**Name:** Soham Belurgikar

**Roll No.:** 2019130006

**Course:** **DA (Data Analytics)**

**Assignment No.:** 1

**Part:** 1

**Name of the Assignment:** Exploratory Data Analysis

**Problem Statement:**

The smartphone market in 2022 is filled with variety of phones catering to every person’s needs. You can buy phones from brands like Samsung, Apple, Xiaomi, buy a phone which costs as low as Rs. 1000 or as high as Rs. 179900, buy phones with colours like Black, Blue, Rose Gold etc.

But which brand has sold the most phones? Which brand offers phones in all price ranges? Does the overall rating of a phone increase with its price? Which colour is the most in-demand?

This EDA aims to answer these questions with the help of statistics as well as plots.

**Implementation:**

[Dataset link](https://www.kaggle.com/devsubhash/flipkart-mobiles-dataset/)

[Colab link](https://colab.research.google.com/drive/1egjsbf4FfM_jrYObOskZOiPrdSIYACnp?usp=sharing)

**The dataset:**

The chosen dataset consists of 2647 samples with 8 attributes, namely:

* Brand - Name of the Mobile Manufacturer
* Model - Model name / number of the Mobile Phone
* Colour - Colour of the model. Missing or Null values indicate no specified colour of the model offered on the ecommerce website.
* Memory - RAM of the model (4GB, 6GB, 8GB, etc.)
* Storage - ROM of the model (32GB, 64GB, 128GB, 256GB, etc.)
* Rating - Rating of the model based on reviews (out of 5). Missing or Null values indicate there are no ratings present for the model.
* Selling Price- Selling Price/Discounted Price of the model in INR when this data was scraped. Ideally price indicates the discounted price of the model
* Original Price- Actual price of the model in INR. Missing values or null values would indicate that the product is being sold at the actual price available in the 'Price' column.

**Importing the required libraries:**

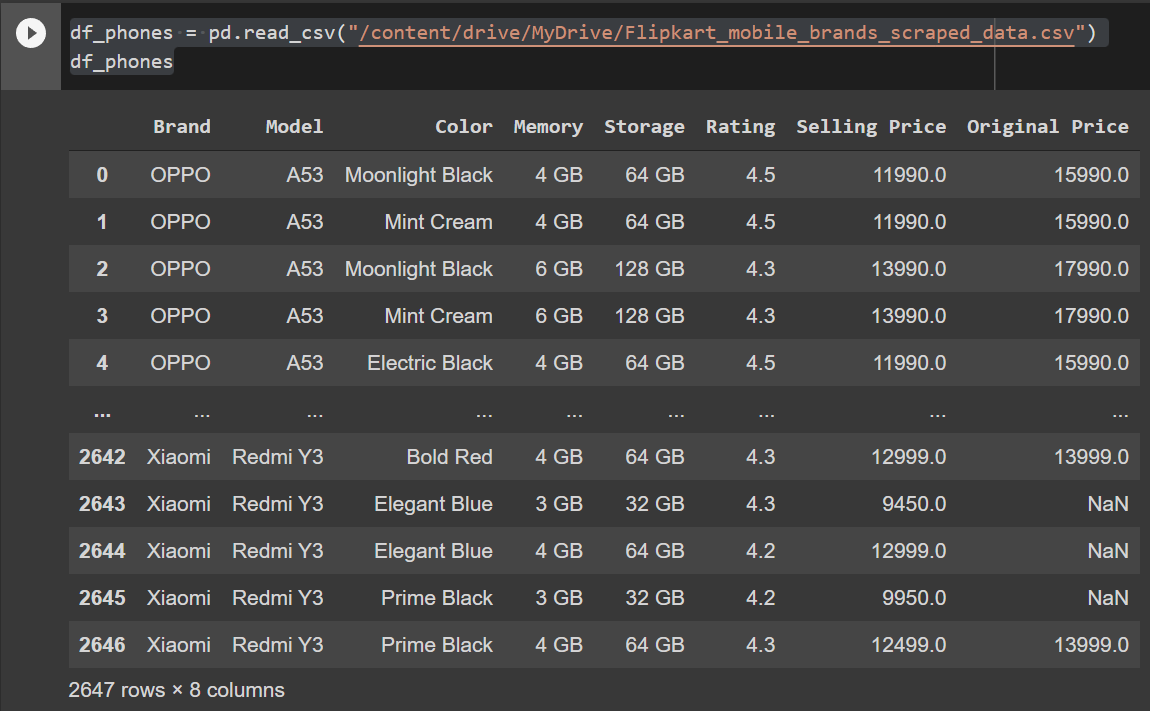
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

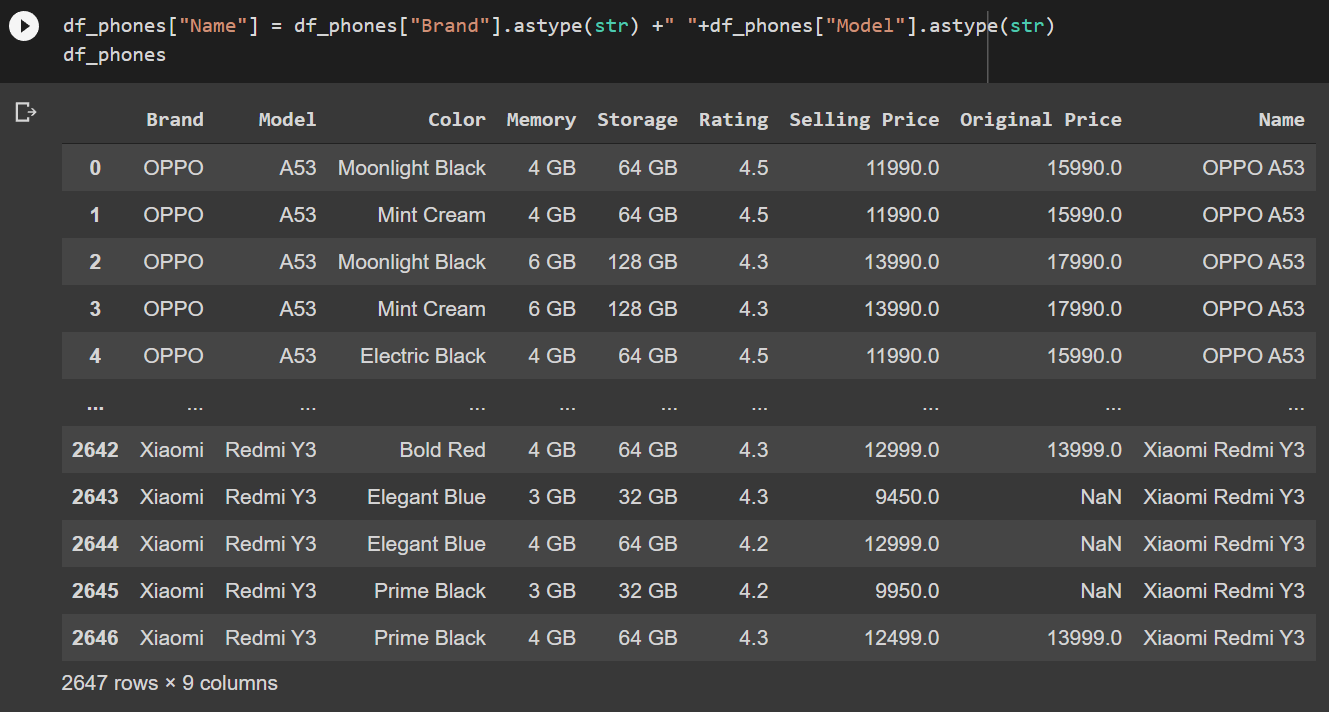
import seaborn as sns

**Loading the data into the dataframe:**

****

**Adding the Name column:**

Name of the phone = Name of Brand + Name of Model



df\_phones.shape

Using .shape() we can get information about the number of rows and columns of the dataset:

(2647, 9)

So, the dataset contains 2647 rows (samples) and 9 columns (features).

**Removing duplicate rows:**

duplicate\_rows\_df = df\_phones[df\_phones.duplicated()]

print("number of duplicate rows: ", duplicate\_rows\_df.shape)

This gives us the number of rows which have the same values for every column:

number of duplicate rows: (107, 9)

So, the dataset contained 107 rows which were duplicates.

df\_phones.count()

You can also check the number of rows that each column contains using the .count() method:

Brand 2647

Model 2645

Color 2505

Memory 2605

Storage 2568

Rating 2647

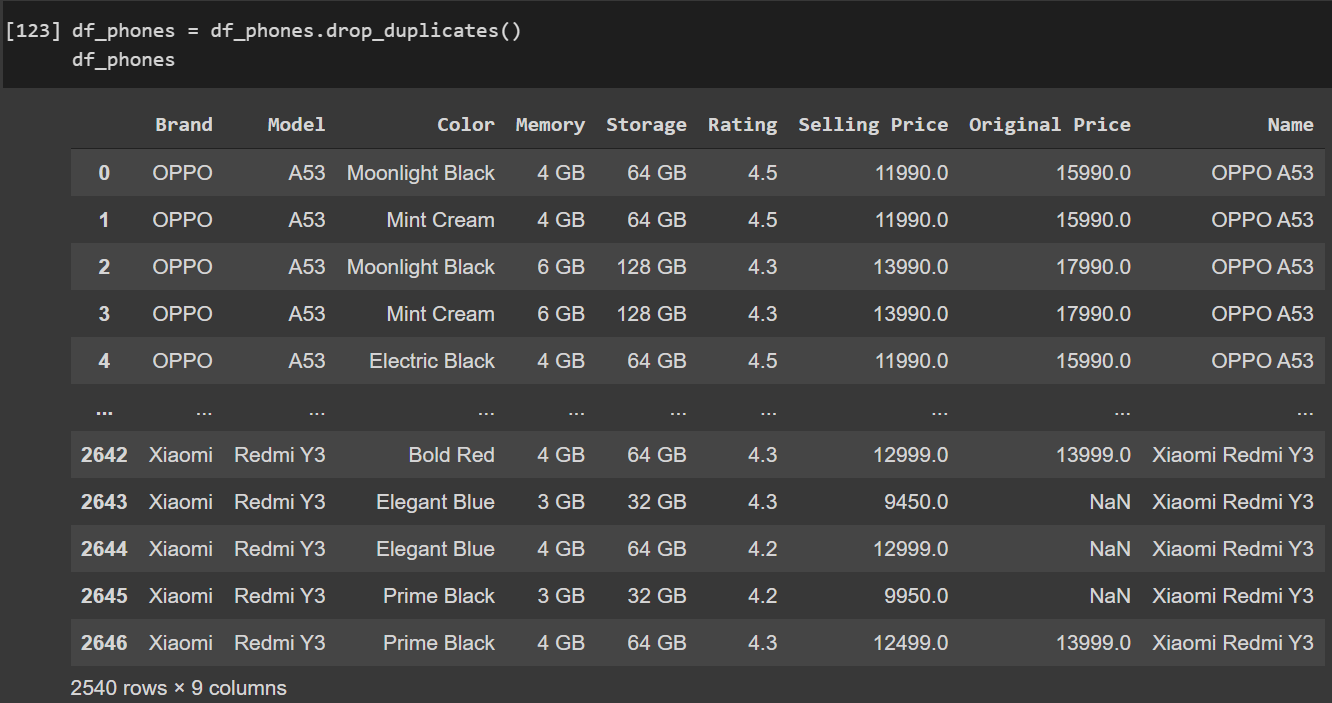
Selling Price 2644

Original Price 969

Name 2647

dtype: int64

You can delete the duplicate rows using just a simple method, i.e., .drop\_duplicates():



df\_phones.count()

Brand 2540

Model 2538

Color 2407

Memory 2501

Storage 2463

Rating 2540

Selling Price 2537

Original Price 934

Name 2540

dtype: int64

**Removing null / missing values:**

print(df\_phones.isnull().sum())

The .isnull().sum() command will return the number of values which are missing for every column:

Brand 0

Model 2

Color 133

Memory 39

Storage 77

Rating 0

Selling Price 3

Original Price 1606

Name 0

dtype: int64

We will drop lines with model unknown or missing memory information or missing storage information. Put missing value of colour to "Base". Drop lines with missing both prices else fill one with the other.

df\_phones = df\_phones.dropna(subset=["Model", "Memory","Storage"])

df\_phones["Selling Price"] = df\_phones["Selling Price"].fillna(df\_phones["Original Price"])

df\_phones["Original Price"] = df\_phones["Original Price"].fillna(df\_phones["Selling Price"])

df\_phones= df\_phones.dropna(subset=["Original Price","Selling Price"])

df\_phones["Color"] = df\_phones["Color"].fillna("Base")

print(df\_phones.isnull().sum())

Brand 0

Model 0

Color 0

Memory 0

Storage 0

Rating 0

Selling Price 0

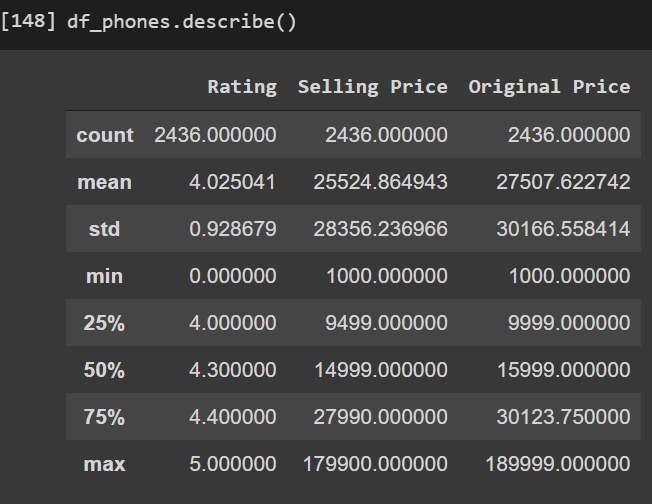
Original Price 0

Name 0

dtype: int64

Now our dataset is free of null values.

**Statistics:**



The .describe() method is very useful for calculating mean, standard deviation, range and percentiles of the data.

We can use .mode() for calculating mode of a particular column:

df\_phones['Selling Price'].mode()

0 9999.0

dtype: float64

For calculating the standard error, we can use .sem():

df\_phones['Selling Price'].sem()

574.5263541029967

Variance is calculated using .var():

df\_phones['Selling Price'].var()

804076174.8774368

Lastly, to calculate the coefficient of variation, we can divide the standard deviation by the mean:

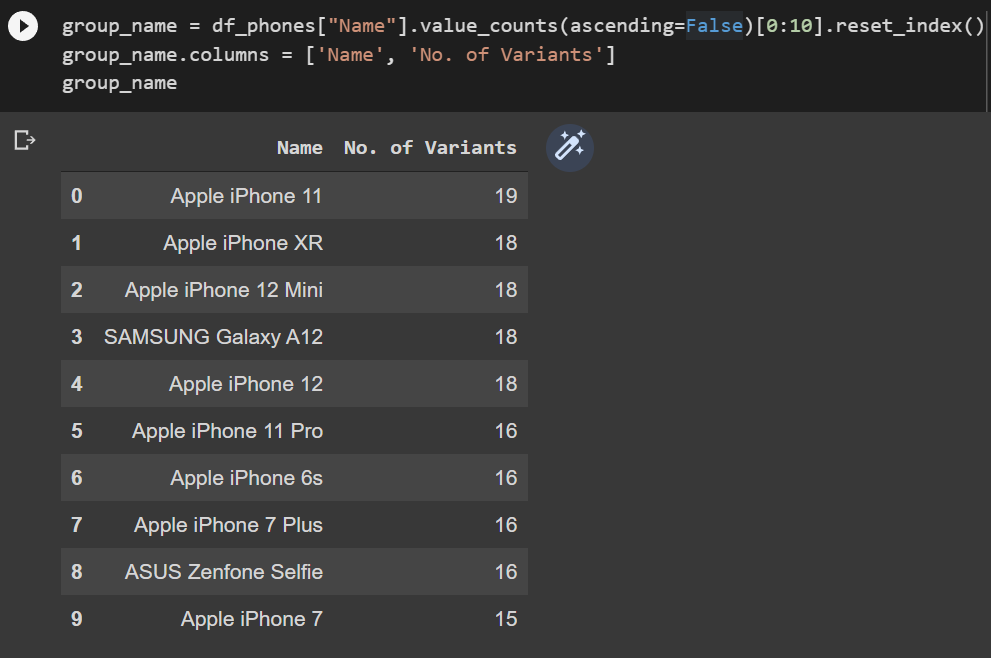
cv\_sell\_price = df\_phones['Selling Price'].std() / df\_phones['Selling Price'].mean()

cv\_sell\_price

1.110926033494914

**Plots:**

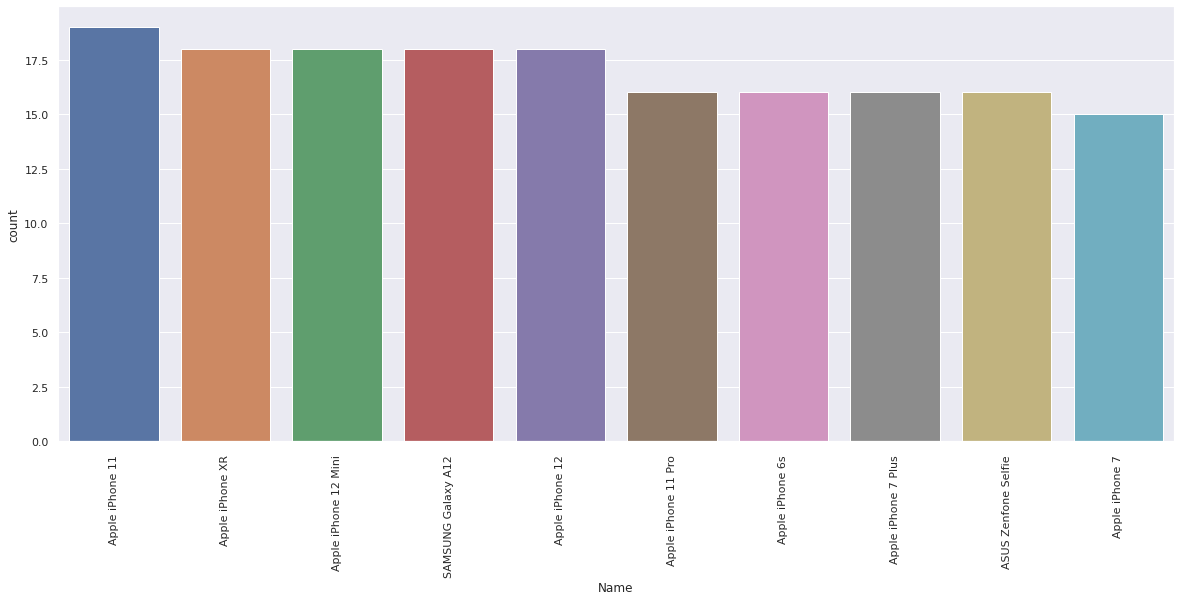
The models of a specific smartphone may differ in memory, storage or colour. Such variations can be found using .groupby:



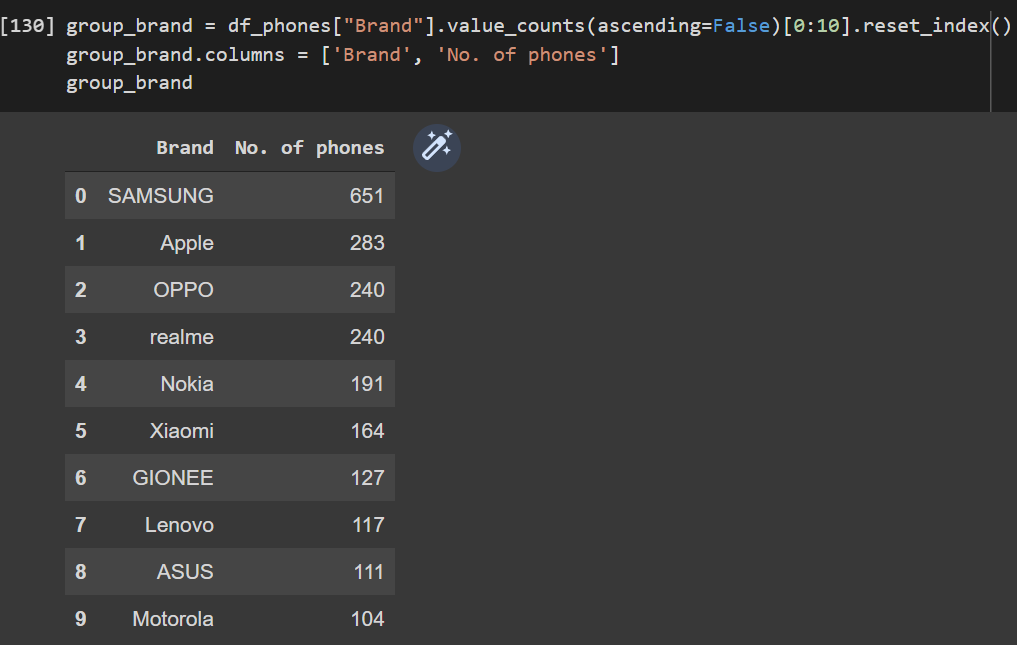
fig, ax = plt.subplots(figsize=(20,8))

ax=sns.countplot(x="Name", data=df\_phones, order=df\_phones['Name'].value\_counts(ascending = False)[:10].index)

plt.xticks(rotation = 90)

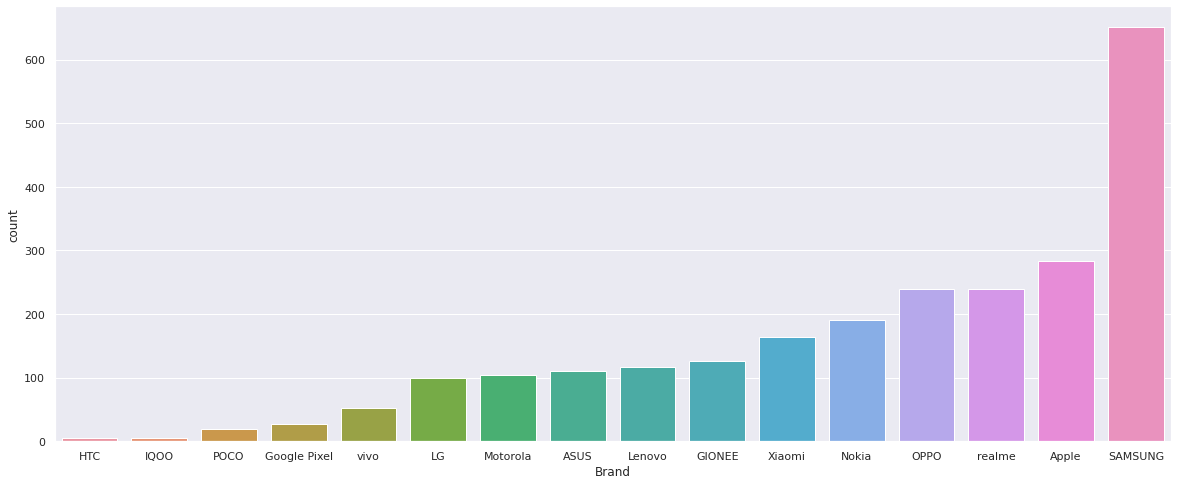


Calculating the no. of smartphones sold by brand:

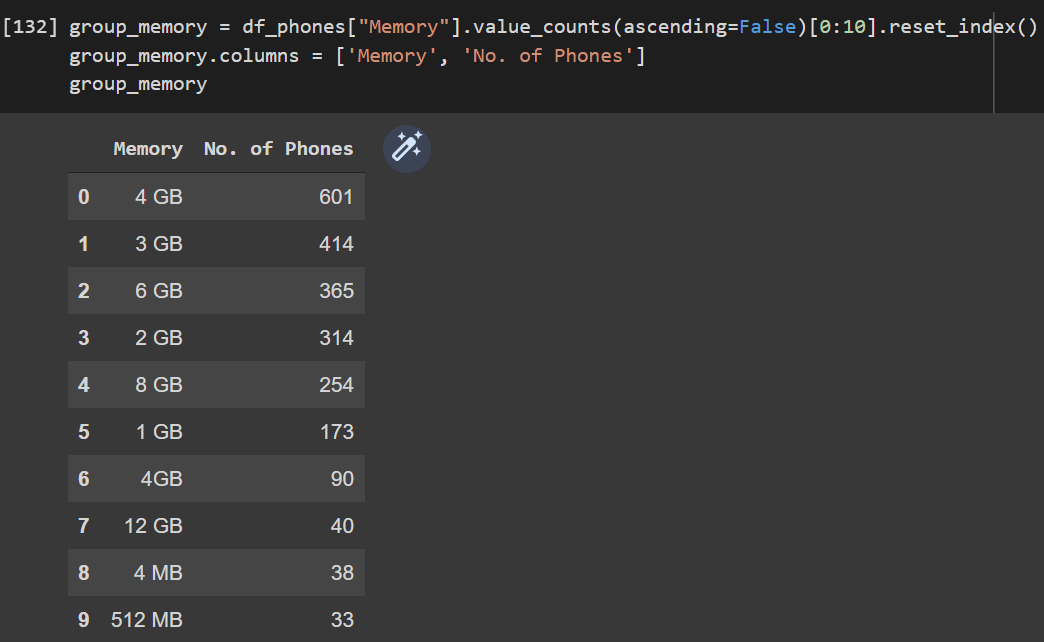


fig, ax = plt.subplots(figsize=(20,8))

ax=sns.countplot(x="Brand", data=df\_phones, order=df\_phones['Brand'].value\_counts(ascending = True).index)



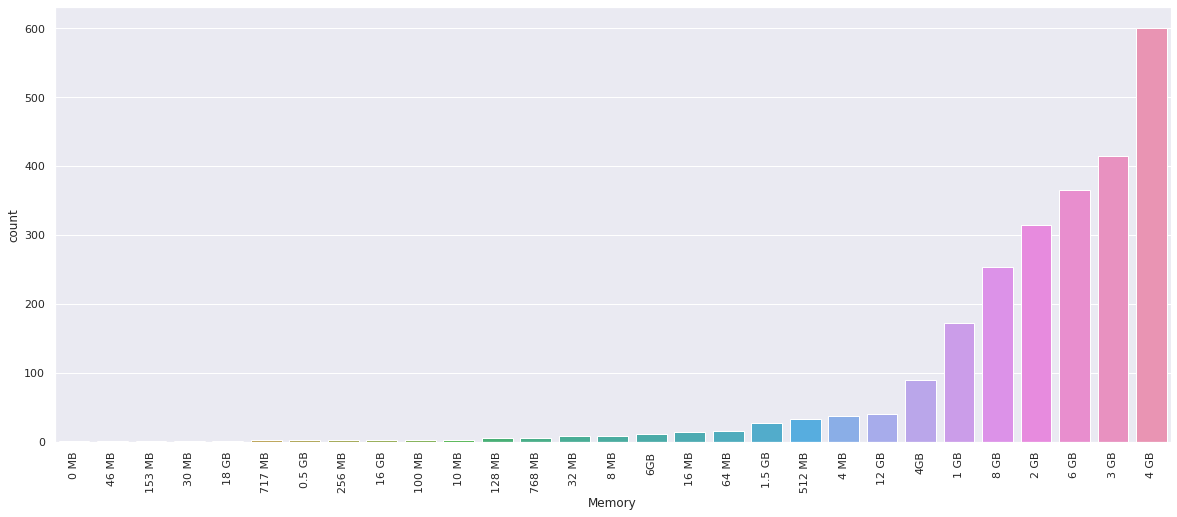
Calculating the no. of smartphones sold by memory:



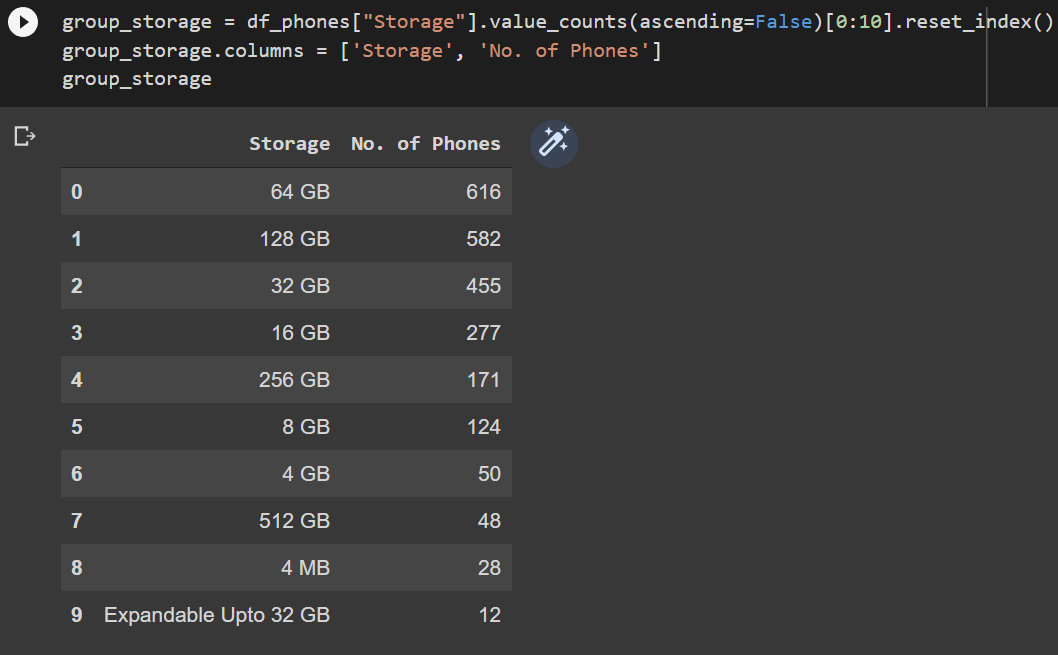
fig, ax = plt.subplots(figsize=(20,8))

ax=sns.countplot(x="Memory", data=df\_phones, order=df\_phones['Memory'].value\_counts(ascending = True).index)

plt.xticks(rotation = 90)



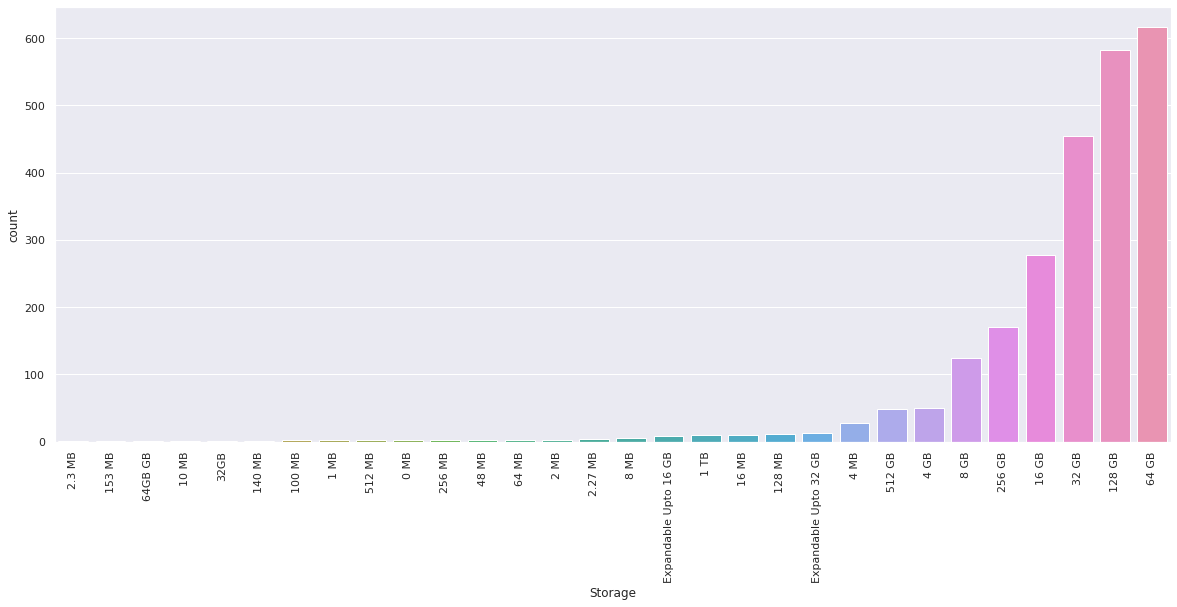
Calculating the no. of smartphones sold by storage:



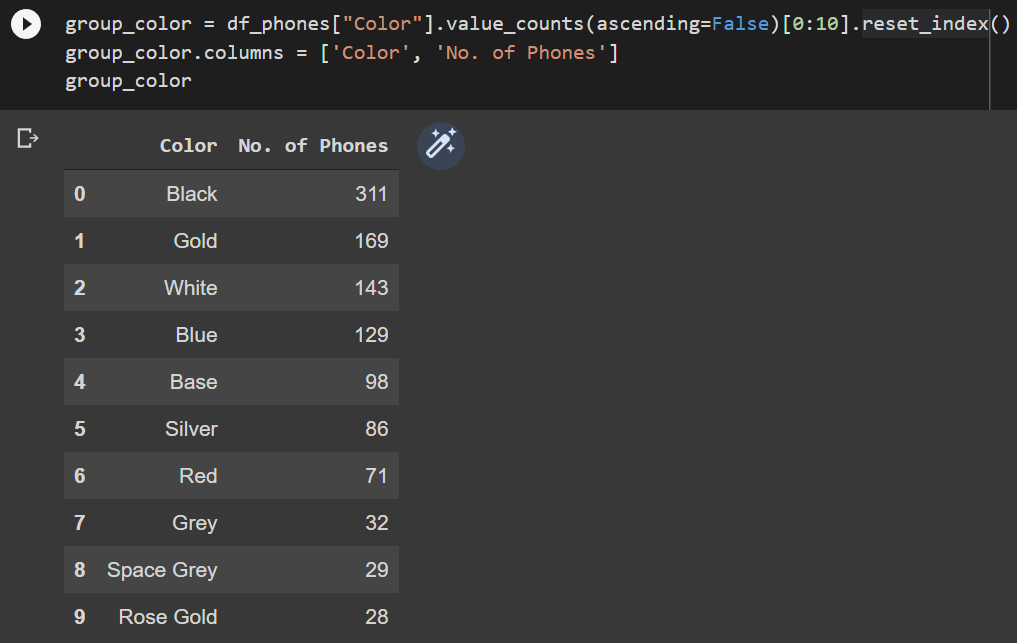
fig, ax = plt.subplots(figsize=(20,8))

ax=sns.countplot(x="Storage", data=df\_phones, order=df\_phones['Storage'].value\_counts(ascending = True).index)

plt.xticks(rotation = 90)

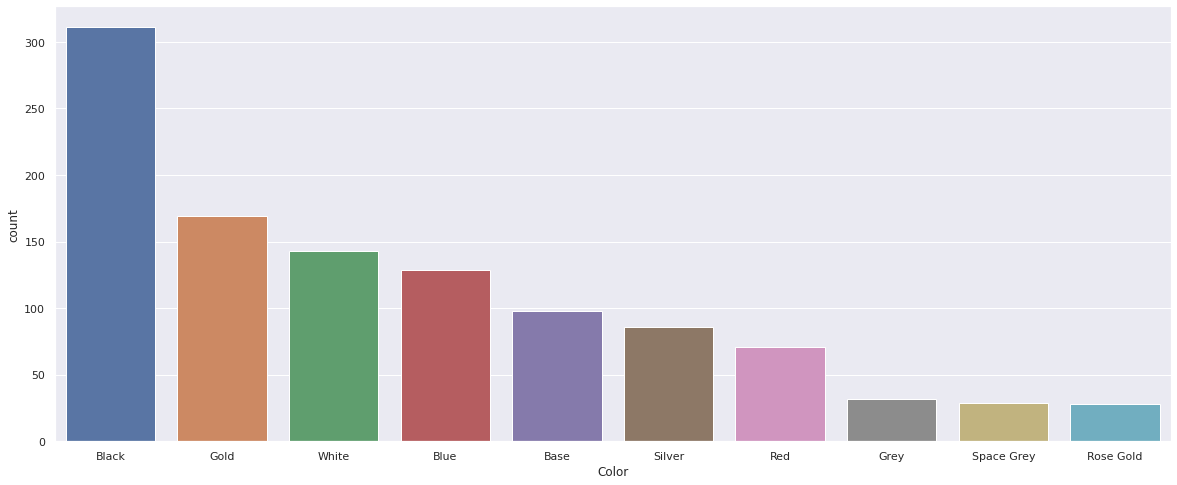


Calculating the no. of smartphones sold by colour:



fig, ax = plt.subplots(figsize=(20,8))

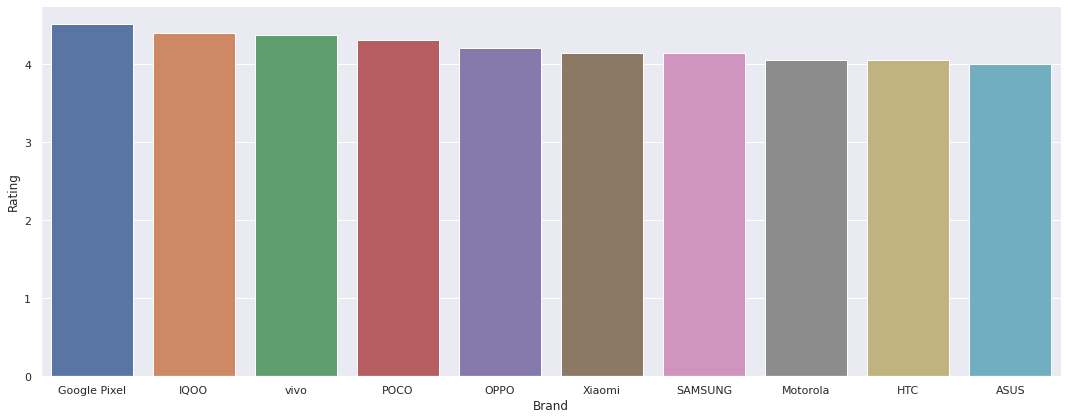
ax=sns.countplot(x="Color", data=df\_phones, order=df\_phones['Color'].value\_counts(ascending = False)[:10].index)



Top 10 brands sorted by average rating:



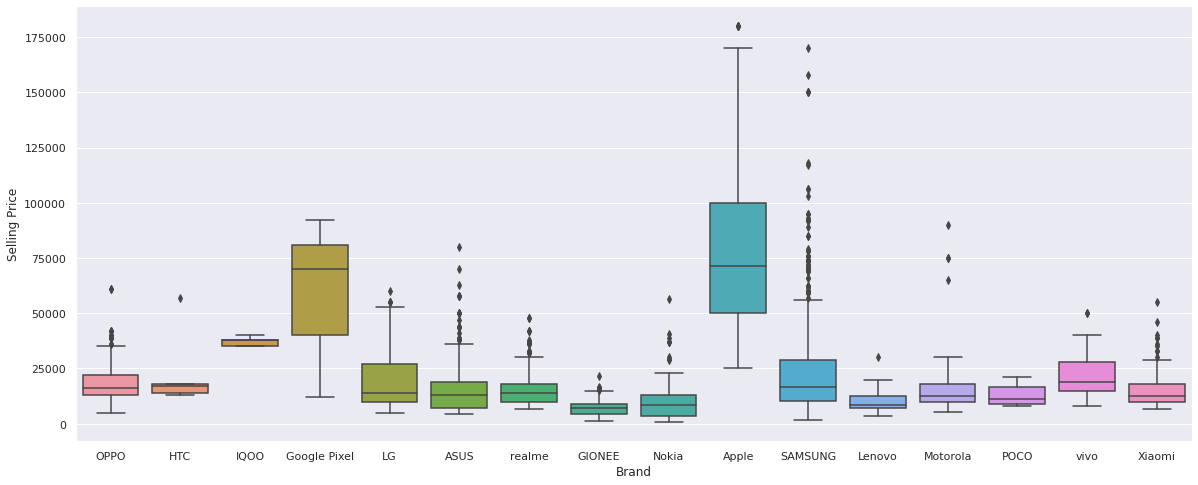
sns.catplot(x="Brand", y="Rating", kind="bar", data=phone\_rating, height=6, aspect=2.5)



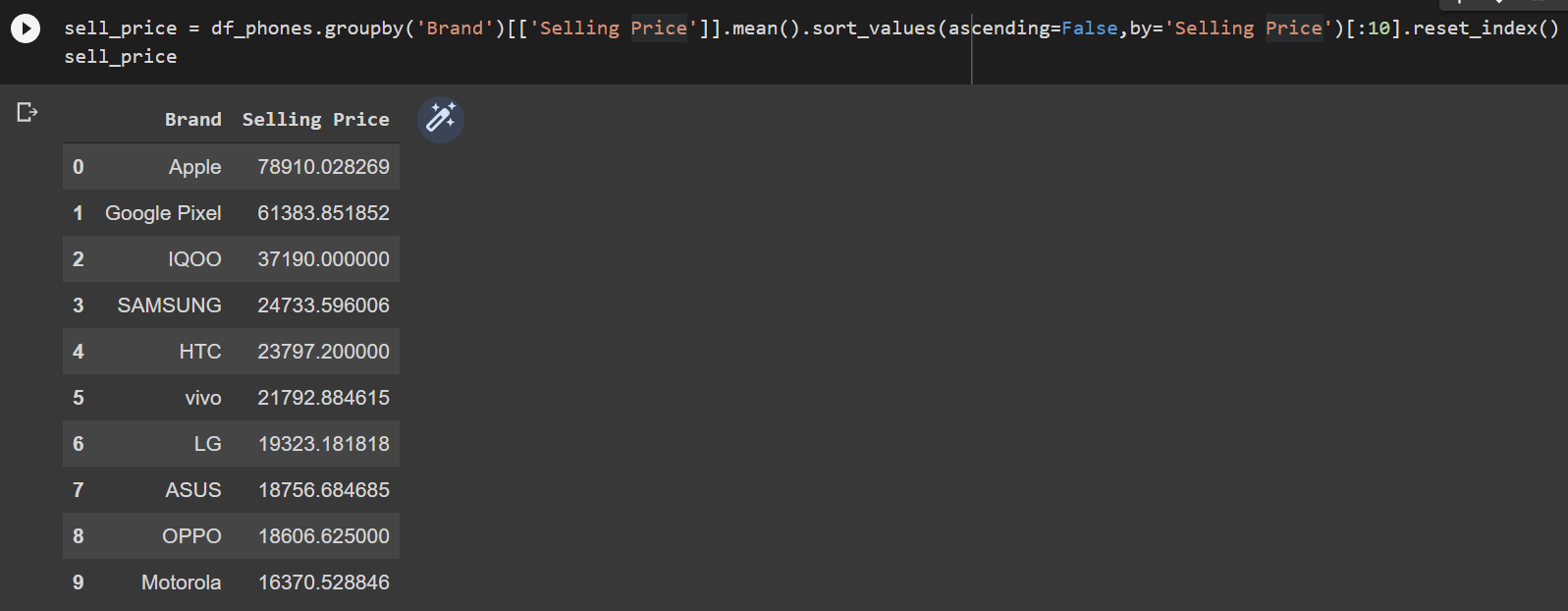
Finding outliers for selling price of every brand:

sns.set(rc={'figure.figsize':(20,8)})

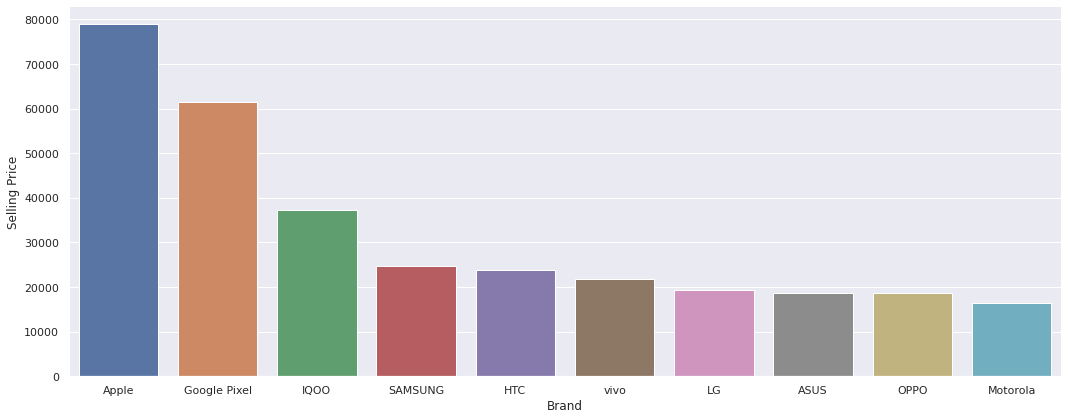
sns.boxplot(x="Brand", y="Selling Price", data=df\_phones)



Top 10 brands sorted by average selling price:



sns.catplot(x="Brand", y="Selling Price", kind="bar", data=sell\_price, height=6, aspect=2.5)



**Heat Map:**

We can find the dependent variables using heat map.

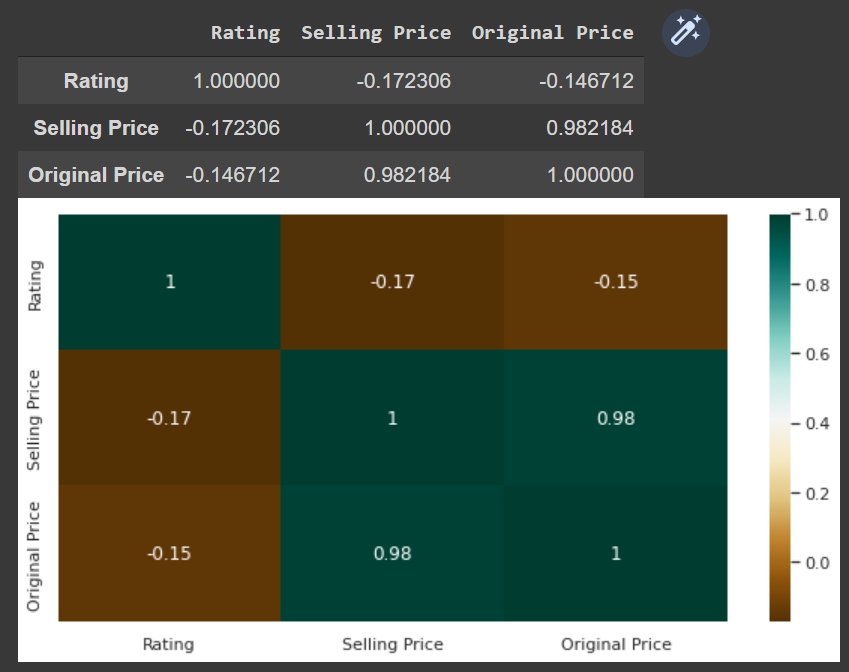
plt.figure(figsize=(10, 5))

c = df\_phones.corr()

sns.heatmap(c,cmap="BrBG",annot=True)

c

It first finds the correlation between any two variables and then plots it along with a colour theme using the seaborn library:



Relation between Selling Price and Rating:

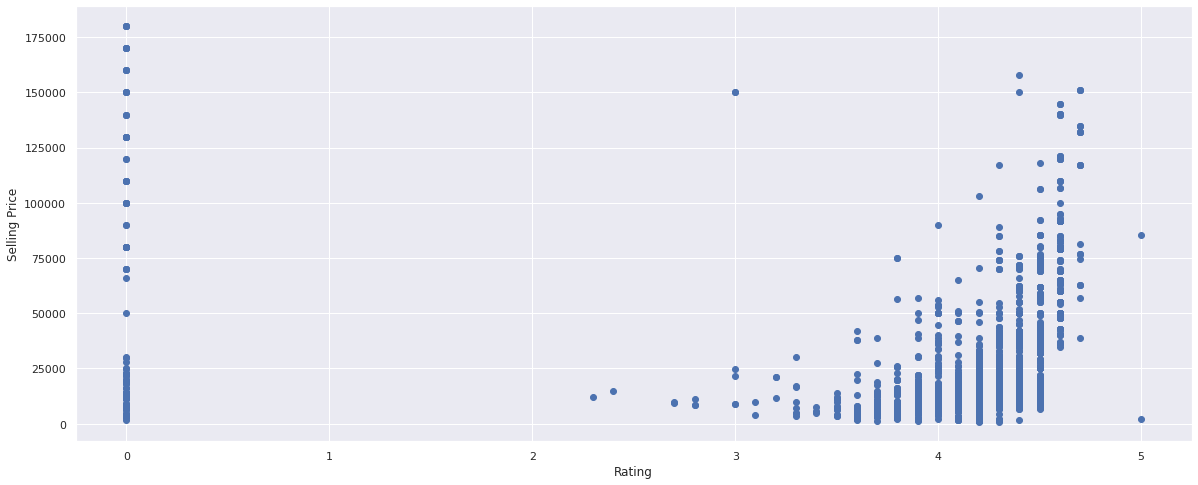
fig, ax = plt.subplots(figsize=(20, 8))

ax.scatter(df\_phones['Rating'], df\_phones['Selling Price'])

ax.set\_xlabel('Rating')

ax.set\_ylabel('Selling Price')

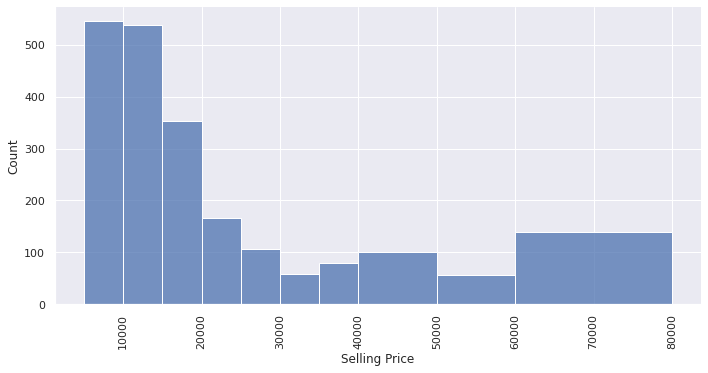
plt.show()



No. of smartphones by price range:

sns.displot(df\_phones, x='Selling Price',bins=[5000,10000,15000,20000,25000,30000,35000,40000,50000,60000,80000], aspect=2)

plt.xticks(rotation = 90)



**Conclusion:**   
  
The mean rating of all smartphones was 4.02 whereas the mean selling price and original price were Rs. 25524.86 and Rs. 27507.62 respectively.  
  
We can see that no. of variants for Apple iPhones are higher compared to other brands, meaning that Apple offers more variety in terms of colour, memory and storage.  
  
Samsung dominates the smartphone market by selling the most number of smartphones, followed by Apple, OPPO and Realme.  
  
In terms of specifications of mobile phones, 4GB (RAM) and 64 GB (Storage) are the most common types.  
  
Black is the most in-demand colour, followed by Gold and White.  
  
Google Pixel is the highest rated smartphone brand with over 4.5 average rating. Xiaomi and Samsung have the 5th and 6th highest average rating respectively, while Apple doesn't even make it to the top 10.  
  
Selling Price of Samsung contains a relatively higher number of outliers, which can suggest that while Samsung sells majority of its phones in the lower and mid-budget price bracket, it also contains a substantial amount of phones in the high-budget price bracket.  
  
Apple is quite rightly famous for being the most expensive brand, with its average selling price of almost Rs. 79000.  
  
The scatterplot suggests that there might be a positive direct relationship between rating and Selling Price, since Selling Price increases as Rating increases.  
  
The displot allows us to look at each price range and count the no. of phones in it. The low-budget price range is clearly the winner, suggesting that the population prefers a cheaper smartphone even if it sacrifices on some of the specs.

**References:** [Dataset link](https://www.kaggle.com/devsubhash/flipkart-mobiles-dataset/)