Amazon Sales - Analysis

INTRODUCTION

This dataset consists more than 1000 of real products with their identification number listed in the Amazon marketplace specifically from the region India. I noticed the region due to the currency used in the dataset is Rupee India. My objective is to clean and prepare the data due to the raw data being very unorganized. I will then move on to finding insights about the data and try to elaborate in the form of visualization.

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
#Importing files
df=pd.read_csv('amazon.csv')
df
```

Out[3]:

	product_id	product_name	category	discounted_price
0	B07JW9H4J1	Wayona Nylon Braided USB to Lightning Fast Cha...	Computers&Accessories|Accessories&Peripherals|...	₹39
1	B098NS6PVG	Ambrane Unbreakable 60W / 3A Fast Charging 1.5...	Computers&Accessories|Accessories&Peripherals|...	₹19
2	B096MSW6CT	Sounce Fast Phone Charging Cable & Data Sync U...	Computers&Accessories|Accessories&Peripherals|...	₹19
3	B08HDJ86NZ	boAt Deuce USB 300 2 in 1 Type-C & Micro USB S...	Computers&Accessories|Accessories&Peripherals|...	₹32
4	B08CF3B7N1	Portronics Konnect L 1.2M Fast Charging 3A 8 P...	Computers&Accessories|Accessories&Peripherals|...	₹15
...	...	...	...	..
1460	B08L7J3T31	Noir Aqua - 5pcs PP Spun Filter + 1 Spanner | ...	Home&Kitchen|Kitchen&HomeAppliances|WaterPurif...	₹37
1461	B01M6453MB	Prestige Delight PRWO Electric Rice Cooker (1 ...	Home&Kitchen|Kitchen&HomeAppliances|SmallKitch...	₹2,28
1462	B009P2LIL4	Bajaj Majesty RX10 2000 Watts Heat Convector R...	Home&Kitchen|Heating,Cooling&AirQuality|RoomHe...	₹2,21
1463	B00J5DYCCA	Havells Ventil Air DSP 230mm Exhaust Fan (Pist...	Home&Kitchen|Heating,Cooling&AirQuality|Fans|E...	₹1,39
1464	B01486F4G6	Borosil Jumbo 1000-Watt Grill Sandwich Maker (...	Home&Kitchen|Kitchen&HomeAppliances|SmallKitch...	₹2,86

1465 rows × 16 columns

In [6]:
```
#Checking the columns names
df.columns
```

Out[6]:
```
Index(['product_id', 'product_name', 'category', 'discounted_price',
       'actual_price', 'discount_percentage', 'rating', 'rating_count',
       'about_product', 'user_id', 'user_name', 'review_id', 'review_title',
       'review_content', 'img_link', 'product_link'],
      dtype='object')
```

In [7]:
```
#checking First Few Rows
df.head()
```

Out[7]:

	product_id	product_name	category	discounted_price	actu
0	B07JW9H4J1	Wayona Nylon Braided USB to Lightning Fast Cha...	Computers&Accessories|Accessories&Peripherals|...	₹399	
1	B098NS6PVG	Ambrane Unbreakable 60W / 3A Fast Charging 1.5...	Computers&Accessories|Accessories&Peripherals|...	₹199	
2	B096MSW6CT	Sounce Fast Phone Charging Cable & Data Sync U...	Computers&Accessories|Accessories&Peripherals|...	₹199	
3	B08HDJ86NZ	boAt Deuce USB 300 2 in 1 Type-C & Micro USB S...	Computers&Accessories|Accessories&Peripherals|...	₹329	
4	B08CF3B7N1	Portronics Konnect L 1.2M Fast Charging 3A 8 P...	Computers&Accessories|Accessories&Peripherals|...	₹154	

In [8]:
```
#checking the datatype
df.dtypes
```

Out[8]:
```
product_id              object
product_name            object
category                object
discounted_price        object
actual_price            object
discount_percentage     object
rating                  object
rating_count            object
about_product           object
user_id                 object
user_name               object
review_id               object
review_title            object
review_content          object
img_link                object
product_link            object
dtype: object
```

In [9]:
```
#Changeing the data type of Discounted_price and actual_price
df['discounted_price']=df['discounted_price'].str.replace("₹",'')
df['discounted_price']=df['discounted_price'].str.replace(",",'')
df['discounted_price']=df['discounted_price'].astype('float64')

df['actual_price']=df['actual_price'].str.replace("₹",'')
df['actual_price']=df['actual_price'].str.replace(",",'')
df['actual_price']=df['actual_price'].astype('float64')
```

In [10]:
```
#Changeing data type values in Discount Percentange
df['discount_percentage']=df['discount_percentage'].str.replace('%','').astype('floate
df['discount_percentage']=df['discount_percentage']/100
df['discount_percentage']
```

Out[10]:
```
0       0.64
1       0.43
2       0.90
3       0.53
4       0.61
        ...
1460    0.59
1461    0.25
1462    0.28
1463    0.26
1464    0.22
Name: discount_percentage, Length: 1465, dtype: float64
```

In [11]:
```
#Finding unsual string in the rating column
df['rating'].value_counts()
```

Out[11]:
```
4.1    244
4.3    230
4.2    228
4.0    129
3.9    123
4.4    123
3.8     86
4.5     75
4       52
3.7     42
3.6     35
3.5     26
4.6     17
3.3     16
3.4     10
4.7      6
3.1      4
5.0      3
3.0      3
4.8      3
3.2      2
2.8      2
2.3      1
|        1
2        1
3        1
2.6      1
2.9      1
Name: rating, dtype: int64
```

In [12]:
```
#Insecting the row
df.query('rating == "|"')
```

Out[12]:

	product_id	product_name	category	discounted_price
1279	B08L12N5H1	Eureka Forbes car Vac 100 Watts Powerful Sucti...	Home&Kitchen|Kitchen&HomeAppliances|Vacuum,Cle...	2099.0

i went to the amazon website and found the similar product id with the same product having the rating of 4 . so i am going to give the item rating of 4.0 Providing the website link:https://www.amazon.in/Eureka-Forbes-Vacuum-Cleaner-Washable/dp/B08L12N5H1

In [13]:
```
#changing rating column datatype
df['rating']=df['rating'].str.replace("|",'4.0').astype('float64')
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_8060\2915955404.py:2: FutureWarning: The
default value of regex will change from True to False in a future version. In additio
n, single character regular expressions will *not* be treated as literal strings when
regex=True.
  df['rating']=df['rating'].str.replace("|",'4.0').astype('float64')
```

In [27]:
```
#changing the rating_count data type
df['rating_count']=df['rating_count'].replace(",",'').astype('float64')
```

In [28]:
```
#checking duplicates
duplicates=df.duplicated()
df[duplicates]
```

Out[28]:

product_id	product_name	category	discounted_price	actual_price	discount_percentage	rating	ra

In [29]:
```
#Rechecking missing values
df.isnull().sum()
```

Out[29]:
```
product_id             0
product_name           0
category               0
discounted_price       0
actual_price           0
discount_percentage    0
rating                 0
rating_count           0
about_product          0
user_id                0
user_name              0
review_id              0
review_title           0
review_content         0
img_link               0
product_link           0
dtype: int64
```

In [30]:
```
#filing null value with the mode
df['rating_count'].fillna(df['rating_count'].mode()[0],inplace=True)
```

In [31]:
```
#Recheching the missing values
df.isnull().sum()
```

Out[31]:
```
product_id             0
product_name           0
category               0
discounted_price       0
actual_price           0
discount_percentage    0
rating                 0
rating_count           0
about_product          0
user_id                0
user_name              0
review_id              0
review_title           0
review_content         0
img_link               0
product_link           0
dtype: int64
```

In [32]:
```
#Creating New data Frame with selected columns
df1=df[['product_id','product_name','category','discounted_price','actual_price','disc
```

In [33]:
```
#Splitting the strings into category column
catsplit=df['category'].str.split('|',expand=True)
```

```
catsplit
```

Out[33]:

	0	1	2	
0	Computers&Accessories	Accessories&Peripherals	Cables&Accessories	Cab
1	Computers&Accessories	Accessories&Peripherals	Cables&Accessories	Cab
2	Computers&Accessories	Accessories&Peripherals	Cables&Accessories	Cab
3	Computers&Accessories	Accessories&Peripherals	Cables&Accessories	Cab
4	Computers&Accessories	Accessories&Peripherals	Cables&Accessories	Cab
...	...	...	...	
1460	Home&Kitchen	Kitchen&HomeAppliances	WaterPurifiers&Accessories	WaterPurifierAccessor
1461	Home&Kitchen	Kitchen&HomeAppliances	SmallKitchenAppliances	Rice&PastaCook
1462	Home&Kitchen	Heating,Cooling&AirQuality	RoomHeaters	HeatConvect
1463	Home&Kitchen	Heating,Cooling&AirQuality	Fans	ExhaustFa
1464	Home&Kitchen	Kitchen&HomeAppliances	SmallKitchenAppliances	SandwichMak

1465 rows × 7 columns

In [34]:
```
catsplit=catsplit.rename(columns={0:'category_1',1:'category_2',2:'category_3'})
```

In [35]:
```
#Adding column into New Dataframe
df1['category_1']=catsplit['category_1']
df1['category_2']=catsplit['category_2']
df1.drop(columns='category',inplace=True)
df1
```

Out[35]:

	product_id	product_name	discounted_price	actual_price	discount_percentage	rating	rating
0	B07JW9H4J1	Wayona Nylon Braided USB to Lightning Fast Cha...	399.0	1099.0	0.64	4.2	
1	B098NS6PVG	Ambrane Unbreakable 60W / 3A Fast Charging 1.5...	199.0	349.0	0.43	4.0	
2	B096MSW6CT	Sounce Fast Phone Charging Cable & Data Sync U...	199.0	1899.0	0.90	3.9	
3	B08HDJ86NZ	boAt Deuce USB 300 2 in 1 Type-C & Micro USB S...	329.0	699.0	0.53	4.2	
4	B08CF3B7N1	Portronics Konnect L 1.2M Fast Charging 3A 8 P...	154.0	399.0	0.61	4.2	
...	...	...	...	...	...	...	
1460	B08L7J3T31	Noir Aqua - 5pcs PP Spun Filter + 1 Spanner | ...	379.0	919.0	0.59	4.0	
1461	B01M6453MB	Prestige Delight PRWO Electric Rice Cooker (1 ...	2280.0	3045.0	0.25	4.1	
1462	B009P2LIL4	Bajaj Majesty RX10 2000 Watts Heat Convector R...	2219.0	3080.0	0.28	3.6	
1463	B00J5DYCCA	Havells Ventil Air DSP 230mm Exhaust Fan (Pist...	1399.0	1890.0	0.26	4.0	
1464	B01486F4G6	Borosil Jumbo 1000-Watt Grill Sandwich Maker (...	2863.0	3690.0	0.22	4.3	

1465 rows × 9 columns

In [36]:
```
#Counting Values in Category_1 column
df1['category_1'].value_counts()
```

Out[36]:
```
Electronics              526
Computers&Accessories    453
Home&Kitchen             448
OfficeProducts            31
MusicalInstruments         2
HomeImprovement            2
Toys&Games                 1
Car&Motorbike              1
Health&PersonalCare        1
Name: category_1, dtype: int64
```

In [37]:
```
#Arranging Srtings in category_1 column
df1['category_1']=df1['category_1'].str.replace('&',' & ')
df1['category_1']=df1['category_1'].str.replace('OfficeProducts','Office Products')
df1['category_1']=df1['category_1'].str.replace('MusicalInstruments','Musical Instrume
df1['category_1']=df1['category_1'].str.replace('HomeImprovement','Home Improvement')
```

In [38]:
```
#Counting values in category_2 column
df1['category_2'].value_counts()
```

Out[38]:
```
Accessories&Peripherals                    381
Kitchen&HomeAppliances                     308
HomeTheater,TV&Video                       162
Mobiles&Accessories                        161
Heating,Cooling&AirQuality                 116
WearableTechnology                          76
Headphones,Earbuds&Accessories              66
NetworkingDevices                           34
OfficePaperProducts                         27
ExternalDevices&DataStorage                 18
Cameras&Photography                         16
HomeStorage&Organization                    16
HomeAudio                                   16
GeneralPurposeBatteries&BatteryChargers     14
Accessories                                 14
Printers,Inks&Accessories                   11
CraftMaterials                               7
Components                                   5
OfficeElectronics                            4
Electrical                                   2
Monitors                                     2
Microphones                                  2
Arts&Crafts                                  1
PowerAccessories                             1
Tablets                                      1
Laptops                                      1
Kitchen&Dining                               1
CarAccessories                               1
HomeMedicalSupplies&Equipment                1
Name: category_2, dtype: int64
```

In [39]:
```
#Arranging strings in category_2 columns
df1['category_1']=df1['category_1'].str.replace('&',' & ')
df1['category_1']=df1['category_1'].str.replace(',',', ')
df1['category_1']=df1['category_1'].str.replace('HomeAppliances','Home Appliances')
df1['category_1']=df1['category_1'].str.replace('HomeTheater','Home Theater')
```

```
df1['category_1']=df1['category_1'].str.replace('WearableTechnology','Wearable Technol
df1['category_1']=df1['category_1'].str.replace('NetworkingDevices','Networking Device
df1['category_1']=df1['category_1'].str.replace('OfficePaperProducts','Office Paper Pr
df1['category_1']=df1['category_1'].str.replace('ExternalDevices','External Devices')
df1['category_1']=df1['category_1'].str.replace('DataStorage','Data Storage')
df1['category_1']=df1['category_1'].str.replace('HomeStorage','Home Storage')
df1['category_1']=df1['category_1'].str.replace('HomeAudio ','Home Audio')
df1['category_1']=df1['category_1'].str.replace('GeneralPurposeBatteries','General Pur
df1['category_1']=df1['category_1'].str.replace('BatteryChargers','Battery Chargers')
df1['category_1']=df1['category_1'].str.replace('CraftMaterials','Craft Materials')
df1['category_1']=df1['category_1'].str.replace('OfficeElectronics','Office Electronic
df1['category_1']=df1['category_1'].str.replace('PowerAccessories','Power Accessories'
df1['category_1']=df1['category_1'].str.replace('CarAccessories','Car Accessories')
df1['category_1']=df1['category_1'].str.replace('HomeMedicalSupplies','Home Medical Su
```

In [40]:
```
#Removing wide space from Product_id
df1['product_id'].str.strip()
```

Out[40]:
```
0       B07JW9H4J1
1       B098NS6PVG
2       B096MSW6CT
3       B08HDJ86NZ
4       B08CF3B7N1
           ...
1460    B08L7J3T31
1461    B01M6453MB
1462    B009P2LIL4
1463    B00J5DYCCA
1464    B01486F4G6
Name: product_id, Length: 1465, dtype: object
```

In [41]:
```
#Creating Categories for Rankings
rating_score=[]
for score in df1['rating']:
    if score <2.0 : rating_score.append('Poor')
    elif score < 3.0 : rating_score.append('Below Average')
    elif score < 4.0 : rating_score.append('Average')
    elif score < 5.0 : rating_score.append('Above Average')
    elif score ==5.0 : rating_score.append('Excellent')
```

Created a a Rating Category that consists of:

1. Score below 2.0 = Poor
2. Score range of 2.0 - 2.9 = Below Average
3. Score range of 3.0 - 3.9 = Average
4. Score Range of 4.0 - 4.9 = Above Average
5. Score of 5.0 = Excellent

In [42]:
```
#Creating the new column changing the datatype
df1['rating_score'] =rating_score
df1['rating_score'] =df1['rating_score'].astype('category')
```

In [43]:
```
#Reordering Categories
df1['rating_score']=df1['rating_score'].cat.reorder_categories(['Below Average','Avera
                                                                'Excellent'],ordered=T
```

In [44]:
```
#Creating Difference of price column
df1['difference_price']=df1['actual_price']-df1['discounted_price']
```

In [45]:
```
#Result After Cleaning
df1.head()
```

Out[45]:

	product_id	product_name	discounted_price	actual_price	discount_percentage	rating	rating_co
0	B07JW9H4J1	Wayona Nylon Braided USB to Lightning Fast Cha...	399.0	1099.0	0.64	4.2	2426
1	B098NS6PVG	Ambrane Unbreakable 60W / 3A Fast Charging 1.5...	199.0	349.0	0.43	4.0	4399
2	B096MSW6CT	Sounce Fast Phone Charging Cable & Data Sync U...	199.0	1899.0	0.90	3.9	792
3	B08HDJ86NZ	boAt Deuce USB 300 2 in 1 Type-C & Micro USB S...	329.0	699.0	0.53	4.2	9436
4	B08CF3B7N1	Portronics Konnect L 1.2M Fast Charging 3A 8 P...	154.0	399.0	0.61	4.2	1690

In [46]:
```
#Subsetting Reviewing Identification
reviewers=df[['user_id','user_name']]
reviewers
```

Out[46]:

	user_id	user_name
0	AG3D6O4STAQKAY2UVGEUV46KN35Q,AHMY5CWJMMK5BJRBB...	Manav,Adarsh gupta,Sundeep,S.Sayeed Ahmed,jasp...
1	AECPFYFQVRUWC3KGNLJIOREFP5LQ,AGYYVPDD7YG7FYNBX...	ArdKn,Nirbhay kumar,Sagar Viswanathan,Asp,Plac...
2	AGU3BBQ2V2DDAMOAKGFAWDDQ6QHA,AESFLDV2PT363T2AQ...	Kunal,Himanshu,viswanath,sai niharka,saqib mal...
3	AEWAZDZZJLQUYVOVGBEUKSLXHQ5A,AG5HTSFRRE6NL3M5S...	Omkar dhale,JD,HEMALATHA,Ajwadh a.,amar singh ...
4	AE3Q6KSUK5P75D5HFYHCRAOLODSA,AFUGIFH5ZAFXRDSZH...	rahuls6099,Swasat Borah,Ajay Wadke,Pranali,RVK...
...	...	...
1460	AHITFY6AHALOFOHOZEOC6XBP4FEA,AFRABBODZJZQB6Z4U...	Prabha ds,Raghuram bk,Real Deal,Amazon Custome...
1461	AFG5FM3NEMOL6BNFRV2NK5FNJCHQ,AGEINTRN6Z563RMLH...	Manu Bhai,Naveenpittu,Evatira Sangma,JAGANNADH...
1462	AGVPWCMAHYQWJOQKMUJN4DW3KM5Q,AF4Q3E66MY4SR7YQZ...	Nehal Desai,Danish Parwez,Amazon Customer,Amaz...
1463	AF2JQCLSCY3QJATWUNNHUSVUPNQQ,AFDMLUXC5LS5RXDJS...	Shubham Dubey,E.GURUBARAN,Mayank S.,eusuf khan...
1464	AFGW5PT3R6ZAVQR4Y5MWVAKBZAYA,AG7QNJ2SCS5VS5VYY...	Rajib,Ajay B,Vikas Kahol,PARDEEP,Anindya Prama...

1465 rows × 2 columns

In [47]:
```
#Splitting user_id
splitting_user_id=reviewers['user_id'].str.split(',',expand=False)
splitting_user_id
```

Out[47]:
```
0       [AG3D6O4STAQKAY2UVGEUV46KN35Q, AHMY5CWJMMK5BJR...
1       [AECPFYFQVRUWC3KGNLJIOREFP5LQ, AGYYVPDD7YG7FYN...
2       [AGU3BBQ2V2DDAMOAKGFAWDDQ6QHA, AESFLDV2PT363T2...
3       [AEWAZDZZJLQUYVOVGBEUKSLXHQ5A, AG5HTSFRRE6NL3M...
4       [AE3Q6KSUK5P75D5HFYHCRAOLODSA, AFUGIFH5ZAFXRDS...
                              ...
1460    [AHITFY6AHALOFOHOZEOC6XBP4FEA, AFRABBODZJZQB6Z...
1461    [AFG5FM3NEMOL6BNFRV2NK5FNJCHQ, AGEINTRN6Z563RM...
1462    [AGVPWCMAHYQWJOQKMUJN4DW3KM5Q, AF4Q3E66MY4SR7Y...
1463    [AF2JQCLSCY3QJATWUNNHUSVUPNQQ, AFDMLUXC5LS5RXD...
1464    [AFGW5PT3R6ZAVQR4Y5MWVAKBZAYA, AG7QNJ2SCS5VS5V...
Name: user_id, Length: 1465, dtype: object
```

In [48]:
```
#Making user_id Display 1 per Row
reviewer_exp_id=splitting_user_id.explode()
reviewer_clean_id=reviewer_exp_id.reset_index(drop=True)
reviewer_clean_id
```

Out[48]:
```
0        AG3D6O4STAQKAY2UVGEUV46KN35Q
1        AHMY5CWJMMK5BJRBBSNLYT3ONILA
2        AHCTC6ULH4XB6YHDY6PCH2R772LQ
3        AGYHHIERNXKA6P5T7CZLXKVPT7IQ
4        AG4OGOFWXJZTQ2HKYIOCOY3KXF2Q
                    ...
11498    AHXCDNSXAESERITAFELQABFVNLCA
11499    AGRZD6CHLCUNOLMMIMIHUCG7PIFA
11500    AFQZVGSOSOJHKFQQMCEI4725QEKQ
11501    AEALVGXXIP46OZVXKRUXSDWZJMEA
11502    AGEFL3AY7YXEFZA4ZJU3LP7K7OJQ
Name: user_id, Length: 11503, dtype: object
```

In [49]:
```
#Splitting user_name
splitting_user_name=reviewers['user_name'].str.split(',',expand=False)
splitting_user_name
```

Out[49]:
```
0       [Manav, Adarsh gupta, Sundeep, S.Sayeed Ahmed,...
1       [ArdKn, Nirbhay kumar, Sagar Viswanathan, Asp,...
2       [Kunal, Himanshu, viswanath, sai niharka, saqi...
3       [Omkar dhale, JD, HEMALATHA, Ajwadh a., amar s...
4       [rahuls6099, Swasat Borah, Ajay Wadke, Pranali...
                              ...
1460    [Prabha ds, Raghuram bk, Real Deal, Amazon Cus...
1461    [Manu Bhai, Naveenpittu, Evatira Sangma, JAGAN...
1462    [Nehal Desai, Danish Parwez, Amazon Customer, ...
1463    [Shubham Dubey, E.GURUBARAN, Mayank S., eusuf ...
1464    [Rajib, Ajay B, Vikas Kahol, PARDEEP, Anindya ...
Name: user_name, Length: 1465, dtype: object
```

In [50]:
```
#Making user_name Display 1 per Row
reviewer_exp_name=splitting_user_name.explode()
reviewer_clean_name=reviewer_exp_name.reset_index(drop=True)
reviewer_clean_name
```

Out[50]:
```
0                  Manav
1            Adarsh gupta
2                 Sundeep
3           S.Sayeed Ahmed
4           jaspreet singh
                ...
11510             PARDEEP
11511    Anindya Pramanik
11512         Vikas Singh
11513     Harshada Pimple
11514              Saw a.
Name: user_name, Length: 11515, dtype: object
```

In [51]:
```
#Coverting 2 dataframes to merge
df21=pd.DataFrame(data=reviewer_clean_id)
df22=pd.DataFrame(data=reviewer_clean_name)
```

In [52]:
```
#Merging 2 DataFramea
df2=pd.merge(df21,df22,left_index=True,right_index=True)
```

In [53]:
```
df2.head()
```

Out[53]:

	user_id	user_name
0	AG3D6O4STAQKAY2UVGEUV46KN35Q	Manav
1	AHMY5CWJMMK5BJRBBSNLYT3ONILA	Adarsh gupta
2	AHCTC6ULH4XB6YHDY6PCH2R772LQ	Sundeep
3	AGYHHIERNXKA6P5T7CZLXKVPT7IQ	S.Sayeed Ahmed
4	AG4OGOFWXJZTQ2HKYIOCOY3KXF2Q	jaspreet singh

DATA EXPLORATION

In this stage I will try to elaborate my insights through Visualizations, Pivot Tables, and short explanations.

In [43]:
```
#Setting Visualization Styles
sns.set_style(style='darkgrid')
sns.set_palette(palette='icefire')
```

Observation 1: Product Category

Below are the list of Main Category and Sub-Category to help determine which sub-category belongs to which main category:

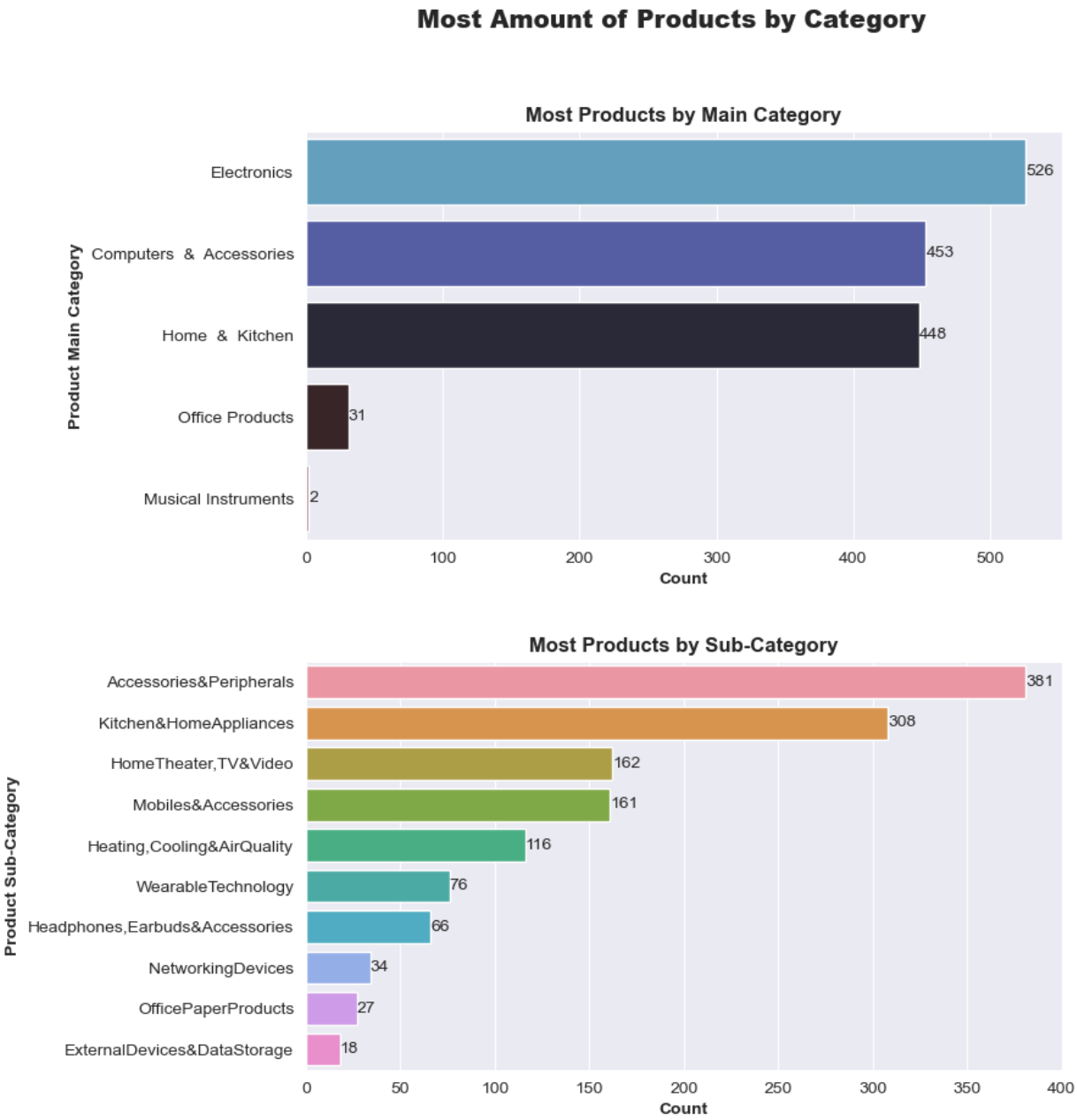
In [44]:
```
#Main category and sub_category
main_sub=df1[['category_1','category_2','product_id']]
main_sub=main_sub.rename(columns={'category_1':'Main Category','category_2':'Sub-Categ
main_sub_piv=pd.pivot_table(main_sub, index=['Main Category', 'Sub-Category'], aggfunc
main_sub_piv
```

Out[44]:

Main Category	Sub-Category	Producd ID
Car & Motorbike	CarAccessories	1
Computers & Accessories	Accessories&Peripherals	381
	Components	5
	ExternalDevices&DataStorage	18
	Laptops	1
	Monitors	2
	NetworkingDevices	34
	Printers,Inks&Accessories	11
	Tablets	1
Electronics	Accessories	14
	Cameras&Photography	16
	GeneralPurposeBatteries&BatteryChargers	14
	Headphones,Earbuds&Accessories	66
	HomeAudio	16
	HomeTheater,TV&Video	162
	Mobiles&Accessories	161
	PowerAccessories	1
	WearableTechnology	76
Health & PersonalCare	HomeMedicalSupplies&Equipment	1
Home & Kitchen	CraftMaterials	7
	Heating,Cooling&AirQuality	116
	HomeStorage&Organization	16
	Kitchen&Dining	1
	Kitchen&HomeAppliances	308
Home Improvement	Electrical	2
Musical Instruments	Microphones	2
Office Products	OfficeElectronics	4
	OfficePaperProducts	27
Toys & Games	Arts&Crafts	1

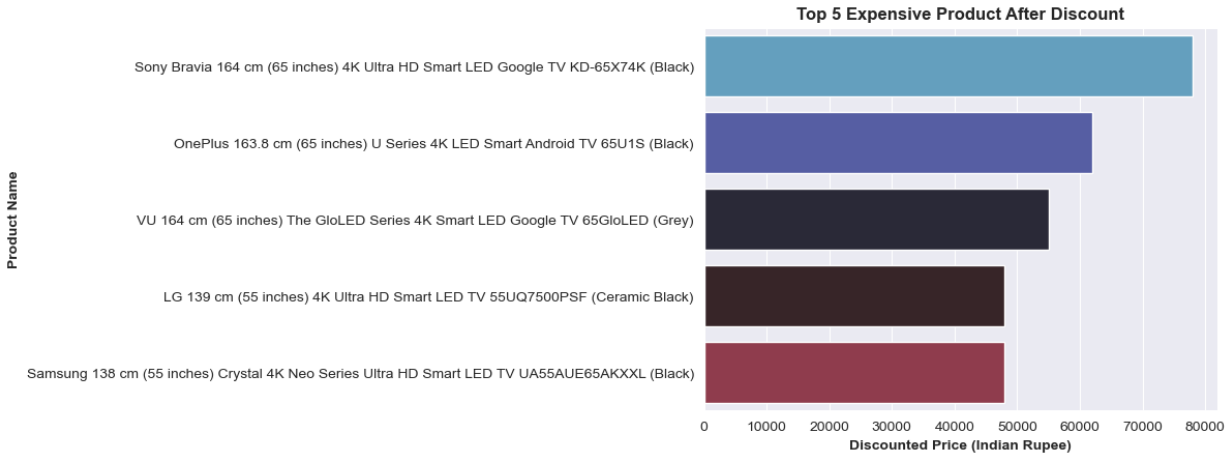
In [57]:
```
#Most Amountof Product by category
most_main_items = df1['category_1'].value_counts().head(5).rename_axis('category_1').r

most_sub_items = df1['category_2'].value_counts().head(10).rename_axis('category_2').r
```

```
fig, ax = plt.subplots(2, 1, figsize=(8, 10))
fig.suptitle('Most Amount of Products by Category', fontweight='heavy', size='x-large'

sns.barplot(ax=ax[0], data=most_main_items, x='counts', y='category_1')
sns.barplot(ax=ax[1], data=most_sub_items, x='counts', y='category_2')

plt.subplots_adjust(hspace = 0.3)

ax[0].set_xlabel('Count', fontweight='bold')
ax[0].set_ylabel('Product Main Category', fontweight='bold')

ax[1].set_xlabel('Count', fontweight='bold')
ax[1].set_ylabel('Product Sub-Category', fontweight='bold')

ax[0].set_title('Most Products by Main Category', fontweight='bold')
ax[1].set_title('Most Products by Sub-Category', fontweight='bold')


ax[0].bar_label(ax[0].containers[0])
ax[1].bar_label(ax[1].containers[0])

plt.show()
```



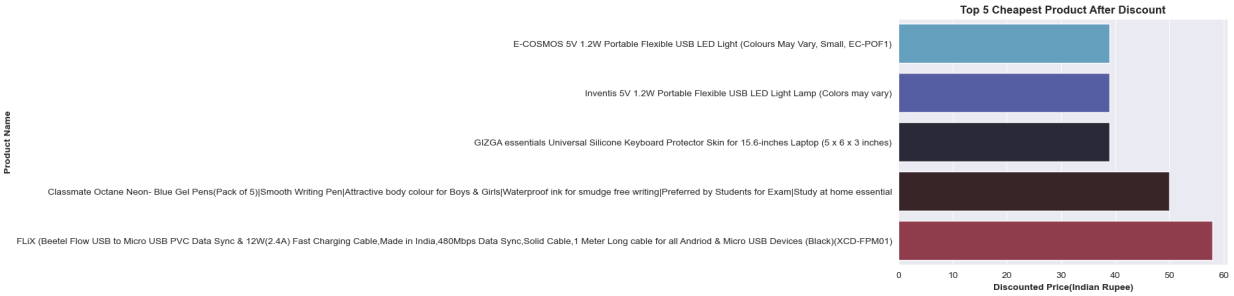
Electronics especially accessories & pripherals and Kitchen & homeappliance contain most of the products in this data set . In general most products are related to the electric devices in this dataset.

In [90]:
```
#Top 5 Most Expensive Products After Discount
disc_exp=sns.barplot(data=df1.sort_values('discounted_price',ascending=False).head(5),
disc_exp.set_title('Top 5 Expensive Product After Discount',fontweight='bold')
disc_exp.set_xlabel('Discounted Price (Indian Rupee)',fontweight='bold')
disc_exp.set_ylabel('Product Name',fontweight='bold')
plt.show()
```



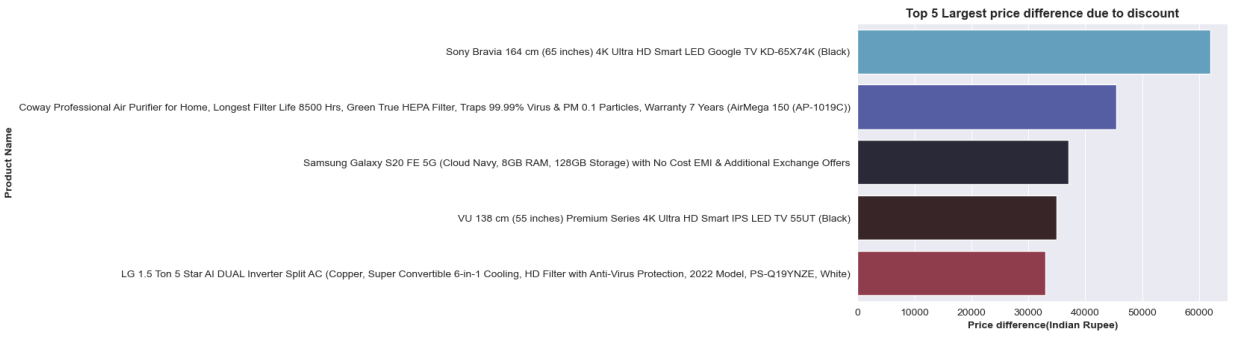
Sony Bravia 164 cm (65 inches) is the most expensive product after discount

In [91]:
```
#Top 5 cheapest Products After Discount
disc_cheap=sns.barplot(data=df1.sort_values('discounted_price').head(5),x='discounted_
disc_cheap.set_title('Top 5 Cheapest Product After Discount',fontweight='bold')
disc_cheap.set_xlabel('Discounted Price(Indian Rupee)',fontweight='bold')
disc_cheap.set_ylabel('Product Name',fontweight='bold')
plt.show()
```



E-cosmos 5V 1.2W Portale Flexible is the cheapest product after discount

In [73]:
```
#Top 5 Largest price difference due to the discount in products
price_diff=sns.barplot(data=df1.sort_values('difference_price',ascending=False).head(5
price_diff.set_title('Top 5 Largest price difference due to discount',fontweight='bold
price_diff.set_xlabel('Price difference(Indian Rupee)',fontweight='bold')
price_diff.set_ylabel('Product Name',fontweight='bold')
plt.show()
```

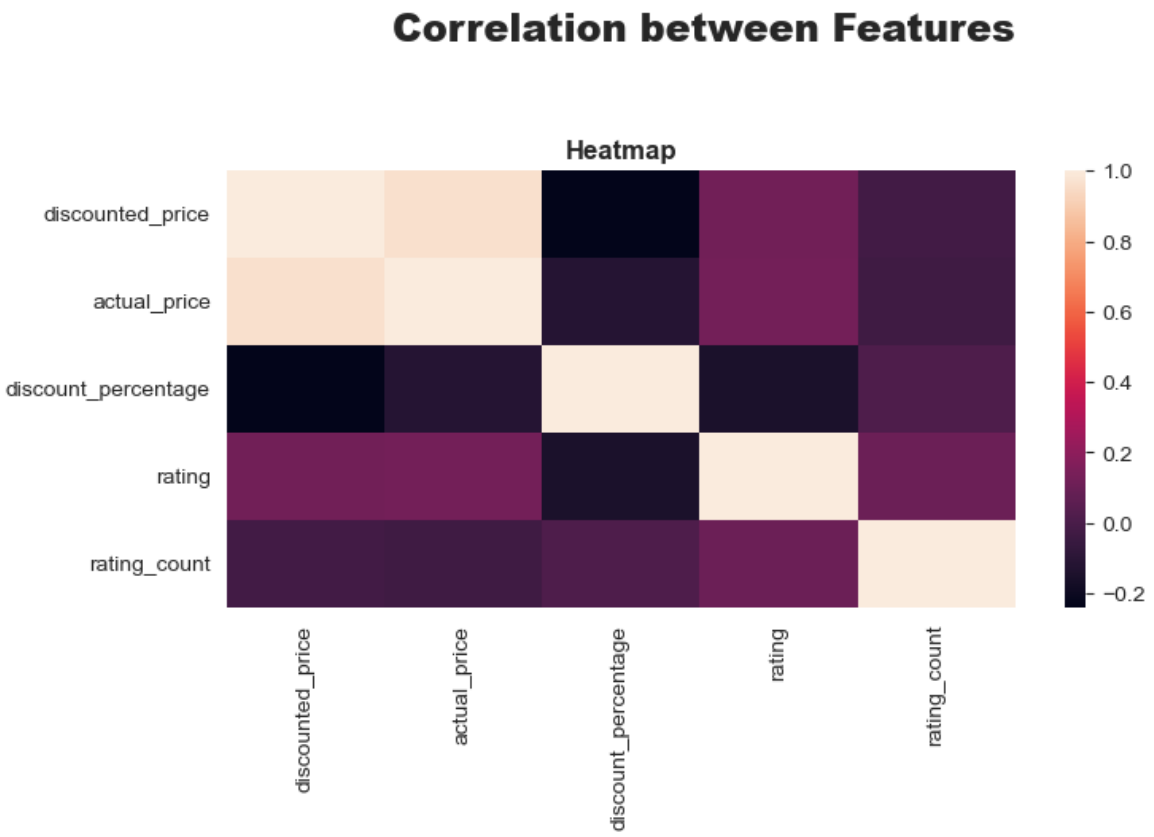


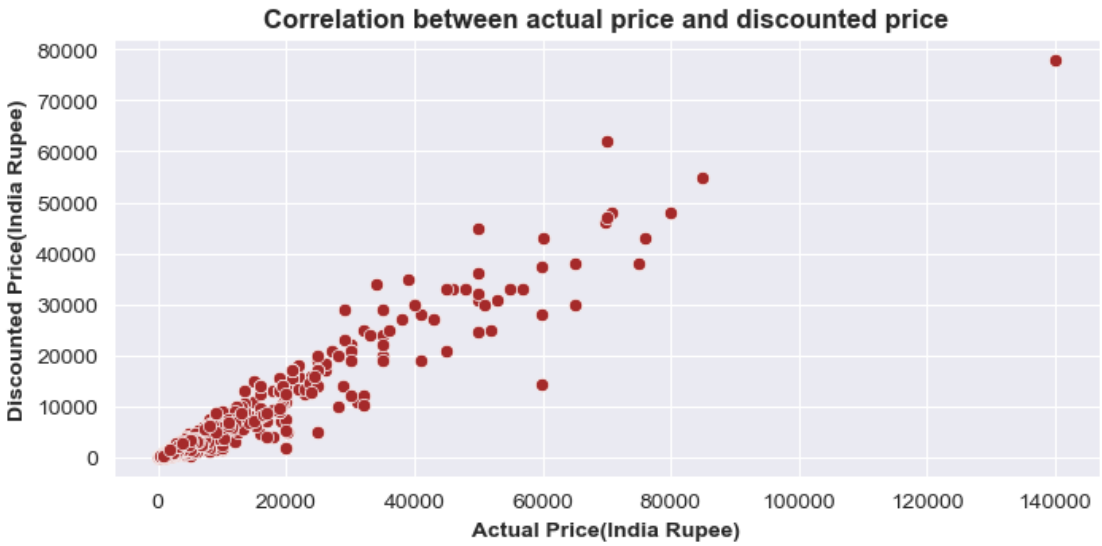
Sony Bravia 164cm having the largest price difference due to discount

Observation 2 : Correlation Between Features

In [92]:
```
#heatmap and Correlation Between Features
fig, ax = plt.subplots(2, 1, figsize=(8, 10))
fig.suptitle('Correlation between Features',fontweight='heavy',size='xx-large')
sns.heatmap(ax=ax[0],data=df1.corr())
sns.scatterplot(ax=ax[1],data=df1,y='discounted_price',x='actual_price',color='brown')
plt.subplots_adjust(hspace=0.8)
ax[1].set_xlabel('Actual Price(India Rupee)',fontweight='bold')
ax[1].set_ylabel('Discounted Price(India Rupee)',fontweight='bold')
ax[0].set_title('Heatmap',fontweight='bold')
ax[1].set_title('Correlation between actual price and discounted price',fontweight='bo
plt.show()
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_44336\1110266529.py:4: FutureWarning: Th
e 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_o
nly to silence this warning.
  sns.heatmap(ax=ax[0],data=df1.corr())
```



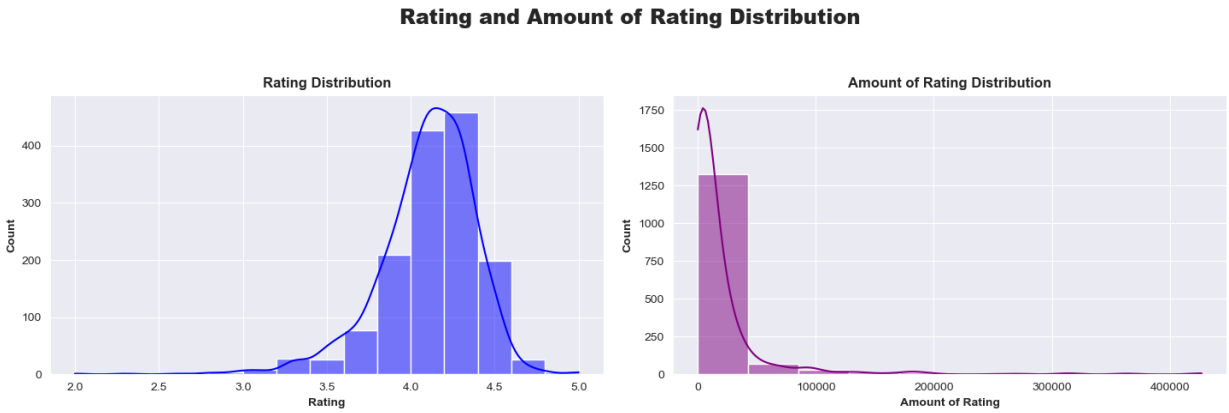


There are almost no correlation between the dataset but there is positive correlation between the discounted price of product and actual price of product.

Observation 3 : Product Rating

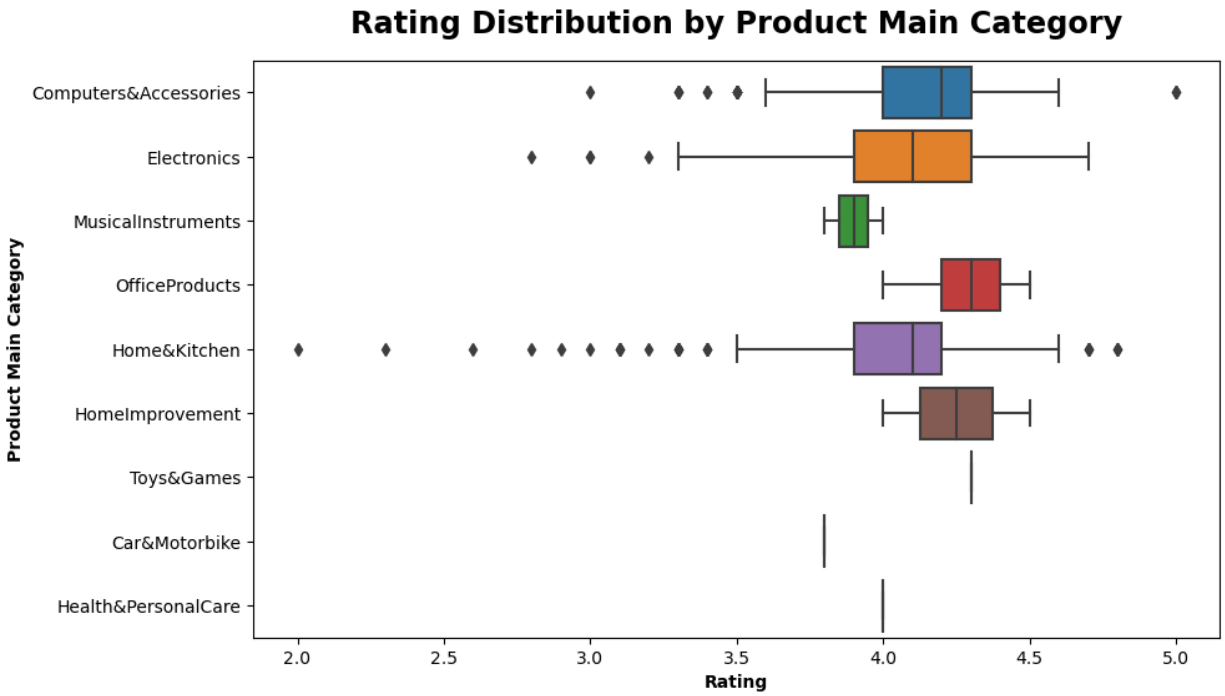
In [98]:
```
# Rating and Amount of rating distribution
fig, ax = plt.subplots(1, 2, figsize=(15, 5))
fig.suptitle('Rating and Amount of Rating Distribution',fontweight='heavy',size='xx-la
fig.tight_layout(pad=3.0)
sns.histplot(ax=ax[0],data=df1,x='rating',bins=15,kde=True,color='blue')
```

```
sns.histplot(ax=ax[1],data=df1,x='rating_count',bins=10,kde=True,color='purple')
ax[0].set_xlabel('Rating',fontweight='bold')
ax[1].set_xlabel('Amount of Rating',fontweight='bold')
ax[0].set_ylabel('Count',fontweight='bold')
ax[1].set_ylabel('Count',fontweight='bold')
ax[0].set_title('Rating Distribution',fontweight='bold')
ax[1].set_title('Amount of Rating Distribution',fontweight='bold')
plt.show()
```



Most of the product range around 4.0 to 4.37 with no products under the score of 2.0. The Raating Distribution is Slightly left-Skewed. The amount of ratings given to a product is very widespread. Most of the products that have been rated, have around 0 - 5000 amount of rating for each product. Interestingly there are products that have more than 40,000 ratings. The amount of ratings distribution is highly right skewed.

In [47]:
```
#Rating Distribution by Product Main category
fig, ax = plt.subplots(figsize=(10, 6))
sns.boxplot(ax=ax, data=df1,x='rating',y='category_1')
ax.set_title('Rating Distribution by Product Main Category',fontweight='heavy',size='x
ax.set_xlabel('Rating',fontweight='bold')
ax.set_ylabel('Product Main Category',fontweight='bold')
plt.show()
```



Toys&Games,Car&Motorbike and health&PersonalCare product rating around 3.7 to 4.6. All homeImpprovement and officeProduct have the minimal rating of 4.0. Many of the Computer & Accessories, and Electronics products have ratings in the range of 3.6 - 4.6. Though these categories do have products that have a high rating such as 5.0 and low rating, going down to 2.75.

Noticeably, the Home & Kitchen products have a really widespread rating going to as high as 4.75 and going as low as 2.0 rating, which is the lowest rating out of all the products in this dataset. However, most of the products in this category fall in the range of around 3.8 - 4.6.

In [70]:
```
#Rating of Products Based on Rating category
rating_main_cat=df1.groupby(['category_1','rating_score']).agg('count').iloc[:,1].rena
rating_main_cat=rating_main_cat.rename(columns={'category_1':'Main category','rating_s
rating_main_cat
```

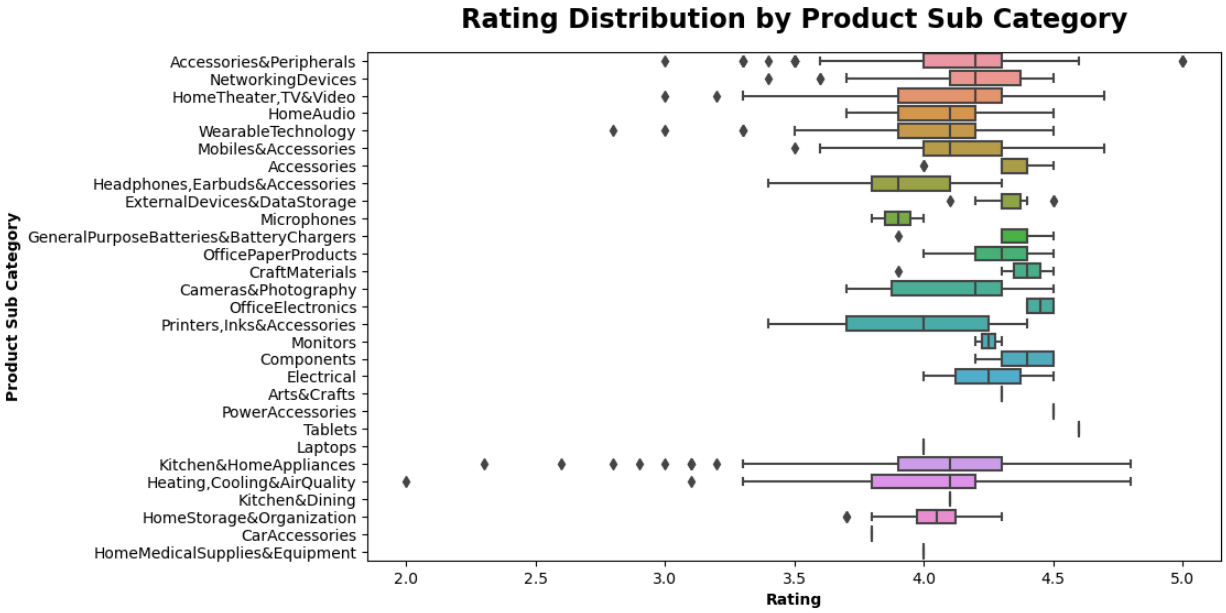
Out[70]:

	Main category	Rating Category	Amount
0	Car & Motorbike	Below Average	0
1	Car & Motorbike	Average	1
2	Car & Motorbike	Above Average	0
3	Car & Motorbike	Excellent	0
4	Computers & Accessories	Below Average	0
5	Computers & Accessories	Average	75
6	Computers & Accessories	Above Average	375
7	Computers & Accessories	Excellent	3
8	Electronics	Below Average	1
9	Electronics	Average	132
10	Electronics	Above Average	393
11	Electronics	Excellent	0
12	Health & PersonalCare	Below Average	0
13	Health & PersonalCare	Average	0
14	Health & PersonalCare	Above Average	1
15	Health & PersonalCare	Excellent	0
16	Home & Kitchen	Below Average	5
17	Home & Kitchen	Average	139
18	Home & Kitchen	Above Average	304
19	Home & Kitchen	Excellent	0
20	Home Improvement	Below Average	0
21	Home Improvement	Average	0
22	Home Improvement	Above Average	2
23	Home Improvement	Excellent	0
24	Musical Instruments	Below Average	0
25	Musical Instruments	Average	1
26	Musical Instruments	Above Average	1
27	Musical Instruments	Excellent	0
28	Office Products	Below Average	0
29	Office Products	Average	0
30	Office Products	Above Average	31
31	Office Products	Excellent	0
32	Toys & Games	Below Average	0
33	Toys & Games	Average	0

	Main category	Rating Category	Amount
34	Toys & Games	Above Average	1
35	Toys & Games	Excellent	0

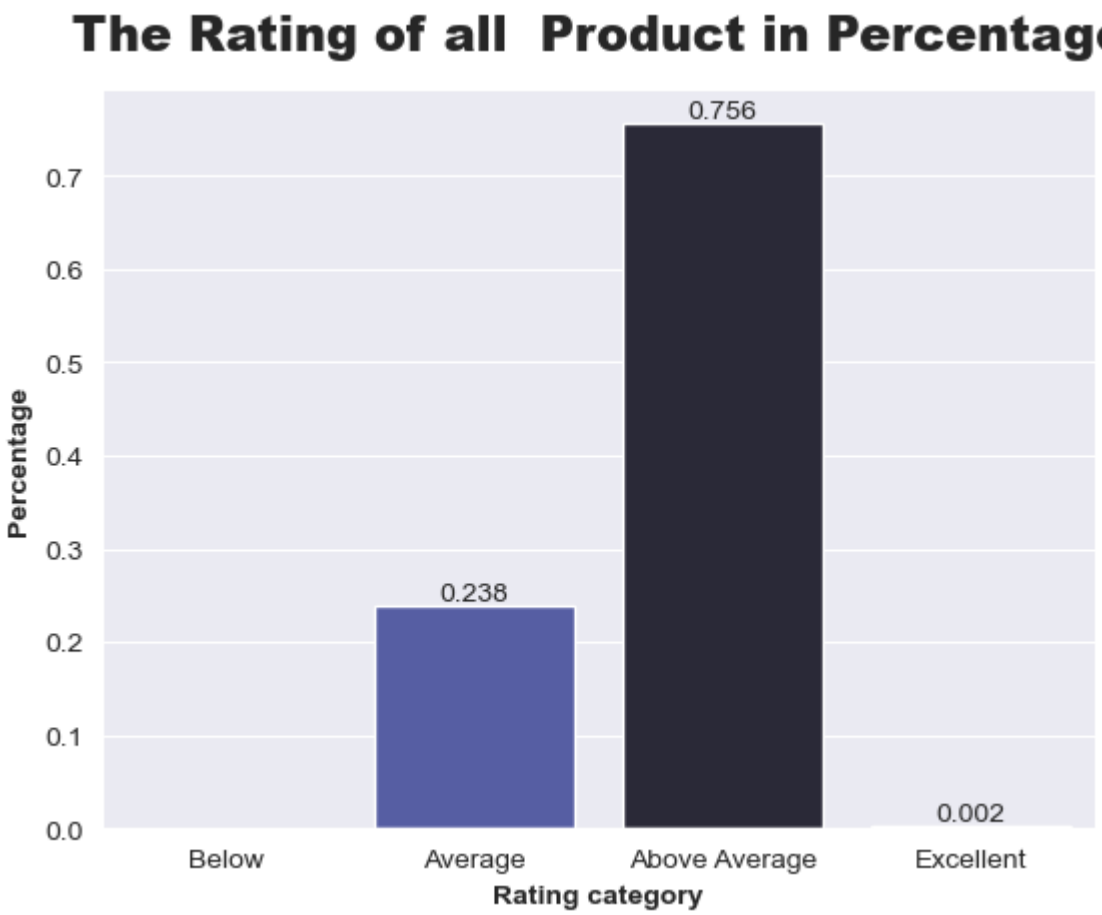
This list mention about the product and product Main category and amount of rating

In [71]:
```
#Rating Distribution by Product Sub-Category
fig, ax = plt.subplots(figsize=(10, 6))
sns.boxplot(ax=ax, data=df1,x='rating',y='category_2')
ax.set_title('Rating Distribution by Product Sub Category',fontweight='heavy',size='x
ax.set_xlabel('Rating',fontweight='bold')
ax.set_ylabel('Product Sub Category',fontweight='bold')
plt.show()
```



What i observed in this Graph of Rating Distribution by Product Sub Category is that Accessories & Peripherals is highly rated product .The Lowest rated product came from the sub category of heating,cooling & air quality.

In [46]:
```
#The Rating of all Product in Percentage
rating_ordered=['Below','Average','Above Average','Excellent']
rating_count=df1['rating_score'].value_counts(normalize=True).rename_axis('rating').re
rating_count['counts']=rating_count['counts'].round(3)
rating_count_plot=sns.barplot(data=rating_count ,x='rating',y='counts',order=rating_or
rating_count_plot.set_xlabel('Rating category',fontweight='bold')
rating_count_plot.set_ylabel('Percentage',fontweight='bold')
rating_count_plot.set_title('The Rating of all  Product in Percentage',fontweight='hea
rating_count_plot.bar_label(rating_count_plot.containers[0])

plt.show()
```



Most of the product in the dataset have been rated Above average.There are extremely few products are rated below Average and Excellent.No Products are rated poor in this dataset

In [47]:
```
#Pivoting the Rating table
def p25(g):
    return np.percentile(g,25)
def p75(g):
    return np.percentile(g,75)
rating_pivot=df1.pivot_table(values=['rating','rating_count'],index=['category_1','cat
                             aggfunc=([p25,np.median,np.mean,p75]))
rating_pivot=rating_pivot.rename(columns={'rating':'Rating','rating_count':'Rating_cou
                                 index={'category_1':'Main_category','category_2':'Sub_
rating_pivot
```

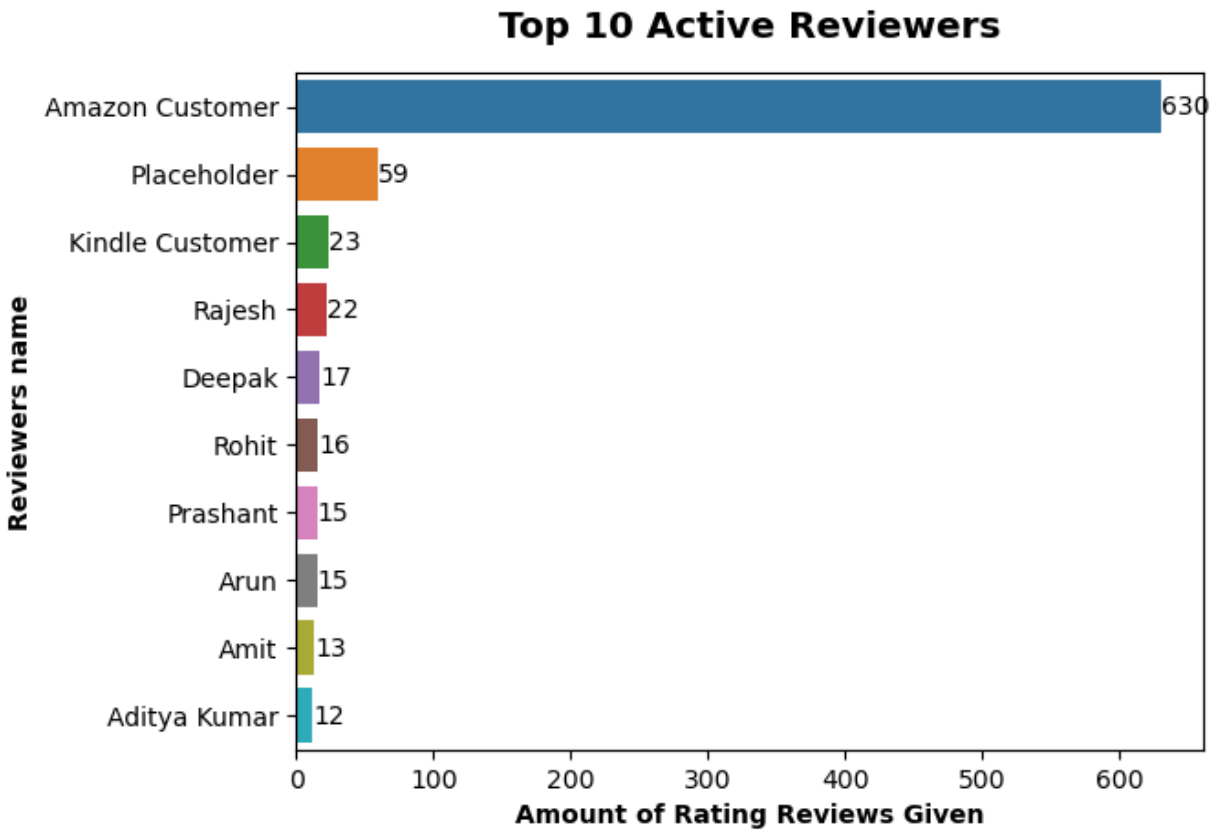
Out[47]:

		p25		Median	
		Rating	Rating_count	Rating	Rating_count
category_1	category_2				
Car & Motorbike	CarAccessories	3.800	1118.00	3.80	1118.0
Computers & Accessories	Accessories&Peripherals	4.000	1396.00	4.20	6736.0
	Components	4.300	2515.00	4.40	3029.0
	ExternalDevices&DataStorage	4.300	19747.50	4.30	45835.5
	Laptops	4.000	323.00	4.00	323.0
	Monitors	4.225	2166.25	4.25	2318.5
	NetworkingDevices	4.100	10281.75	4.20	18262.0
	Printers,Inks&Accessories	3.700	3435.50	4.00	4567.0
	Tablets	4.600	2886.00	4.60	2886.0
Electronics	Accessories	4.300	67259.00	4.40	67260.0
	Cameras&Photography	3.875	5384.25	4.20	11865.5
	GeneralPurposeBatteries&BatteryChargers	4.300	1269.50	4.40	12829.0
	Headphones,Earbuds&Accessories	3.800	9881.75	3.90	40296.0
	HomeAudio	3.900	2625.75	4.10	8746.5
	HomeTheater,TV&Video	3.900	426.50	4.20	1611.0
	Mobiles&Accessories	4.000	3197.00	4.10	13246.0
	PowerAccessories	4.500	20668.00	4.50	20668.0
	WearableTechnology	3.900	5683.75	4.10	17832.0
Health & PersonalCare	HomeMedicalSupplies&Equipment	4.000	3663.00	4.00	3663.0
Home & Kitchen	CraftMaterials	4.350	6542.50	4.40	9427.0
	Heating,Cooling&AirQuality	3.800	248.25	4.10	1743.5
	HomeStorage&Organization	3.975	870.75	4.05	2366.5
	Kitchen&Dining	4.100	270563.00	4.10	270563.0
	Kitchen&HomeAppliances	3.900	626.25	4.10	2305.5
Home Improvement	Electrical	4.125	3432.00	4.25	4283.0
Musical Instruments	Microphones	3.850	32329.50	3.90	44441.0
Office Products	OfficeElectronics	4.400	5426.50	4.45	7185.0
	OfficePaperProducts	4.200	2560.50	4.30	3785.0
Toys & Games	Arts&Crafts	4.300	15867.00	4.30	15867.0

This is the specific data on Rating and Amount of the rating for each main and sub-category of Product from the dataset.

Observation 3 : Reviewers

In [69]:
```
#Reviewers who gave rating and reviews for more than one product
top_reviewer=data=df2['user_name'].value_counts().head(10).rename_axis('username').res
top_review_plot=sns.barplot(data=top_reviewer,x='counts',y='username')
top_review_plot.bar_label(top_review_plot.containers[0])

top_review_plot.set_xlabel('Amount of Rating Reviews Given',fontweight='bold')
top_review_plot.set_ylabel('Reviewers name',fontweight='bold')
top_review_plot.set_title('Top 10 Active Reviewers',fontweight='heavy',size='x-large',
plt.show()
```

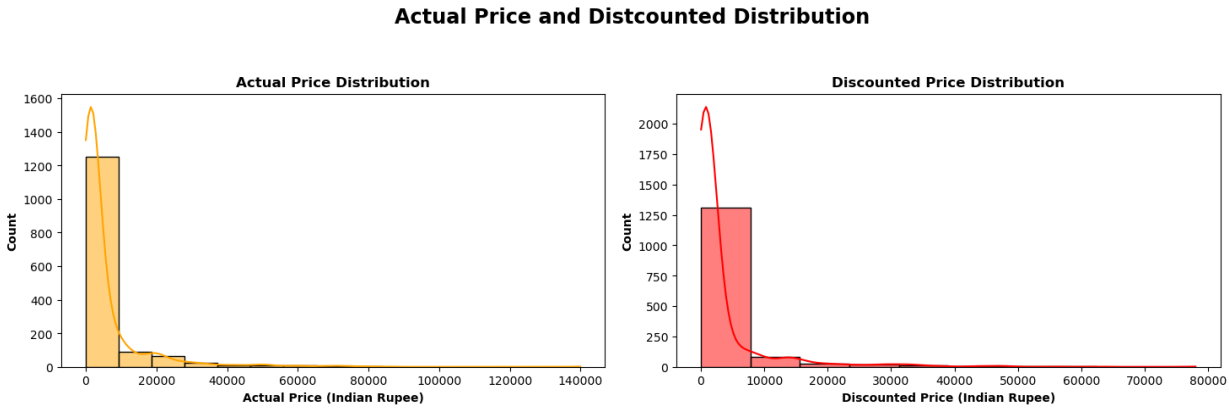


There are more than 500 active reviewers who review the product ananomously under the alias of Amazon customer,Placeholder, kindle customer There are more than 8 people who have given ratings and reviews to more than 10 products on this dataset.

Observation 3 : Product Pricing

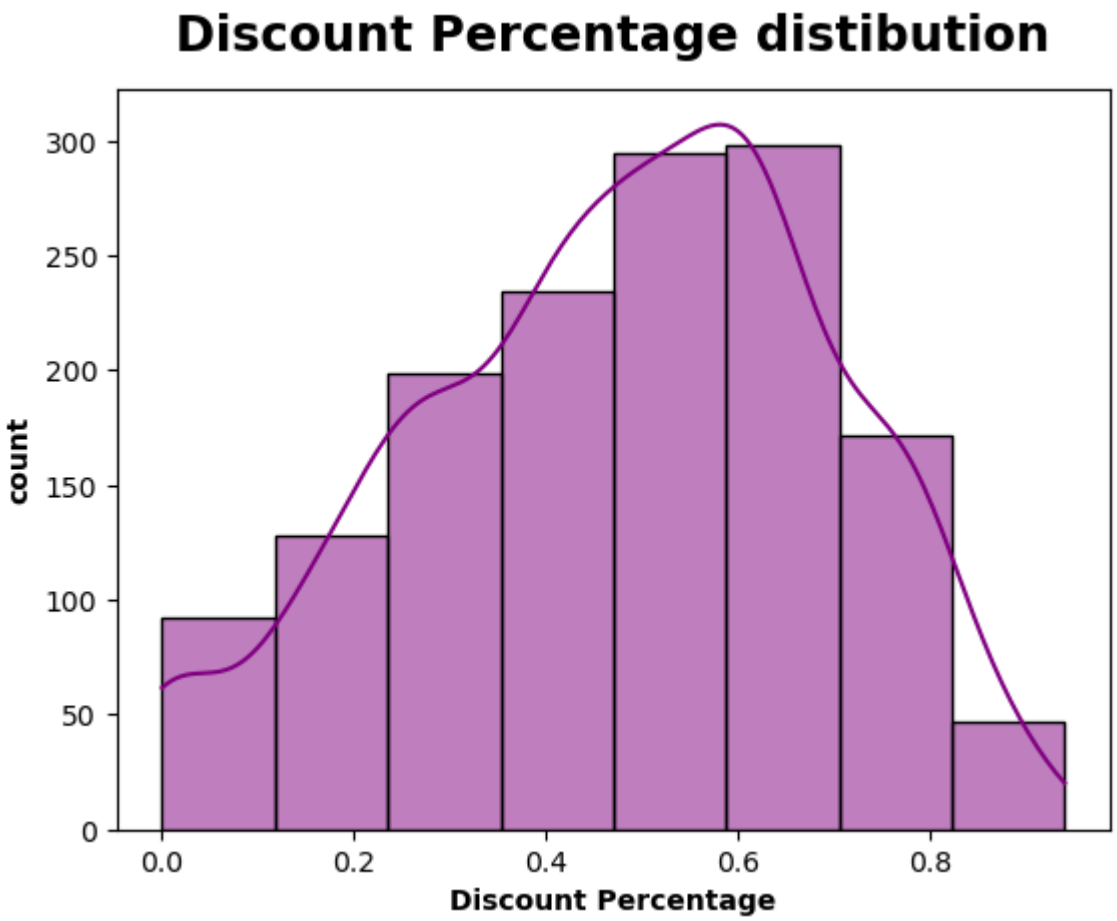
In [72]:
```
#Actual price and discounted Price distribution
fig, ax = plt.subplots(1, 2, figsize=(15, 5))
fig.suptitle('Actual Price and Distcounted Distribution',fontweight='heavy',size='xx-l
fig.tight_layout(pad=3.0)
```

```
sns.histplot(ax=ax[0],data=df1,x='actual_price',bins=15,kde=True,color='orange')
sns.histplot(ax=ax[1],data=df1,x='discounted_price',bins=10,kde=True,color='red')
ax[0].set_xlabel('Actual Price (Indian Rupee)',fontweight='bold')
ax[1].set_xlabel('Discounted Price (Indian Rupee)',fontweight='bold')
ax[0].set_ylabel('Count',fontweight='bold')
ax[1].set_ylabel('Count',fontweight='bold')
ax[0].set_title('Actual Price Distribution',fontweight='bold')
ax[1].set_title('Discounted Price Distribution',fontweight='bold')
plt.show()
```



Both of the Graph shows the same results which is positive Skewed to right.

In [75]:
```
#Discount Percentage distribution

Disc_per=sns.histplot(data=df1 ,x='discount_percentage',bins=8,kde=True,color='purple'
Disc_per.set_xlabel('Discount Percentage',fontweight='bold')
Disc_per.set_ylabel('count',fontweight='bold')
Disc_per.set_title('Discount Percentage distibution',fontweight='heavy',size='xx-large


plt.show()
```

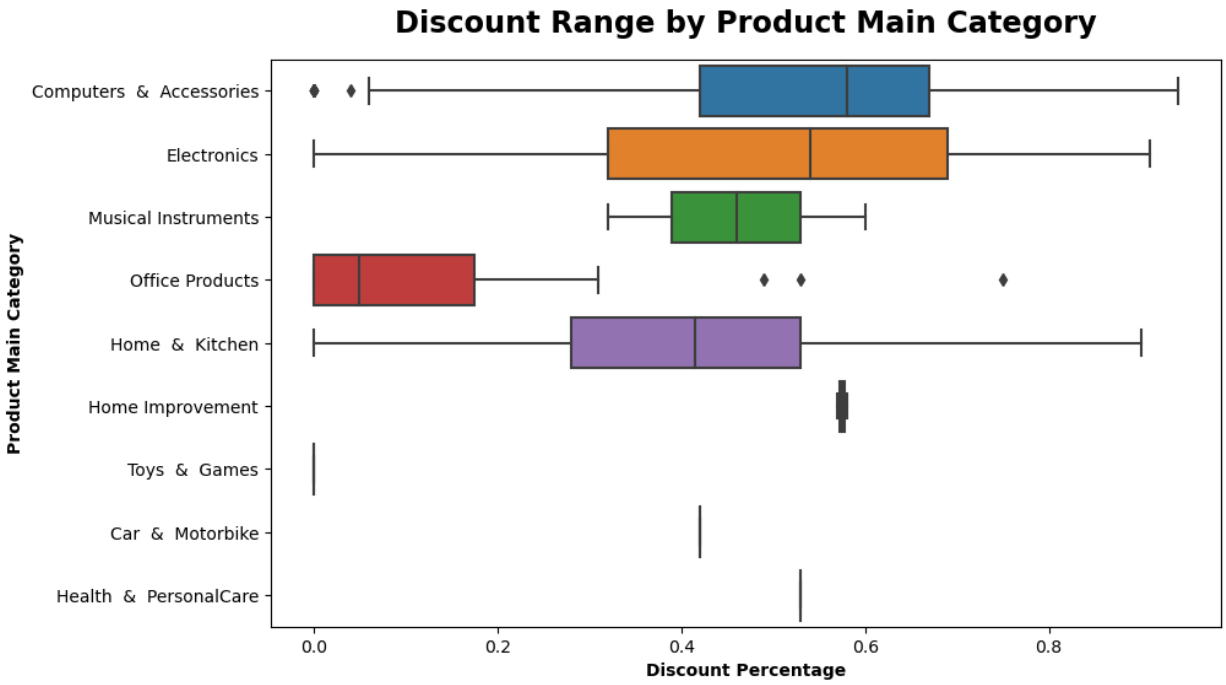


Most of the Product have the discount of more than 50% to 80%.

In [76]:
```
#Specfic details of discount percentage
df1['discount_percentage'].describe()
```

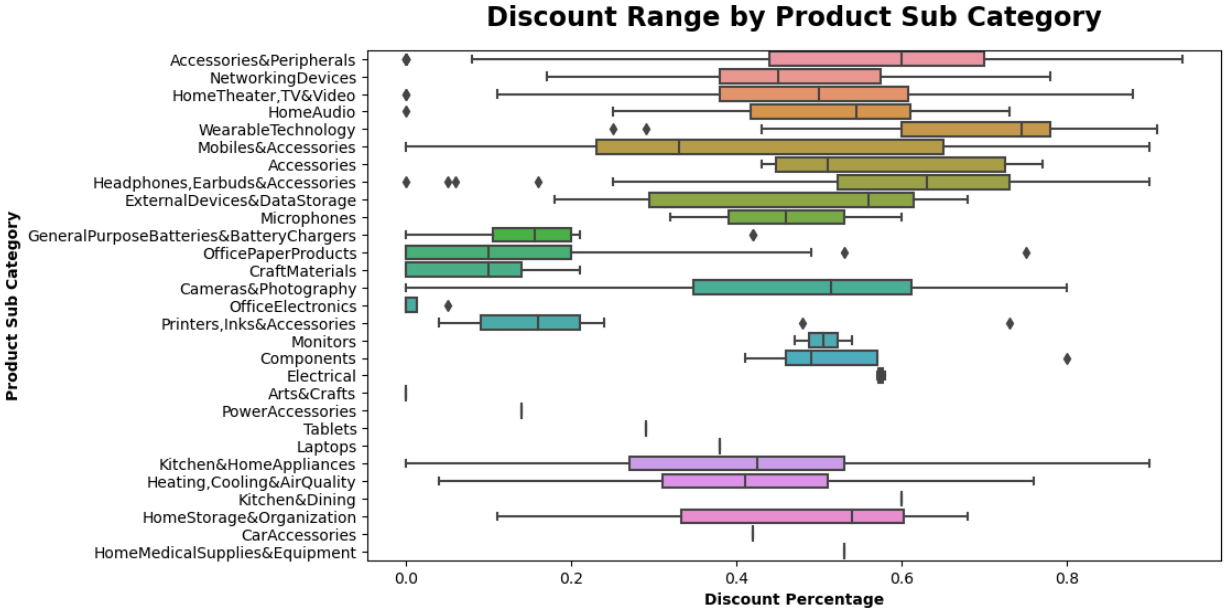
Out[76]:
```
count    1465.000000
mean        0.476915
std         0.216359
min         0.000000
25%         0.320000
50%         0.500000
75%         0.630000
max         0.940000
Name: discount_percentage, dtype: float64
```

In [84]:
```
#The Discount range by Product Main Category
fig, ax = plt.subplots(figsize=(10, 6))
sns.boxplot(data=df1,x='discount_percentage',y='category_1')
ax.set_title('Discount Range by Product Main Category',fontweight='heavy',size='xx-lar
ax.set_xlabel('Discount Percentage',fontweight='bold')
ax.set_ylabel('Product Main Category',fontweight='bold')
plt.show()
```



Computers & Accessoies,Electronics,Home & Kitchen have a large widely spread discount ranging from mininal 10% to 90%. Toys & game,Car & Motorbike,Health & PersonalCare,Home Improvement are the least spread discount. office product does not give a large amount of discount as compared to product main category.

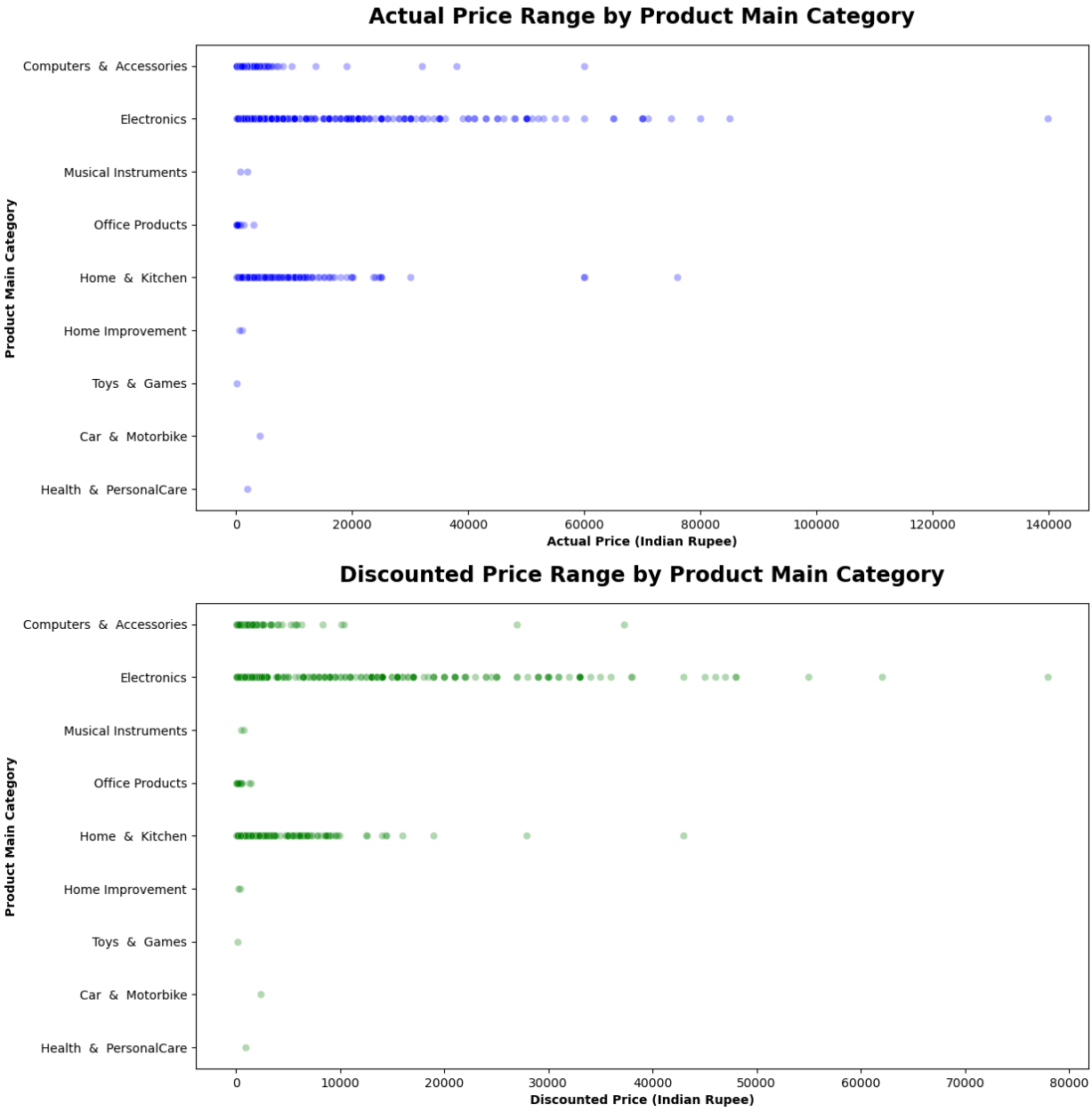
In [85]:
```
#The Discount range by Product Sub Category
fig, ax = plt.subplots(figsize=(10, 6))
sns.boxplot(data=df1,x='discount_percentage',y='category_2')
ax.set_title('Discount Range by Product Sub Category',fontweight='heavy',size='xx-larg
ax.set_xlabel('Discount Percentage',fontweight='bold')
ax.set_ylabel('Product Sub Category',fontweight='bold')
plt.show()
```



In [93]:
```
#Actual Price range and discounted Price range  by product Main Category
fig, ax = plt.subplots(2,1,figsize=(13, 15))
fig.suptitle('Price Range by Product Main category',fontweight='heavy',size='xx-large'
```

```
sns.scatterplot(ax=ax[0],data=df1,x='actual_price',y='category_1',alpha=0.3,color='blu
sns.scatterplot(ax=ax[1],data=df1,x='discounted_price',y='category_1',alpha=0.3,color=

ax[0].set_title('Actual Price Range by Product Main Category',fontweight='heavy',size=
ax[0].set_xlabel('Actual Price (Indian Rupee)',fontweight='bold')
ax[0].set_ylabel('Product Main Category',fontweight='bold')

ax[1].set_title('Discounted Price Range by Product Main Category',fontweight='heavy',s
ax[1].set_xlabel('Discounted Price (Indian Rupee)',fontweight='bold')
ax[1].set_ylabel('Product Main Category',fontweight='bold')
plt.show()
```

Price Range by Product Main category



There is the decrease in the product category of electronic after applying Discount . Most of the product's actual price falls below 20,000 Rupee. For the discounted price, most of the products fall under 10,000 Rupee.

In [94]:
```
#Actual Price range and discounted Price range  by product Sub Category
fig, ax = plt.subplots(2,1,figsize=(13, 15))
fig.suptitle('Price Range by Product Sub category',fontweight='heavy',size='xx-large')
sns.scatterplot(ax=ax[0],data=df1,x='actual_price',y='category_2',alpha=0.3,color='red
sns.scatterplot(ax=ax[1],data=df1,x='discounted_price',y='category_2',alpha=0.3,color=

ax[0].set_title('Actual Price Range by Product Sub Category',fontweight='heavy',size='
ax[0].set_xlabel('Actual Price (Indian Rupee)',fontweight='bold')
ax[0].set_ylabel('Product Sub Category',fontweight='bold')

ax[1].set_title('Discounted Price Range by Product Sub Category',fontweight='heavy',si
ax[1].set_xlabel('Discounted Price (Indian Rupee)',fontweight='bold')
ax[1].set_ylabel('Product Sub Category',fontweight='bold')
plt.show()
```

In [95]:
```
#Pivoting the Price
def p25(g):
    return np.percentile(g,25)
def p75(g):
    return np.percentile(g,75)
```

```
Price_pivot=df1.pivot_table(values=['actual_price','discounted_price'],index=['categor
                            aggfunc=([p25,np.median,np.mean,p75]))

Price_pivot
```

Out[95]:

		p25			
		actual_price	discounted_price	actual_price	d
category_1	category_2				
Car & Motorbike	CarAccessories	4000.00	2339.00	4000.0	
Computers & Accessories	Accessories&Peripherals	499.00	199.00	999.0	
	Components	3100.00	1709.00	3500.0	
	ExternalDevices&DataStorage	1074.25	504.00	1575.0	
	Laptops	59890.00	37247.00	59890.0	
	Monitors	15090.00	7249.00	16430.0	
	NetworkingDevices	1208.00	530.00	1949.0	
	Printers,Inks&Accessories	811.00	597.00	1999.0	
	Tablets	37999.00	26999.00	37999.0	
Electronics	Accessories	1150.00	479.00	1800.0	
	Cameras&Photography	946.00	386.50	1999.0	
	GeneralPurposeBatteries&BatteryChargers	205.00	166.75	282.5	
	Headphones,Earbuds&Accessories	999.00	450.50	1994.5	
	HomeAudio	1274.00	736.50	2394.5	
	HomeTheater,TV&Video	824.00	349.00	2749.0	
	Mobiles&Accessories	1299.00	399.00	2999.0	
	PowerAccessories	1499.00	1289.00	1499.0	
	WearableTechnology	5999.00	1599.00	7990.0	
Health & PersonalCare	HomeMedicalSupplies&Equipment	1900.00	899.00	1900.0	
Home & Kitchen	CraftMaterials	132.50	114.50	225.0	
	Heating,Cooling&AirQuality	1990.00	1049.00	3062.5	
	HomeStorage&Organization	374.00	199.00	649.0	
	Kitchen&Dining	495.00	199.00	495.0	
	Kitchen&HomeAppliances	1000.00	596.00	1962.5	
Home Improvement	Electrical	699.00	293.00	799.0	
Musical Instruments	Microphones	1023.00	558.00	1347.0	
Office Products	OfficeElectronics	511.25	501.50	542.5	
	OfficePaperProducts	120.00	107.00	175.0	
Toys & Games	Arts&Crafts	150.00	150.00	150.0	