Exploratory Data Analysis

Overview of the dataset:

- In this project, 50 pages from Flipkart – smart phones data has been scrapped
- Population dataset contains 996 rows/observations and 16 columns/fields

Field Name	Data Type	Blanks	Unique Values
Product_Name	String	0	382
Rating	float64	0	22
Display	float64	0	43
Battery	int32	0	57
Price	int32	0	219
RAM	String	0	8
ROM	String	0	15
Expandable_Memory	String	258	10
Rear_Camera	String	0	72
Front_Camera	String	92	22
Processor	String	0	163
Brand_Name	category	0	13
Processor_Category	category	0	5
Price_Category	category	0	9
Battery_Category	category	0	6
Rating_Category	category	0	6

- Input and data cleansing files:
    - Input - Flipkart scrapped file
    - Data cleansing:
        - Processor category – to tag all processors from specific brand into one group
        - Brand Name – Based on product name bunch of phones from same brand are grouped into one

Below is the sample dataset:

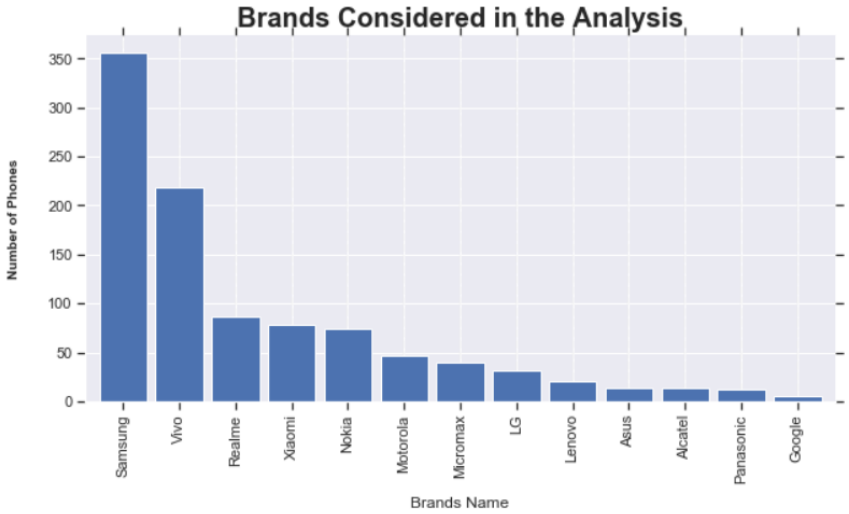
Product_Name	Rating	Display	Price	RAM	ROM	Expandable_Memory	Rear_Camera	Front_Camera	Battery	Processor	Brand_Name	Processor_Category
Realme Narzo 10 (That White, 128 GB)	4.5	6.5	11999	4 GB RAM	128 GB ROM	Expandable Upto 256 GB	48MP + 8MP + 2MP + 2MP	16MP	5000	MediaTek Helio G80 (12 nm)	Realme	MediaTek
Realme Narzo 10 (That White, 128 GB)	4.5	6.5	11999	4 GB RAM	128 GB ROM	Expandable Upto 256 GB	48MP + 8MP + 2MP + 2MP	16MP	5000	MediaTek Helio G80 (12 nm)	Realme	MediaTek
Realme Narzo 10 (That Green, 128 GB)	4.5	6.5	11999	4 GB RAM	128 GB ROM	Expandable Upto 256 GB	48MP + 8MP + 2MP + 2MP	16MP	5000	MediaTek Helio G80 (12 nm)	Realme	MediaTek
Realme Narzo 10 (That Green, 128 GB)	4.5	6.5	11999	4 GB RAM	128 GB ROM	Expandable Upto 256 GB	48MP + 8MP + 2MP + 2MP	16MP	5000	MediaTek Helio G80 (12 nm)	Realme	MediaTek
Motorola G8 Power Lite (Arctic Blue, 64 GB)	4.5	6.5	8999	4 GB RAM	64 GB ROM	Expandable Upto 256 GB	16MP + 2MP + 2MP	8MP	5000	MediaTek Helio P35	Motorola	MediaTek

Exploratory Data Analysis:

Which brand phone do you use?

```
x = data1.groupby('Brand_Name')['Product_Name'].count().sort_values(ascending=False)
ax = x.plot(kind='bar', figsize=(10, 5), zorder=2, width=0.85)
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
#ax.spines['left'].set_visible(False)
#ax.spines['bottom'].set_visible(False)
ax.set_title('Brands Considered in the Analysis', weight='bold', size=20)
ax.tick_params(axis="both", which="both", bottom="off", top="off", labelbottom="on", left="off", right="off", labelleft="on")
vals = ax.get_xticks()
for tick in vals:
    ax.axvline(x=tick, linestyle='dashed', alpha=0.4, color='#eeeeee', zorder=1)
# Set x-axis label
ax.set_xlabel("Brands Name", labelpad=10, size=12)
# Set y-axis label
ax.set_ylabel("Number of Phones", labelpad=20, weight='bold', size=10)

#for p in ax.patches:
    #width, height = p.get_width(), p.get_height()
    #x,y = p.get_xy()
    #ax.annotate('{:.0%}'.format(height), (x,y + height + 0.1 ))
```




```
data1['Brand_Name'].value_counts(sort=False)
```

```
Google        6
Micromax     40
Nokia        74
Xiaomi       78
Panasonic    12
LG           32
Motorola     46
Alcatel      14
Realme       86
Samsung     356
Lenovo       20
Vivo        218
Asus         14
```

```
(data1['Brand_Name'].value_counts(sort=False)/
     data1['Brand_Name'].count())*100
```

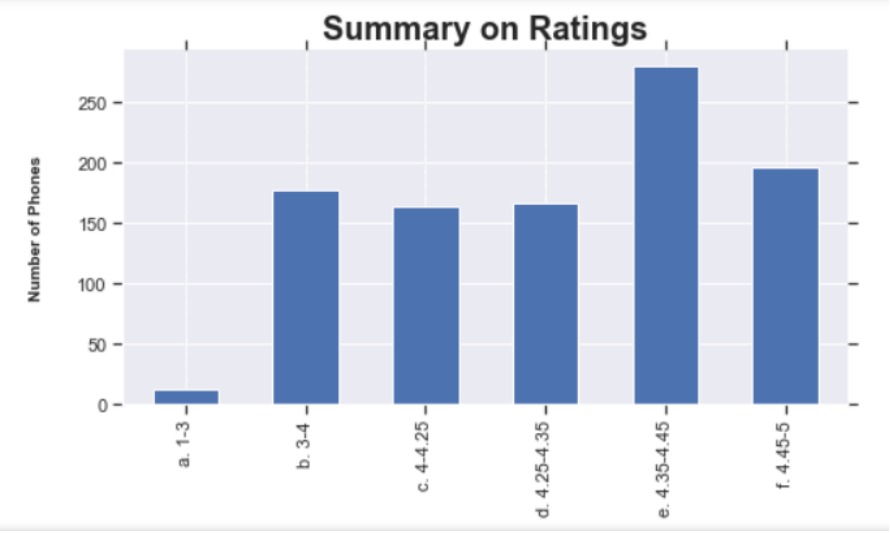
```
Google       0.602410
Micromax     4.016064
Nokia        7.429719
Xiaomi       7.831325
Panasonic    1.204819
LG           3.212851
Motorola     4.618474
Alcatel      1.405622
Realme       8.634538
Samsung     35.742972
Lenovo       2.008032
Vivo        21.887550
Asus         1.405622
```

Insights:

- There are 13 companies considered in this analysis
- Maximum number of records scrapped are from Samsung followed by Vivo, Realme and Xiaomi, top 4 brands constitute to 74% of the total data scrapped
- Google has least number of phones with 0.6% of the data

How much do you rate your phone?

```
x = data1.groupby('Rating_Category')['Product_Name'].count()
ax = x.plot(kind='bar', figsize=(8, 4), zorder=2, width=0.55)
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
#ax.spines['left'].set_visible(False)
#ax.spines['bottom'].set_visible(False)
ax.set_title('Summary on Ratings', weight='bold', size=20)
ax.tick_params(axis="both", which="both", bottom="off", top="off", labelbottom="on", left="off", right="off", labelleft="on")
vals = ax.get_xticks()
for tick in vals:
    ax.axvline(x=tick, linestyle='dashed', alpha=0.4, color='#eeeeee', zorder=1)
# Set x-axis label
ax.set_xlabel("Rating Category", labelpad=20, weight='bold', size=10)
# Set y-axis label
ax.set_ylabel("Number of Phones", labelpad=20, weight='bold', size=10)
```




```
data1['Rating_Category'].value_counts(sort=False)
```

```
a. 1-3          12
b. 3-4         178
c. 4-4.25      164
d. 4.25-4.35   166
e. 4.35-4.45   280
f. 4.45-5      196
```

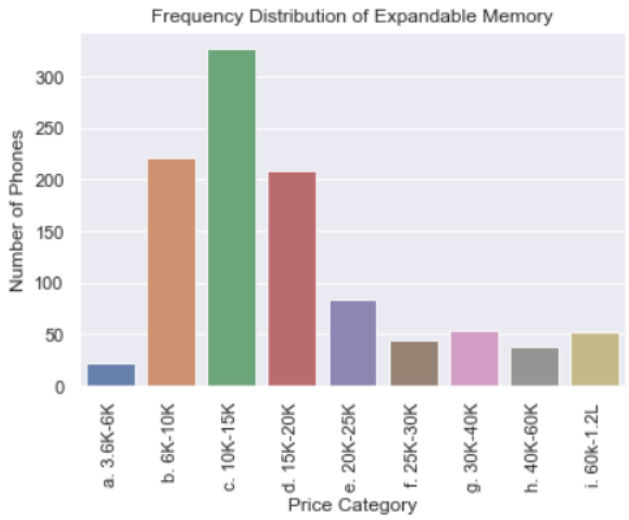
```
(data1['Rating_Category'].value_counts(sort=False)/
     data1['Rating_Category'].count())*100
```

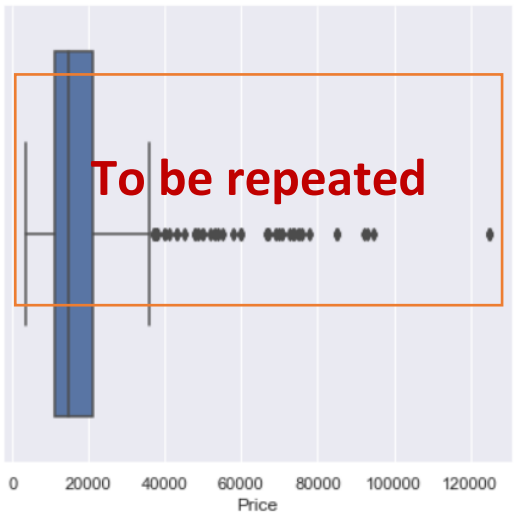
```
a. 1-3          1.204819
b. 3-4         17.871486
c. 4-4.25      16.465863
d. 4.25-4.35   16.666667
e. 4.35-4.45   28.112450
f. 4.45-5      19.678715
```

Insights:

- There are 190 records with rating <3, 19% of the population
- Between rating 4-5 we have 18% of the data i.e. 806 records.Rating 4-5 is further split to better understand the data.
- 61 % data is present within the range 4-4.45

What is the cost range of your phone?






```
data1['Price_Category'].value_counts(sort=False)
```

```
a. 3.6K-6K      24
b. 6K-10K      206
c. 10K-15K     264
d. 15K-20K     234
e. 20K-25K      68
f. 25K-30K      52
g. 30K-40K      46
h. 40K-60K      50
i. 60k-1.2L     52
Name: Price_Category, dtype: int64
```

```
(data1['Price_Category'].value_counts(sort=False)/
     data1['Price_Category'].count())*100
```

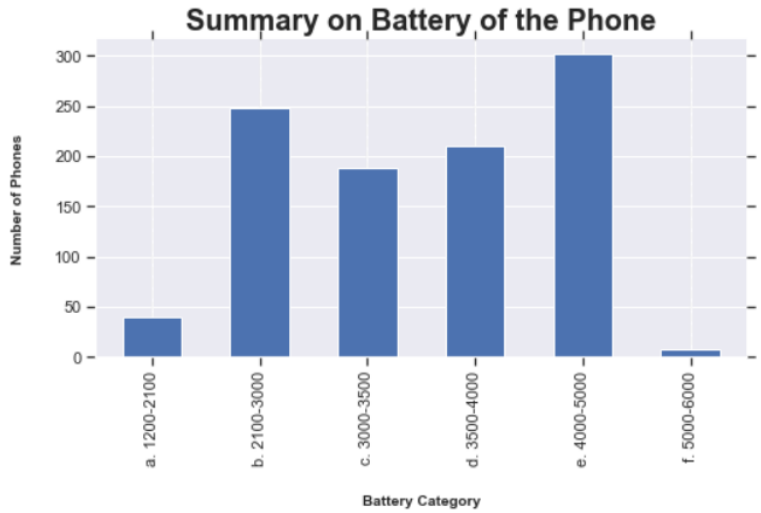
```
a. 3.6K-6K      2.409639
b. 6K-10K      20.682731
c. 10K-15K     26.506024
d. 15K-20K     23.493976
e. 20K-25K      6.827309
f. 25K-30K      5.220884
g. 30K-40K      4.618474
h. 40K-60K      5.020080
i. 60k-1.2L     5.220884
Name: Price_Category, dtype: float64
```

In sights:

- In our analysis 71% of the data has price range 6,000 to 20,000
- Using box plot we observe below Q1 and above Q3 line is near 40,000; after which we can consider it as outliers
- From the summary, records below 6K is 24 records and above 40K is 102 records. Together after excluding outliers we will be loosing 126 records which is 13% of the population
- Out of curiosity want to know which brands these 126 records belongs to:

Price_Category	Brand_Name	Product_Name
a. 3.6K-6K	Alcatel	2
	Micromax	12
	Nokia	4
	Panasonic	2
	Samsung	4
h. 40K-60K	LG	6
	Realme	6
	Samsung	36
	Xiaomi	2
i. 60k-1.2L	Google	2
	Motorola	4
	Samsung	46

Is your display and battery dependent on each other?




```
data1['Battery_Category'].value_counts(sort=False)
```

```
a. 1200-2100     40
b. 2100-3000    248
c. 3000-3500    188
d. 3500-4000    210
e. 4000-5000    302
f. 5000-6000      8
```

```
(data1['Battery_Category'].value_counts(sort=False)/
     data1['Battery_Category'].count())*100
```

```
a. 1200-2100     4.016064
b. 2100-3000    24.899598
c. 3000-3500    18.875502
d. 3500-4000    21.084337
e. 4000-5000    30.321285
f. 5000-6000     0.803213
```

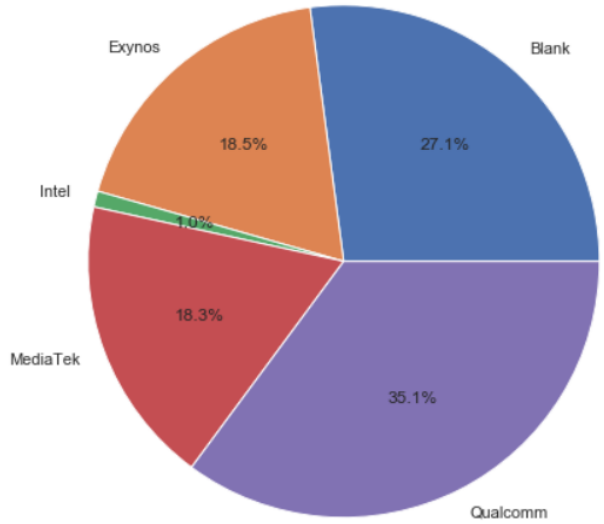
Insights:

- 95% has battery ranging from 2100 – 5000 mAh
- Looking at the summary table we can observe the categories with battery range 1200-2100 and 5000-6000 together constitutes to 4.8% of the population which is 48 records
- Is battery dependent on other features of the phone?

  We will look into it in bivariate analysis!

Processor Category:

```
labels = data1['Processor_Category'].astype('category').cat.categories.tolist()
counts = data1['Processor_Category'].value_counts()
sizes = [counts[var_cat] for var_cat in labels]
fig1, ax1 = plt.subplots()
ax1.pie(sizes, labels=labels, autopct='%1.1f%%', radius=2) #autopct is show the % on plot
#ax1.axis('equal')
plt.show()
```




```
data1['Processor_Category'].value_counts(sort=False)
```

```
Blank       270
MediaTek    182
Exynos      184
Qualcomm    350
Intel        10
```

```
(data1['Processor_Category'].value_counts(sort=False)/
     data1['Processor_Category'].count())*100
```

```
Blank       27.108434
MediaTek    18.273092
Exynos      18.473896
Qualcomm    35.140562
Intel        1.004016
```

Insights:

- Multiple versions from Processor are grouped into each of the above values
- Qualcomm takes a major share of 35% in the entire population; however, 27% is blank.
- If we consider in the data available; then Qualcomm share increases to 48%
- There is only 1% of data available with Intel processor, which is 10 record and can be removed as outlier

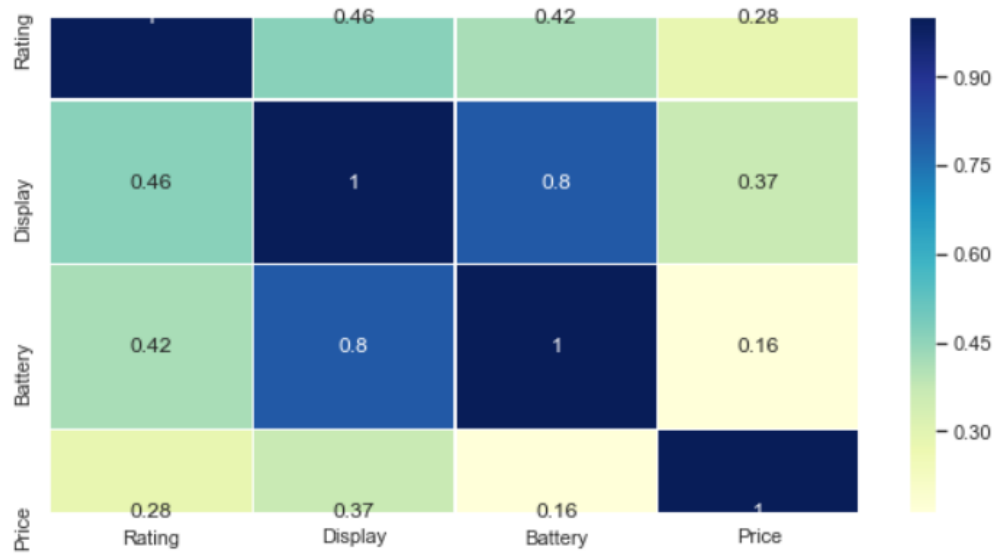
Bi-Variate Analysis

Correlation

```
corelation = data2.corr()
```

```
plt.figure(figsize=(10,5))
sns.heatmap(corelation,xticklabels=corelation.columns,
            yticklabels=corelation.columns,cmap="YlGnBu", annot=True, linewidth=0.5)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1b8db360088>
```

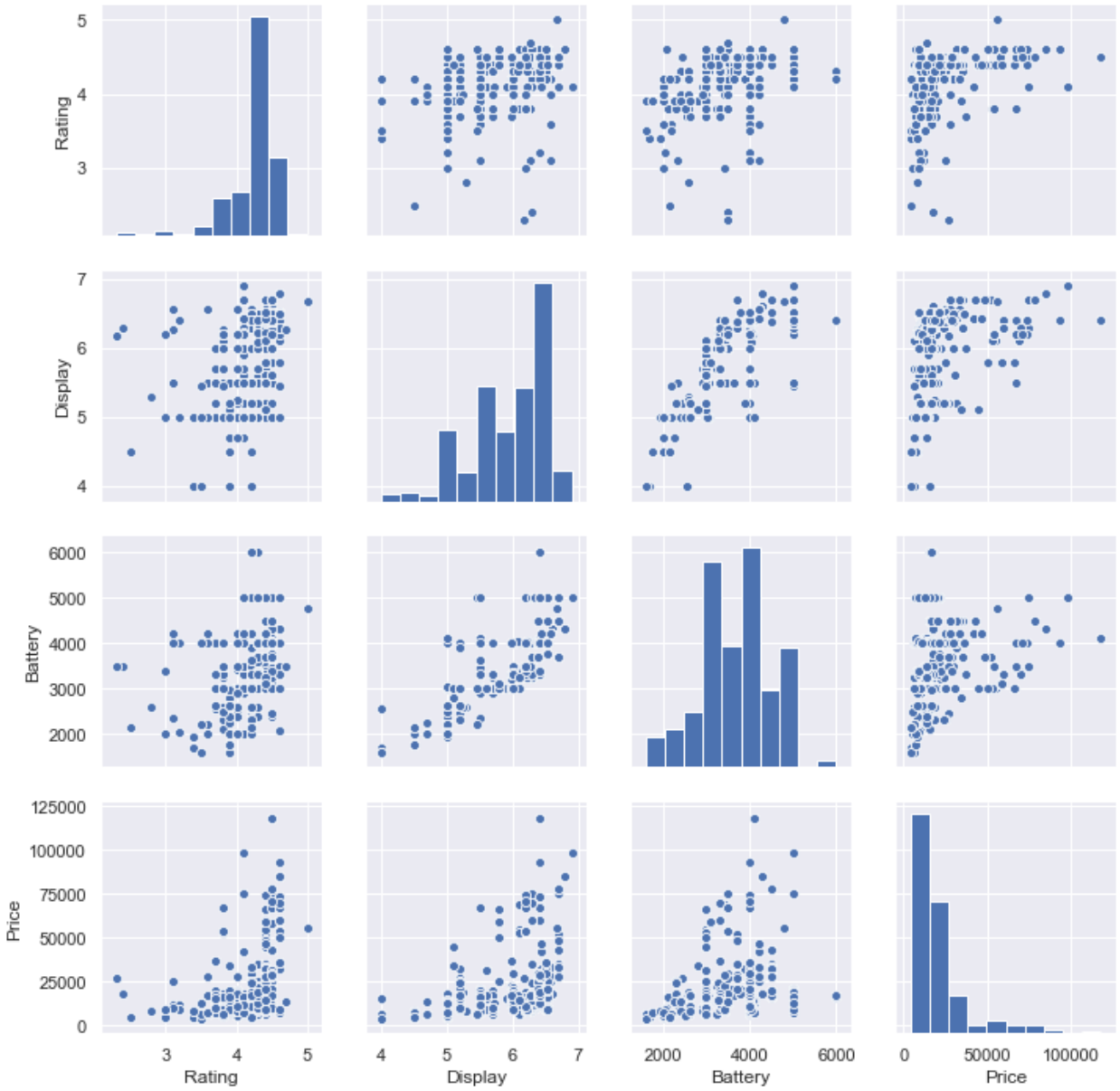


Insights:

Display and battery are strongly dependent on each other

In the next category we have Rating dependent on Display and Battery

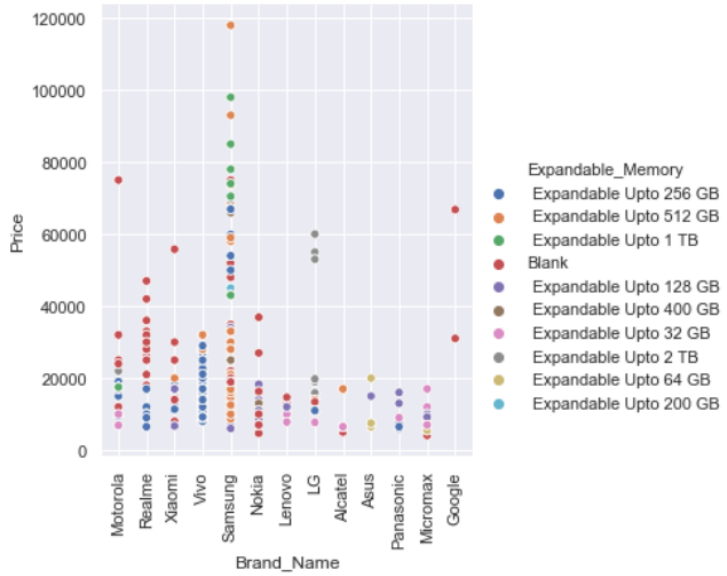
Pairpolt:


Rating
Display
Battery
Price
5
4
3
7
6
5
4
6000
5000
4000
3000
2000
125000
100000
75000
50000
25000
0
3
4
5
4
5
6
7
2000
4000
6000
0
50000
100000
Rating
Display
Battery
Price

Relation between Brand Name Price and Rating:

```
sns.relplot(x='Brand_Name', y='Price', hue='Expandable_Memory', data=data1)
plt.xticks(rotation='vertical')
#sns.barplot(carrier_count.index, carrier_count.values, alpha=0.9)
```

```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12],
 <a list of 13 Text xticklabel objects>)
```



Insights:

Very strong positive correlation would be between Display and Battery

Very weak positive correlation would be between Battery and Price

Relation between RAM, Price and Rating:

```
sns.relplot(x='RAM', y='Price', hue='Rating_Category', data=data1)
plt.xticks(rotation='vertical')
#sns.barplot(carrier_count.index, carrier_count.values, alpha=0.9)
```

```
([0, 1, 2, 3, 4, 5, 6, 7], <a list of 8 Text xticklabel objects>)
```