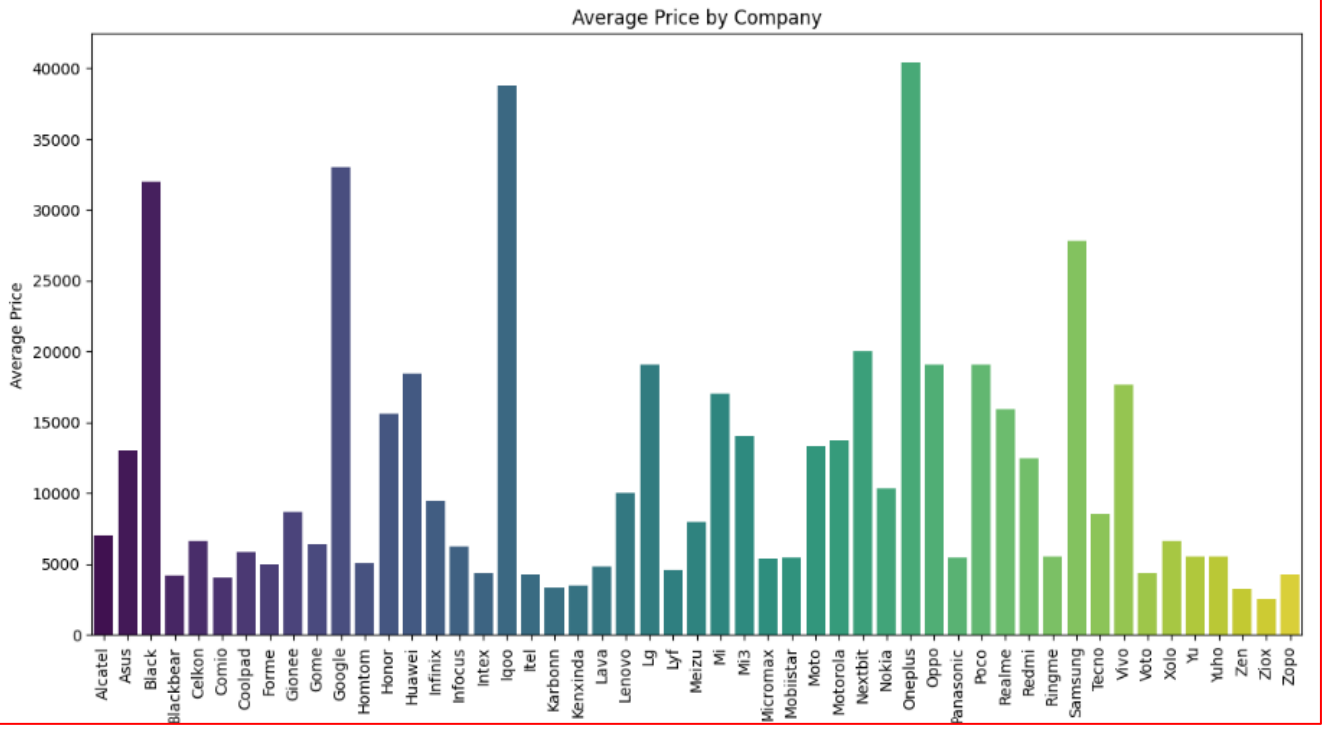
[8]:
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(14, 7))
sns.barplot(data=company_summary, x='Company Name', y='Price', palette='viridis')
plt.title('Average Price by Company')
plt.xlabel('Company Name')
plt.ylabel('Average Price')
plt.xticks(rotation=90)
plt.show()
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_25372\3089230283.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(data=company_summary, x='Company Name', y='Price', palette='viridis')
```

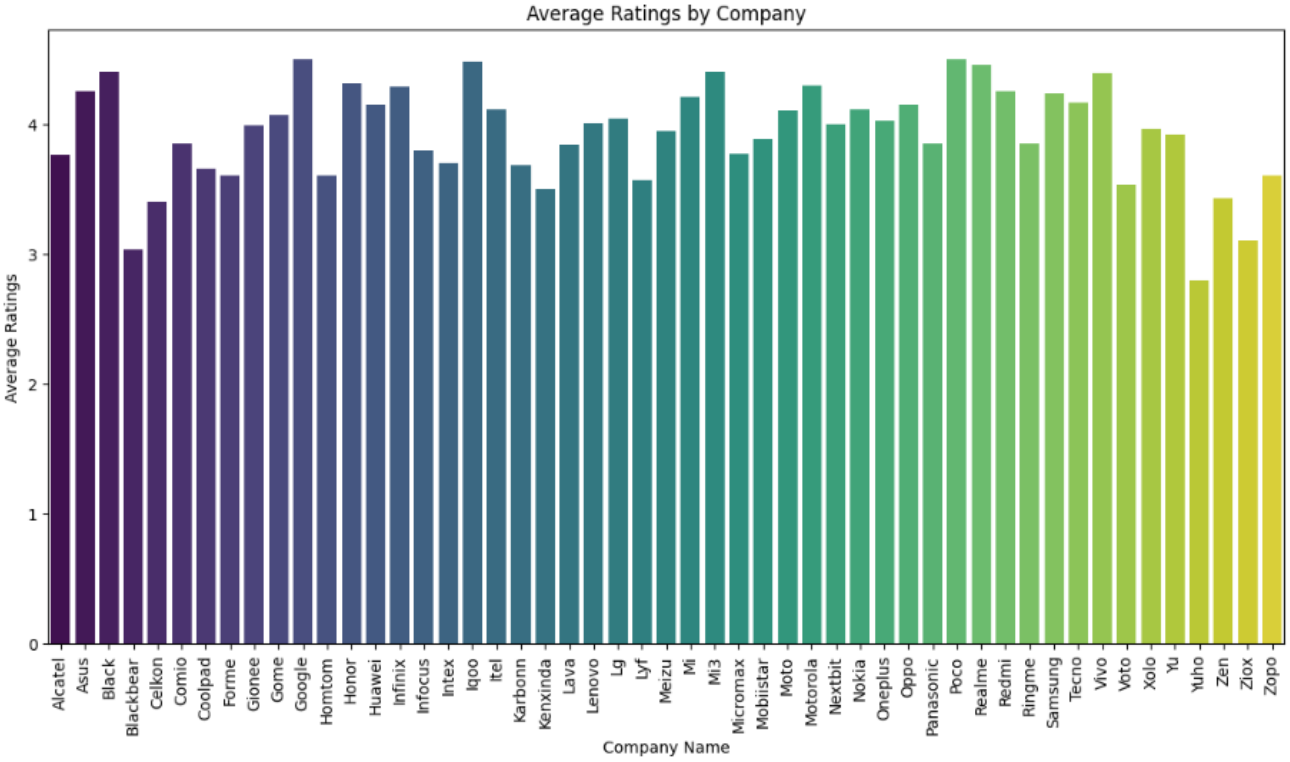


[10]:
```
plt.figure(figsize=(14, 7))
sns.barplot(data=company_summary, x='Company Name', y='Ratings', palette='viridis')
plt.title('Average Ratings by Company')
plt.xlabel('Company Name')
plt.ylabel('Average Ratings')
plt.xticks(rotation=90)
plt.show()
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_25372\2825548871.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(data=company_summary, x='Company Name', y='Ratings', palette='viridis')
```

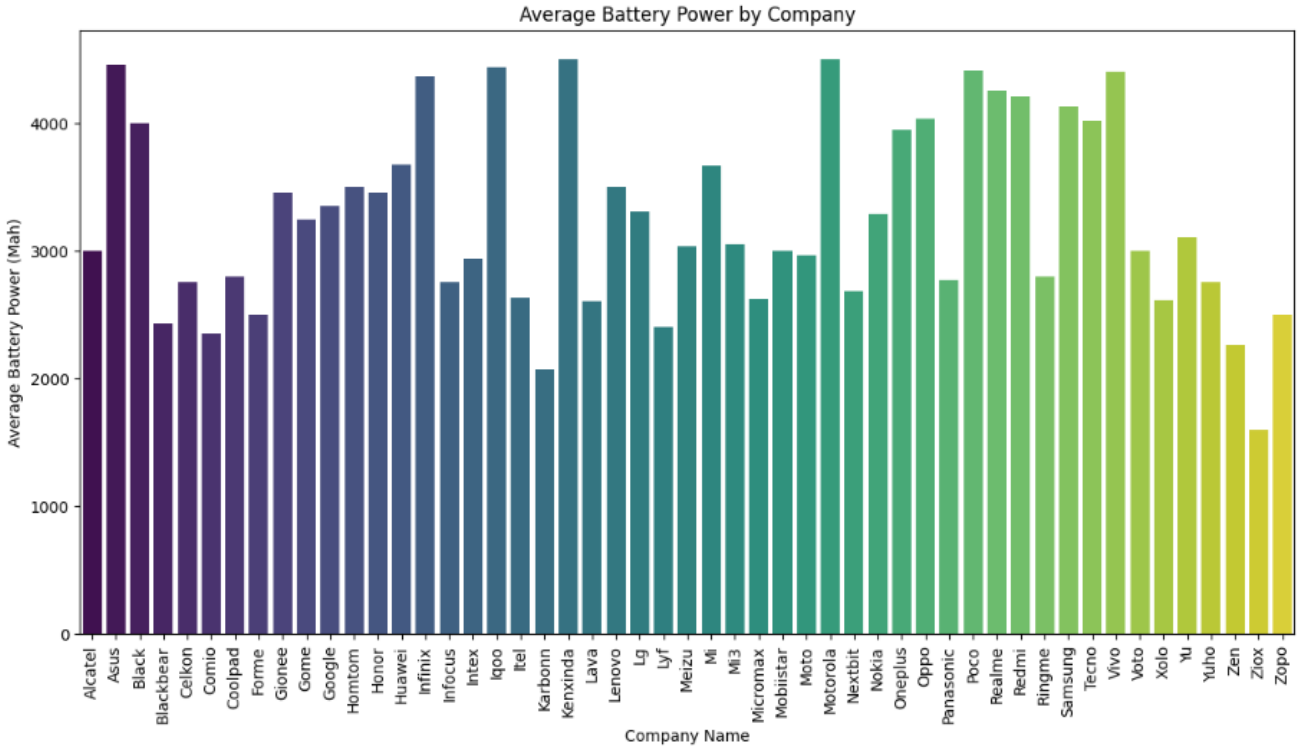


[11]:
```
plt.figure(figsize=(14, 7))
sns.barplot(data=company_summary, x='Company Name', y='Battery Power (Mah)', palette='viridis')
plt.title('Average Battery Power by Company')
plt.xlabel('Company Name')
plt.ylabel('Average Battery Power (Mah)')
plt.xticks(rotation=90)
plt.show()
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_25372\1005315467.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(data=company_summary, x='Company Name', y='Battery Power (Mah)', palette='viridis')
```

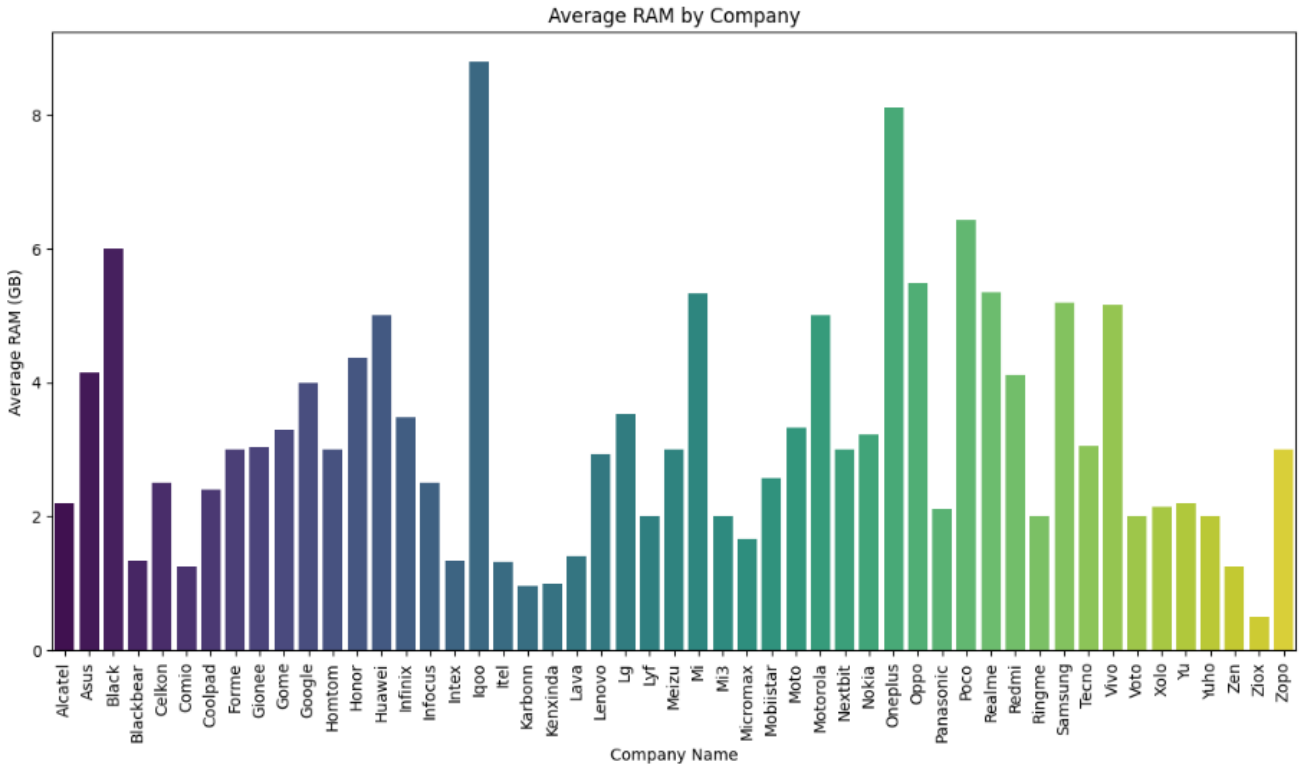


[12]:
```
plt.figure(figsize=(14, 7))
sns.barplot(data=company_summary, x='Company Name', y='RAM (GB)', palette='viridis')
plt.title('Average RAM by Company')
plt.xlabel('Company Name')
plt.ylabel('Average RAM (GB)')
plt.xticks(rotation=90)
plt.show()
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_25372\2331992576.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(data=company_summary, x='Company Name', y='RAM (GB)', palette='viridis')
```



[14]:
```
import pandas as pd
company_summary_melted = pd.melt(company_summary, id_vars=['Company Name'], var_name='Feature', value_name='Average Value')
print(company_summary_melted.head())
```

```
  Company Name Feature  Average Value
0      Alcatel   Price    7019.400000
1         Asus   Price   12973.550000
2        Black   Price   31999.000000
3   Blackbear   Price    4165.666667
4       Celkon   Price    6642.000000
```

[ ]: