Amazon Sales Data

Description:

This dataset contains information on #K+ Amazon products, including their ratings, reviews, and other details.

In [1]:
```
# for remove warning
import warnings
warnings.filterwarnings("ignore")
```

Libraries were used for this EDA analysyis

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

In [3]:
```
df= pd.read_csv("amazon.csv")
df.head(5)
```

Out[3]:

	product_id	product_name	category	discounted_price	actual_price	discount_percentage	rating	rating_count	about_product	
0	B07JW9H4J1	Wayona Nylon Braided USB to Lightning Fast Cha...	Computers&Accessories|Accessories&Peripherals|...	₹399	₹1,099	64%	4.2	24,269	High Compatibility : Compatible With iPhone 12...	AG3D6O4ST
1	B098NS6PVG	Ambrane Unbreakable 60W / 3A Fast Charging 1.5...	Computers&Accessories|Accessories&Peripherals|...	₹199	₹349	43%	4.0	43,994	Compatible with all Type C enabled devices, be...	AECPFYFC
2	B096MSW6CT	Sounce Fast Phone Charging Cable & Data Sync U...	Computers&Accessories|Accessories&Peripherals|...	₹199	₹1,899	90%	3.9	7,928	【 Fast Charger& Data Sync】-With built-in safet...	AGU3BBQ2V
3	B08HDJ86NZ	boAt Deuce USB 300 2 in 1 Type-C & Micro USB S...	Computers&Accessories|Accessories&Peripherals|...	₹329	₹699	53%	4.2	94,363	The boAt Deuce USB 300 2 in 1 cable is compati...	AEWAZDZZ
4	B08CF3B7N1	Portronics Konnect L 1.2M Fast Charging 3A 8 P...	Computers&Accessories|Accessories&Peripherals|...	₹154	₹399	61%	4.2	16,905	[CHARGE & SYNC FUNCTION]- This cable comes wit...	AE3Q6KSU

In [4]:
```
df.columns
```

Out[4]:
```
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')
```

The data set we are working on amazon sales data where columns describes are product_id: Unique identifier for each product

- product_name: Name of the product
- category: Category of the product
- discounted_price: Discounted price of the product
- actual_price: Actual price of the product
- discount_percentage: Percentage of discount for the product
- rating: Rating of the product (#-5)
- rating_count: Number of people who voted for the Amazon rating
- about_product: Description about the product
- user_id: ID of the user who wrote the review
- user_name: Name of the user who wrote the review
- review_id: ID of the user review
- review_title: Short review
- review_content: Long review
- img_link: Image link of the product
- product_link: Qfficial website link of the product

Before analyzing the data lets check if there are any null or missing values in our dataset

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
product_id            0
product_name          0
category              0
discounted_price      0
actual_price          0
discount_percentage   0
rating                0
rating_count          2
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
```

Replace Null Value and change data types

In [6]:
```
df['actual_price'] = df['actual_price'].str.replace('₹','')
df['actual_price'] = df['actual_price'].str.replace(',','').astype('float64')

df['discounted_price'] = df['discounted_price'].str.replace('₹','')
df['discounted_price'] = df['discounted_price'].str.replace(',','').astype('float64')
df['discount_percentage'] = df['discount_percentage'].str.replace('%','').astype('float64')
df['discount_percentage'] = df['discount_percentage']/100
```

```
df['rating_count'] = df['rating_count'].str.replace(',','').astype('float64')
df['rating'].value_counts()
```

Out[6]:
```
rating
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: count, dtype: int64
```

In [7]:
```
df['rating'].value_counts()
```

Out[7]:
```
rating
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: count, dtype: int64
```

In [8]:
```
# To check the rating colume where rating == |
df[df['rating']=='|']
```

Out[8]:

	product_id	product_name	category	discounted_price	actual_price	discount_percentage	rating	rating_count	about_product	
1279	B08L12N5H1	Eureka Forbes car Vac 100 Watts Powerful Sucti...	Home&Kitchen|Kitchen&HomeAppliances|Vacuum,Cle...	2099.0	2499.0	0.16	|	992.0	No Installation is provided for this product|1...	AGTD

In [9]:
```
# Replace '|' with '4.0' in the 'rating' column
df['rating'] = df['rating'].str.replace('|','4.0').astype('float64')
df['rating'].value_counts()
```

Out[9]:
```
rating
4.1    244
4.3    230
4.2    228
4.0    182
3.9    123
4.4    123
3.8     86
4.5     75
3.7     42
3.6     35
3.5     26
4.6     17
3.3     16
3.4     10
4.7      6
3.0      4
3.1      4
5.0      3
4.8      3
3.2      2
2.8      2
2.3      1
2.0      1
2.6      1
2.9      1
Name: count, dtype: int64
```

In [10]:
```
len(df['rating'])
```

Out[10]:
```
1465
```

In [11]:
```
df.dtypes
```

Out[11]:
```
product_id              object
product_name            object
category                object
discounted_price       float64
actual_price           float64
discount_percentage    float64
rating                 float64
rating_count           float64
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 [12]:
```
#check for num of rows and columns
df.shape
```

Out[12]:
```
(1465, 16)
```

In [13]:
```
#Create new data frame with selected columns
#df1 = df[['product_id', 'product_name', 'category', 'discounted_price', 'actual_price', 'discount_percentage', 'rating', 'rating_count']].copy()
```

Split category column

In [60]:
```
categorial_split = df['category'].str.split('|', expand=True)
categorial_split.isnull().sum()
```

Out[60]:
```
0       0
1       0
2       8
3     165
4     943
5    1380
6    1452
dtype: int64
```

Rename column

In [61]:
```
categorial_split = categorial_split.rename(columns={0:'Category', 1:'Sub category'})
```

Add new cols to data frame and drop the old ones

In [62]:
```
df['Category'] = categorial_split['Category']
df['Sub category'] = categorial_split['Sub category']
df.drop(columns ='category', inplace=True)
df
```

Out[62]:

	product_id	product_name	discounted_price	actual_price	discount_percentage	rating	rating_count	about_product	user
0	B07JW9H4J1	Wayona Nylon Braided USB to Lightning Fast Cha...	399.0	1099.0	63.694268	4.2	24269.0	High Compatibility : Compatible With iPhone 12...	AG3D6O4STAQKAY2UVGEUV46KN35Q,AHMY5CWJMMK5BJRB
1	B098NS6PVG	Ambrane Unbreakable 60W / 3A Fast Charging 1.5...	199.0	349.0	42.979943	4.0	43994.0	Compatible with all Type C enabled devices, be...	AECPFYFQVRUWC3KGNLJIOREFP5LQ,AGYYVPDD7YG7FYNB
2	B096MSW6CT	Sounce Fast Phone Charging Cable & Data Sync U...	199.0	1899.0	89.520800	3.9	7928.0	【 Fast Charger& Data Sync】-With built-in safet...	AGU3BBQ2V2DDAMOAKGFAWDDQ6QHA,AESFLDV2PT363T2A(
3	B08HDJ86NZ	boAt Deuce USB 300 2 in 1 Type-C & Micro USB S...	329.0	699.0	52.932761	4.2	94363.0	The boAt Deuce USB 300 2 in 1 cable is compati...	AEWAZDZZJLQUYVOVGBEUKSLXHQ5A,AG5HTSFRRE6NL3M5
4	B08CF3B7N1	Portronics Konnect L 1.2M Fast Charging 3A 8 P...	154.0	399.0	61.403509	4.2	16905.0	[CHARGE & SYNC FUNCTION]- This cable comes wit...	AE3Q6KSUK5P75D5HFYHCRAOLODSA,AFUGIFH5ZAFXRDSZ
...	...	...	...	...	...	...	...	...	
1460	B08L7J3T31	Noir Aqua - 5pcs PP Spun Filter + 1 Spanner | ...	379.0	919.0	58.759521	4.0	1090.0	SUPREME QUALITY 90 GRAM 3 LAYER THIK PP SPUN F...	AHITFY6AHALOFOHOZEOC6XBP4FEA,AFRABBODZJZQB6Z4
1461	B01M6453MB	Prestige Delight PRWO Electric Rice Cooker (1 ...	2280.0	3045.0	25.123153	4.1	4118.0	230 Volts, 400 watts, 1 Year	AFG5FM3NEMOL6BNFRV2NK5FNJCHQ,AGEINTRN6Z563RML
1462	B009P2LIL4	Bajaj Majesty RX10 2000 Watts Heat Convector R...	2219.0	3080.0	27.954545	3.6	468.0	International design and styling|Two heat sett...	AGVPWCMAHYQWJOQKMUJN4DW3KM5Q,AF4Q3E66MY4SR7YQ
1463	B00J5DYCCA	Havells Ventil Air DSP 230mm Exhaust Fan (Pist...	1399.0	1890.0	25.978836	4.0	8031.0	Fan sweep area: 230 MM ; Noise level: (40 - 45...	AF2JQCLSCY3QJATWUNNHUSVUPNQQ,AFDMLUXC5LS5RXDJ
1464	B01486F4G6	Borosil Jumbo 1000-Watt Grill Sandwich Maker (...	2863.0	3690.0	22.411924	4.3	6987.0	Brand-Borosil, Specification â€“ 23V ~ 5Hz;1 W...	AFGW5PT3R6ZAVQR4Y5MWVAKBZAYA,AG7QNJ2SCS5VS5VY

1465 rows × 18 columns

In [63]:
```
#Fixing the strings in Main category
df['Category'].value_counts()
```

Out[63]:
```
Category
Electronics              526
Computers&Accessories    453
Home&Kitchen             448
OfficeProducts            31
MusicalInstruments         2
HomeImprovement            2
Toys&Games                 1
Car&Motorbike              1
Health&PersonalCare        1
Name: count, dtype: int64
```

In [64]:
```
df['Category'] = df['Category'].str.replace('&', ' & ')
df['Category'] = df['Category'].str.replace('OfficeProducts', 'Office Products')
df['Category'] = df['Category'].str.replace('MusicalInstruments', 'Musical Instruments')
df['Category'] = df['Category'].str.replace('HomeImprovement', 'Home Improvement')
```

Fixing string with Sub category

In [65]:
```
df['Sub category'].value_counts()
```

Out[65]:
```
Sub category
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: count, dtype: int64
```

In [66]:
```
df1['Sub category'] = df1['Sub category'].str.replace('&', ' & ')
df1['Sub category'] = df1['Sub category'].str.replace(',', ', ')
df1['Sub category'] = df1['Sub category'].str.replace('HomeAppliances', 'Home Appliances')
df1['Sub category'] = df1['Sub category'].str.replace('AirQuality', 'Air Quality')
df1['Sub category'] = df1['Sub category'].str.replace('WearableTechnology', 'Wearable Technology')
df1['Sub category'] = df1['Sub category'].str.replace('NetworkingDevices', 'Networking Devices')
df1['Sub category'] = df1['Sub category'].str.replace('OfficePaperProducts', 'Office Paper Products')
df1['Sub category'] = df1['Sub category'].str.replace('ExternalDevices', 'External Devices')
df1['Sub category'] = df1['Sub category'].str.replace('DataStorage', 'Data Storage')
df1['Sub category'] = df1['Sub category'].str.replace('HomeStorage', 'Home Storage')
df1['Sub category'] = df1['Sub category'].str.replace('HomeAudio', 'Home Audio')
df1['Sub category'] = df1['Sub category'].str.replace('GeneralPurposeBatteries', 'General Purpose Batteries')
df1['Sub category'] = df1['Sub category'].str.replace('BatteryChargers', 'Battery Chargers')
df1['Sub category'] = df1['Sub category'].str.replace('CraftMaterials', 'Craft Materials')
df1['Sub category'] = df1['Sub category'].str.replace('OfficeElectronics', 'Office Electronics')
df1['Sub category'] = df1['Sub category'].str.replace('PowerAccessories', 'Power Accessories')
df1['Sub category'] = df1['Sub category'].str.replace('CarAccessories', 'Car Accessories')
df1['Sub category'] = df1['Sub category'].str.replace('HomeMedicalSupplies', 'Home Medical Supplies')
df1['Sub category'] = df1['Sub category'].str.replace('HomeTheater', 'Home Theater')
```

In [67]:
```
df.head()
```

Out[67]:

	product_id	product_name	discounted_price	actual_price	discount_percentage	rating	rating_count	about_product	user_id
0	B07JW9H4J1	Wayona Nylon Braided USB to Lightning Fast Cha...	399.0	1099.0	63.694268	4.2	24269.0	High Compatibility : Compatible With iPhone 12...	AG3D6O4STAQKAY2UVGEUV46KN35Q,AHMY5CWJMMK5BJRBB...
1	B098NS6PVG	Ambrane Unbreakable 60W / 3A Fast Charging 1.5...	199.0	349.0	42.979943	4.0	43994.0	Compatible with all Type C enabled devices, be...	AECPFYFQVRUWC3KGNLJIOREFP5LQ,AGYYVPDD7YG7FYNBX...
2	B096MSW6CT	Sounce Fast Phone Charging Cable & Data Sync U...	199.0	1899.0	89.520800	3.9	7928.0	【 Fast Charger& Data Sync】-With built-in safet...	AGU3BBQ2V2DDAMOAKGFAWDDQ6QHA,AESFLDV2PT363T2AQ...
3	B08HDJ86NZ	boAt Deuce USB 300 2 in 1 Type-C & Micro USB S...	329.0	699.0	52.932761	4.2	94363.0	The boAt Deuce USB 300 2 in 1 cable is compati...	AEWAZDZZJLQUYVOVGBEUKSLXHQ5A,AG5HTSFRRE6NL3M5S...
4	B08CF3B7N1	Portronics Konnect L 1.2M Fast Charging 3A 8 P...	154.0	399.0	61.403509	4.2	16905.0	[CHARGE & SYNC FUNCTION]- This cable comes wit...	AE3Q6KSUK5P75D5HFYHCRAOLODSA,AFUGIFH5ZAFXRDSZH...

1 What is the average rating for each product category?

In [68]:
```
average_rating_per_category= df.groupby('Category')['rating'].mean().reset_index()

# Rename the columns for better readability
average_rating_per_category.columns = ['Product Category', 'Average Rating']

# Format the 'Average Rating' column to 2 decimal places
average_rating_per_category['Average Rating'] = average_rating_per_category['Average Rating'].round(2)


# Print the result
print(average_rating_per_category)
```

```
         Product Category  Average Rating
0          Car & Motorbike            3.80
1  Computers & Accessories            4.15
2              Electronics            4.08
3    Health & PersonalCare            4.00
4          Home & Kitchen            4.04
5        Home Improvement            4.25
6     Musical Instruments            3.90
7          Office Products            4.31
8             Toys & Games            4.30
```

What are the top rating_count products by category?

In [69]:
```
#Group by category and sub_category , then count the number of rating of each product
rating_count_by_product= df.groupby(['Category','Sub category'])['rating'].count().reset_index()

# Rename the columns for better readability
rating_count_by_product.columns = ['Product Category', 'Product Sub Category', 'Rating Count']

# Sort the products within each category by rating count in descending order
rating_count_by_product = rating_count_by_product.sort_values(['Product Category', 'Rating Count'], ascending=[True, False])

# Find the top product by rating count for each category
top_products_by_category = rating_count_by_product.groupby('Product Category').head(1)

# Print the result
print(top_products_by_category)
```

	Product Category	Product Sub Category	Rating Count
0	Car & Motorbike	CarAccessories	1
1	Computers & Accessories	Accessories&Peripherals	381
14	Electronics	HomeTheater,TV&Video	162
18	Health & PersonalCare	HomeMedicalSupplies&Equipment	1
23	Home & Kitchen	Kitchen&HomeAppliances	308
24	Home Improvement	Electrical	2
25	Musical Instruments	Microphones	2
27	Office Products	OfficePaperProducts	27
28	Toys & Games	Arts&Crafts	1

In [102…

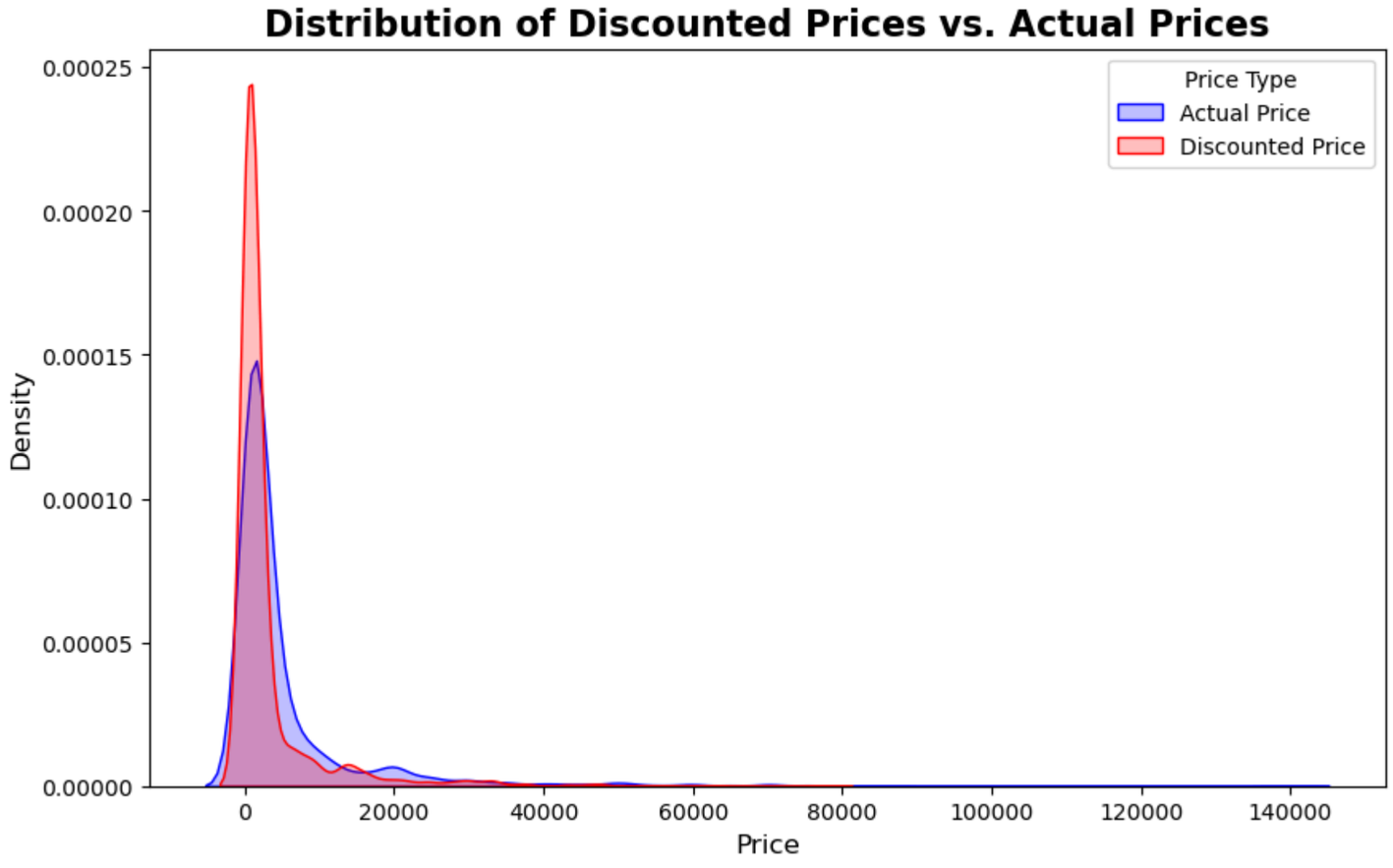
```
#Group by category and sub_category , then count the number of rating of each product
rating_count_by_product= df.groupby('Category')['rating'].count().reset_index()

# Rename the columns for better readability
rating_count_by_product.columns = ['Product Category', 'Rating Count']

# Sort the products within each category by rating count in descending order
rating_count_by_product = rating_count_by_product.sort_values(['Product Category', 'Rating Count'], ascending=[True, False])

# Find the top product by rating count for each category
top_products_by_category = rating_count_by_product.groupby('Product Category').head(1)

# Print the result
print(top_products_by_category)
```

	Product Category	Rating Count
0	Car & Motorbike	1
1	Computers & Accessories	453
2	Electronics	526
3	Health & PersonalCare	1
4	Home & Kitchen	448
5	Home Improvement	2
6	Musical Instruments	2
7	Office Products	31
8	Toys & Games	1

What is the distribution of discounted prices vs. actual prices?

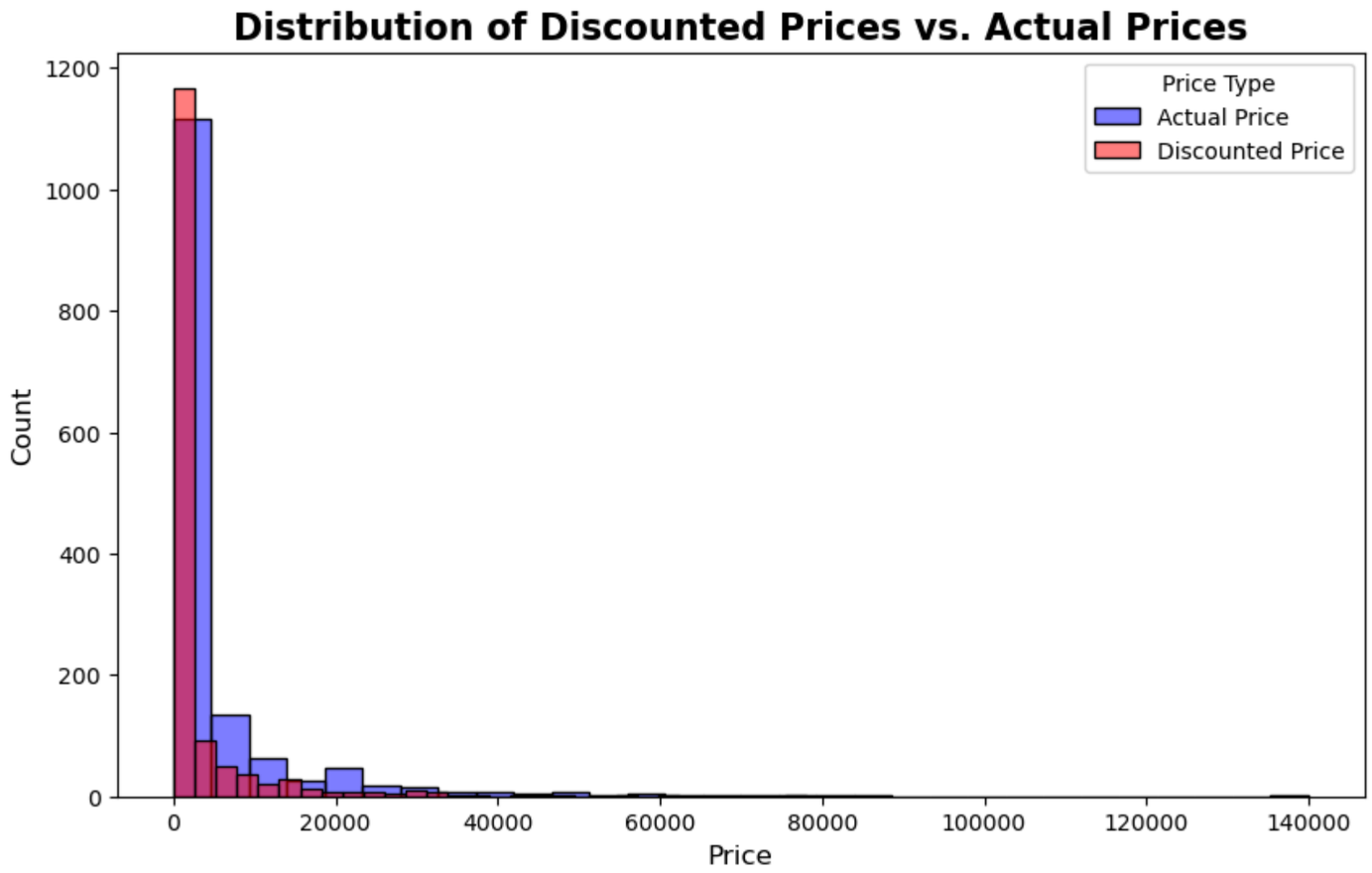
In [71]:

```
# Plot the distributions of actual_price and discounted_price
plt.figure(figsize=(10, 6))

# Plot the actual prices distribution
sns.kdeplot(df['actual_price'], label='Actual Price', color='blue', shade=True, bw_adjust=0.7)

# Plot the discounted prices distribution
sns.kdeplot(df['discounted_price'], label='Discounted Price', color='red', shade=True, bw_adjust=0.7)

# Add title and labels
plt.title('Distribution of Discounted Prices vs. Actual Prices', fontsize=16, fontweight='bold')
plt.xlabel('Price', fontsize=12)
plt.ylabel('Density', fontsize=12)
plt.legend(title='Price Type')

# Show the plot
plt.show()
```



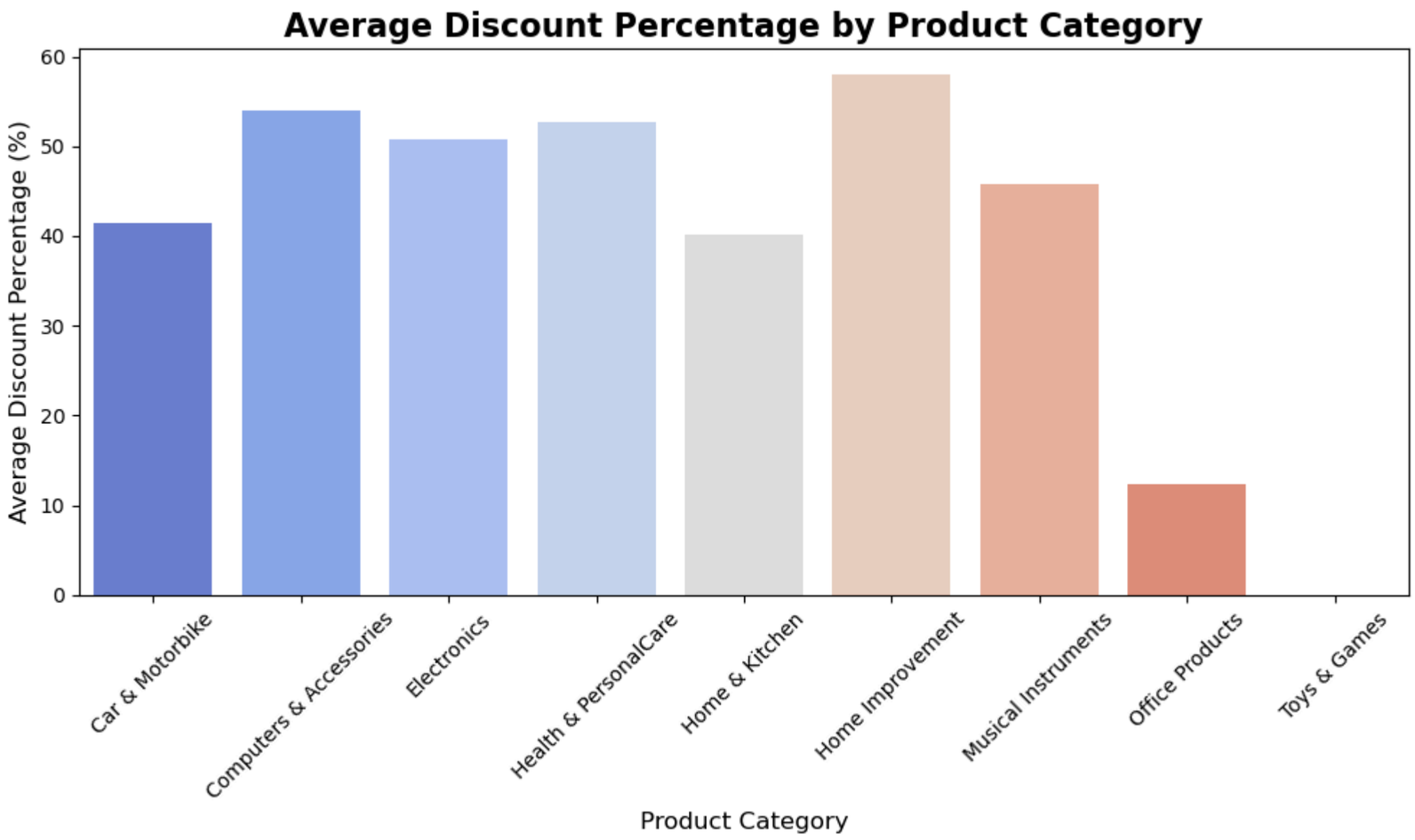
In [72]:

```
plt.figure(figsize=(10, 6))

# Plot the actual prices distribution
sns.histplot(df['actual_price'], label='Actual Price', color='blue', bins=30, alpha=0.5)

# Plot the discounted prices distribution
sns.histplot(df['discounted_price'], label='Discounted Price', color='red', bins=30, alpha=0.5)

# Add title and labels
plt.title('Distribution of Discounted Prices vs. Actual Prices', fontsize=16, fontweight='bold')
plt.xlabel('Price', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.legend(title='Price Type')

# Show the plot
plt.show()
```



How does the average discount percentage vary across categories?

In [73]:
```
# Calculate the discount percentage
df['discount_percentage'] = ((df['actual_price'] - df['discounted_price']) / df['actual_price']) * 100

# Group by product category and calculate the average discount percentage
avg_discount_by_category = df.groupby('Category')['discount_percentage'].mean().reset_index()

# Rename columns for clarity
avg_discount_by_category.columns = ['Product Category', 'Average Discount Percentage']

# Plot the average discount percentage across categories
plt.figure(figsize=(10, 6))
sns.barplot(x='Product Category', y='Average Discount Percentage', data=avg_discount_by_category, palette='coolwarm')

# Add title and labels
plt.title('Average Discount Percentage by Product Category', fontsize=16, fontweight='bold')
plt.xlabel('Product Category', fontsize=12)
plt.ylabel('Average Discount Percentage (%)', fontsize=12)
plt.xticks(rotation=45)

# Show the plot
plt.tight_layout()
plt.show()
```



What are the most popular product names?

In [74]:
```
# Count the number of occurrences of each product
product_counts = df['product_name'].value_counts().reset_index()
product_counts.columns = ['Product Name', 'Count']

# Print the most popular products
print(product_counts.head(1))
```

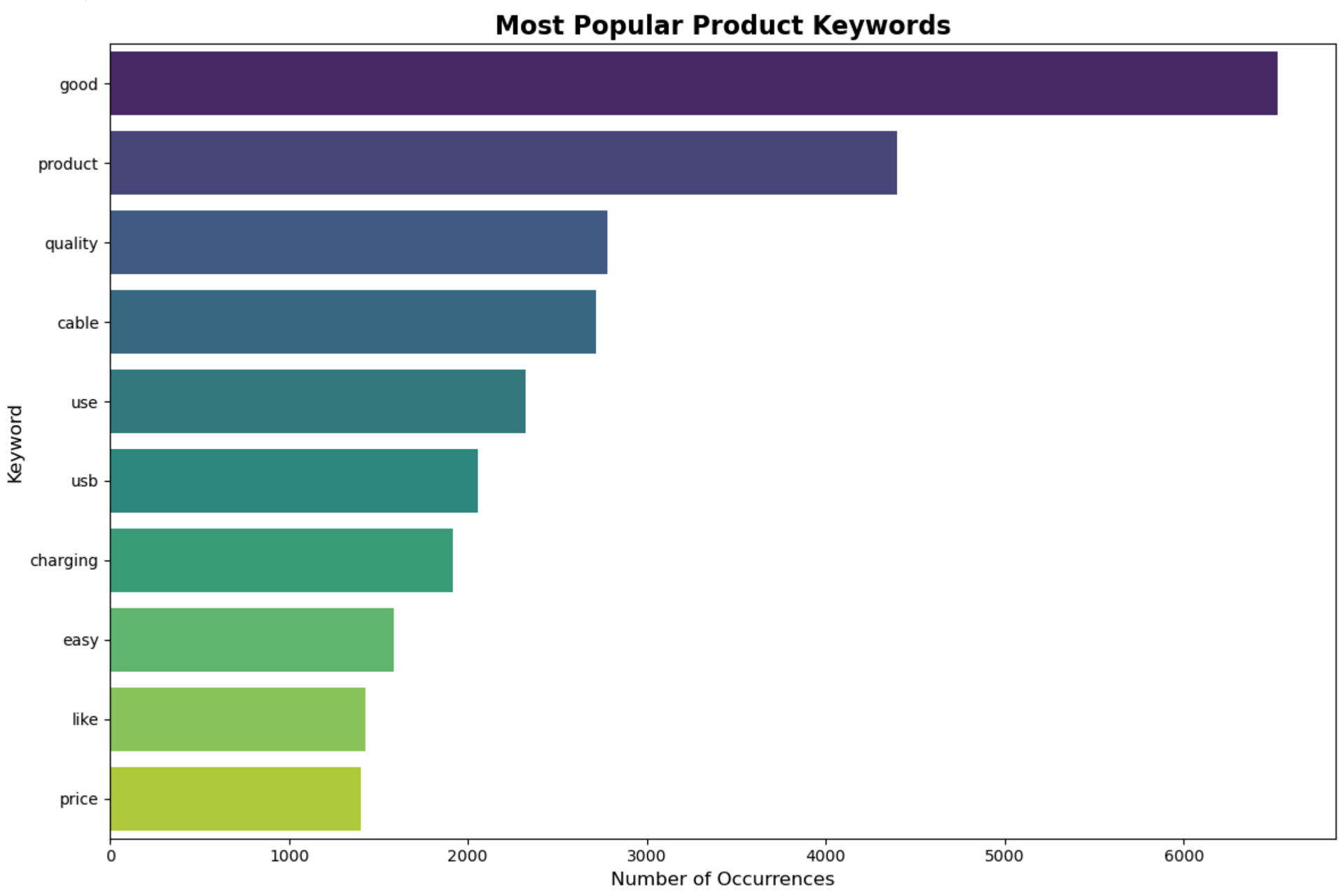
```
                                        Product Name  Count
0  Fire-Boltt Ninja Call Pro Plus 1.83" Smart Wat...      5
```

6.What are the most popular product keywords?

In [59]:
```
from sklearn.feature_extraction.text import CountVectorizer
# Assuming 'df' is your DataFrame containing relevant columns
# Combine text data from multiple columns into a single Series
df['combined_text'] = df['product_name'].fillna('') + ' ' + df['about_product'].fillna('') + ' ' + df['review_content'].fillna('')

# Initialize CountVectorizer to tokenize words and remove stop words
vectorizer = CountVectorizer(stop_words='english')

# Fit and transform the combined text data
X = vectorizer.fit_transform(df['combined_text'])

# Get feature names and sum the occurrences of each word
word_counts = X.sum(axis=0).A1
words = vectorizer.get_feature_names_out()

# Create a DataFrame for the words and their counts
keyword_counts = pd.DataFrame({'Keyword': words, 'Count': word_counts})
```

```
# Sort the keywords by count in descending order
keyword_counts = keyword_counts.sort_values(by='Count', ascending=False)

# Print the most popular keywords
print(keyword_counts.head(10))

# Plot the most popular keywords
plt.figure(figsize=(12, 8))
sns.barplot(x='Count', y='Keyword', data=keyword_counts.head(10), palette='viridis')

# Add title and labels
plt.title('Most Popular Product Keywords', fontsize=16, fontweight='bold')
plt.xlabel('Number of Occurrences', fontsize=12)
plt.ylabel('Keyword', fontsize=12)

# Show the plot
plt.tight_layout()
plt.show()
```

```
       Keyword  Count
8255       good   6522
13153   product   4396
13445   quality   2779
4299      cable   2713
17263       use   2320
17254       usb   2056
4645   charging   1919
6723       easy   1584
10327      like   1428
13049     price   1404
```

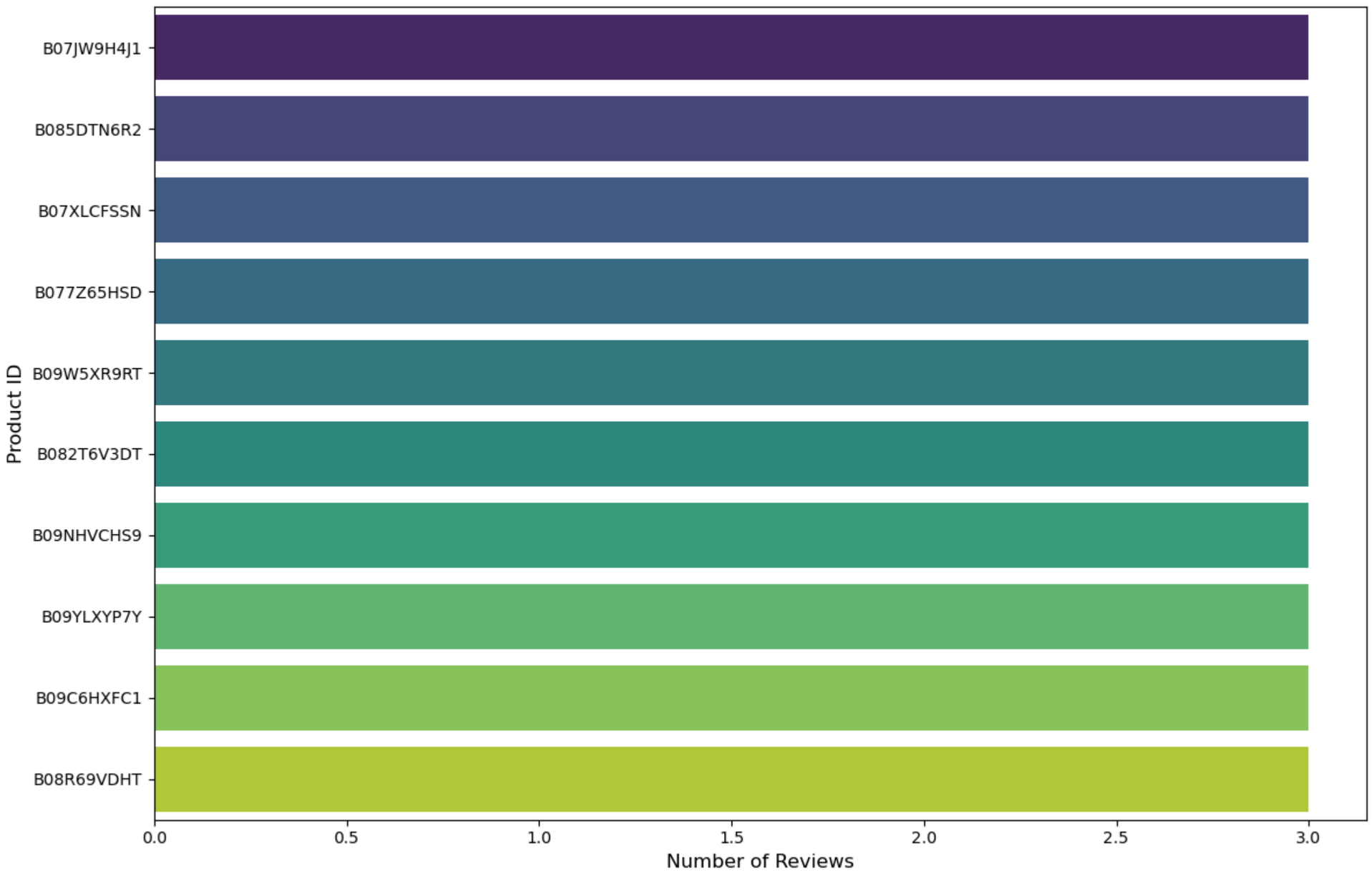


7. What are the most popular product reviews?

In [75]:
```
#Count the Number of Reviews per Product

# Count the number of reviews per product
reviews_count = df['product_id'].value_counts().reset_index()
reviews_count.columns = ['Product ID', 'Number of Reviews']

# Print the most reviewed products
print(reviews_count.head(10))
```

```
   Product ID  Number of Reviews
0  B07JW9H4J1                  3
1  B085DTN6R2                  3
2  B07XLCFSSN                  3
3  B077Z65HSD                  3
4  B09W5XR9RT                  3
5  B082T6V3DT                  3
6  B09NHVCHS9                  3
7  B09YLXYP7Y                  3
8  B09C6HXFC1                  3
9  B08R69VDHT                  3
```

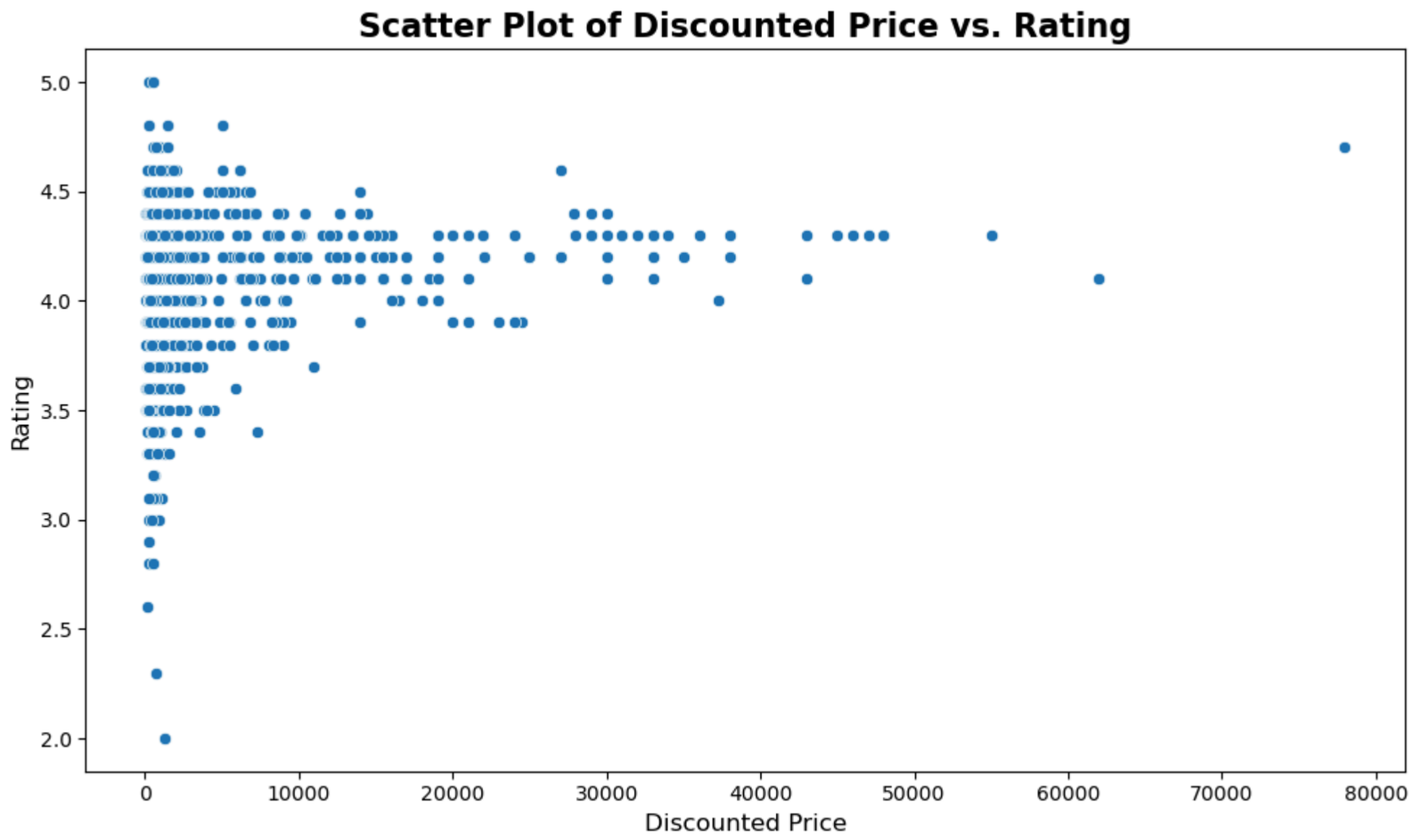
In [78]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming you want to visualize the most reviewed products
plt.figure(figsize=(12, 8))
sns.barplot(x='Number of Reviews', y='Product ID', data=reviews_count.head(10), palette='viridis')

# Add title and labels
plt.title('Most Reviewed Products', fontsize=16, fontweight='bold')
plt.xlabel('Number of Reviews', fontsize=12)
plt.ylabel('Product ID', fontsize=12)

# Show the plot
plt.tight_layout()
plt.show()
```



What is the correlation between discounted_price and rating?

In [80]:
```
# Calculate the Pearson correlation coefficient
correlation = df_cleaned['discounted_price'].corr(df_cleaned['rating'])
print(f"Pearson correlation coefficient: {correlation:.2f}")

# Optional: Visualize the relationship with a scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x='discounted_price', y='rating', data=df_cleaned, palette='viridis')

# Add titles and labels
plt.title('Scatter Plot of Discounted Price vs. Rating', fontsize=16, fontweight='bold')
plt.xlabel('Discounted Price', fontsize=12)
plt.ylabel('Rating', fontsize=12)

# Show the plot
plt.tight_layout()
plt.show()
```

```
Pearson correlation coefficient: 0.12
```



Interpretation:

- Positive Correlation: If the correlation coefficient is close to 1, higher discounted prices tend to be associated with higher ratings.
- Negative Correlation: If the coefficient is close to -1, higher discounted prices tend to be associated with lower ratings.
- No Correlation: A coefficient near 0 suggests no significant linear relationship between discounted price and rating.

9. What are the Top 5 categories based on the highest ratings?

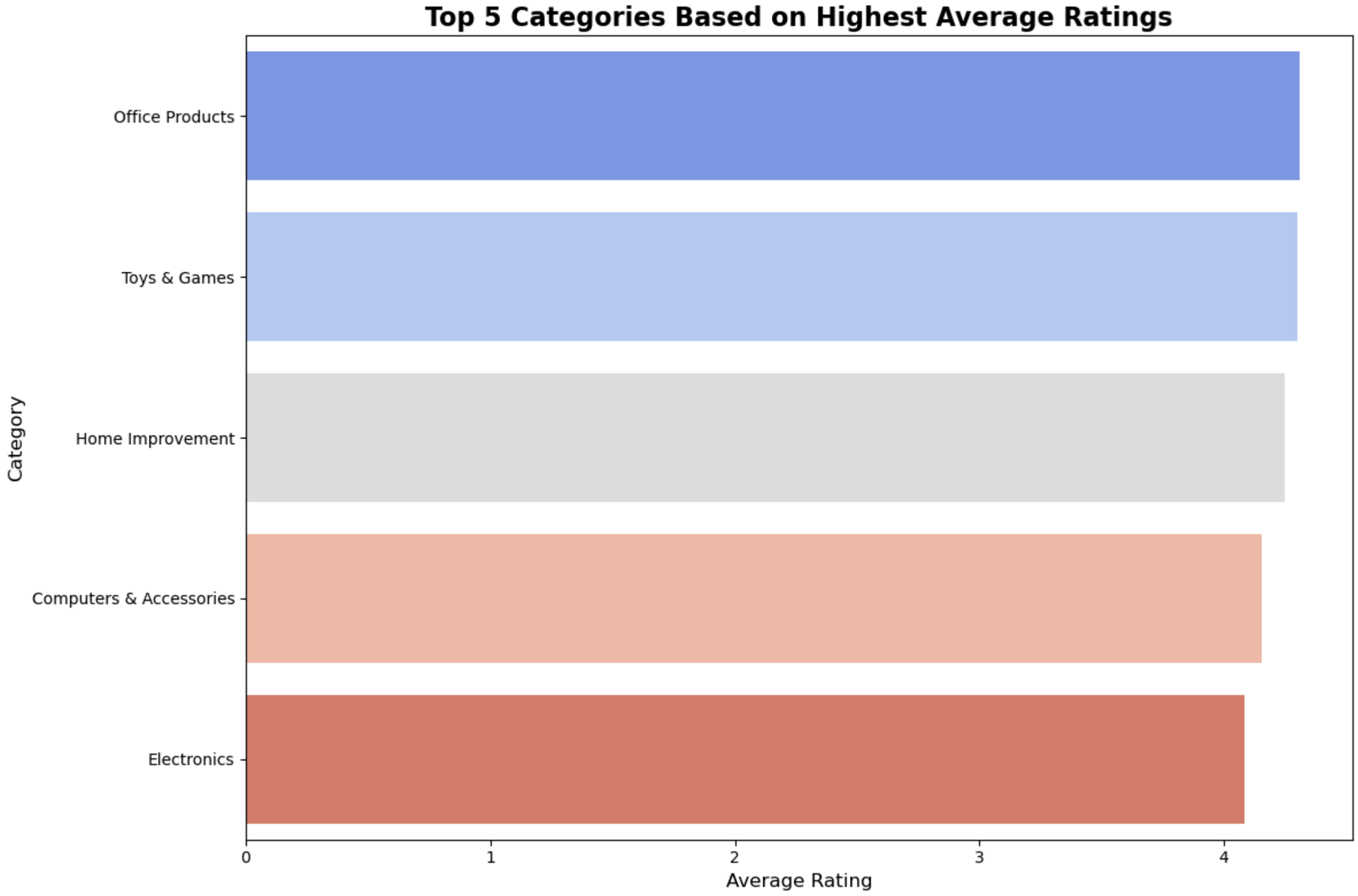
In [87]:
```
# Calculate the average rating for each category
avg_ratings_by_category = df.groupby('Category')['rating'].mean().reset_index()

# Sort categories by average rating in descending order
avg_ratings_by_category = avg_ratings_by_category.sort_values(by='rating', ascending=False)

# Select the top 5 categories
top_5_categories = avg_ratings_by_category.head(5)

# Print the top 5 categories with highest ratings
print(top_5_categories)
```

	Category	rating
7	Office Products	4.309677
8	Toys & Games	4.300000
5	Home Improvement	4.250000
1	Computers & Accessories	4.154967
2	Electronics	4.081749

In [85]:
```
df.columns
```

Out[85]:
```
Index(['product_id', 'product_name', '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', 'combined_text', 'Category',
       'Sub category'],
      dtype='object')
```

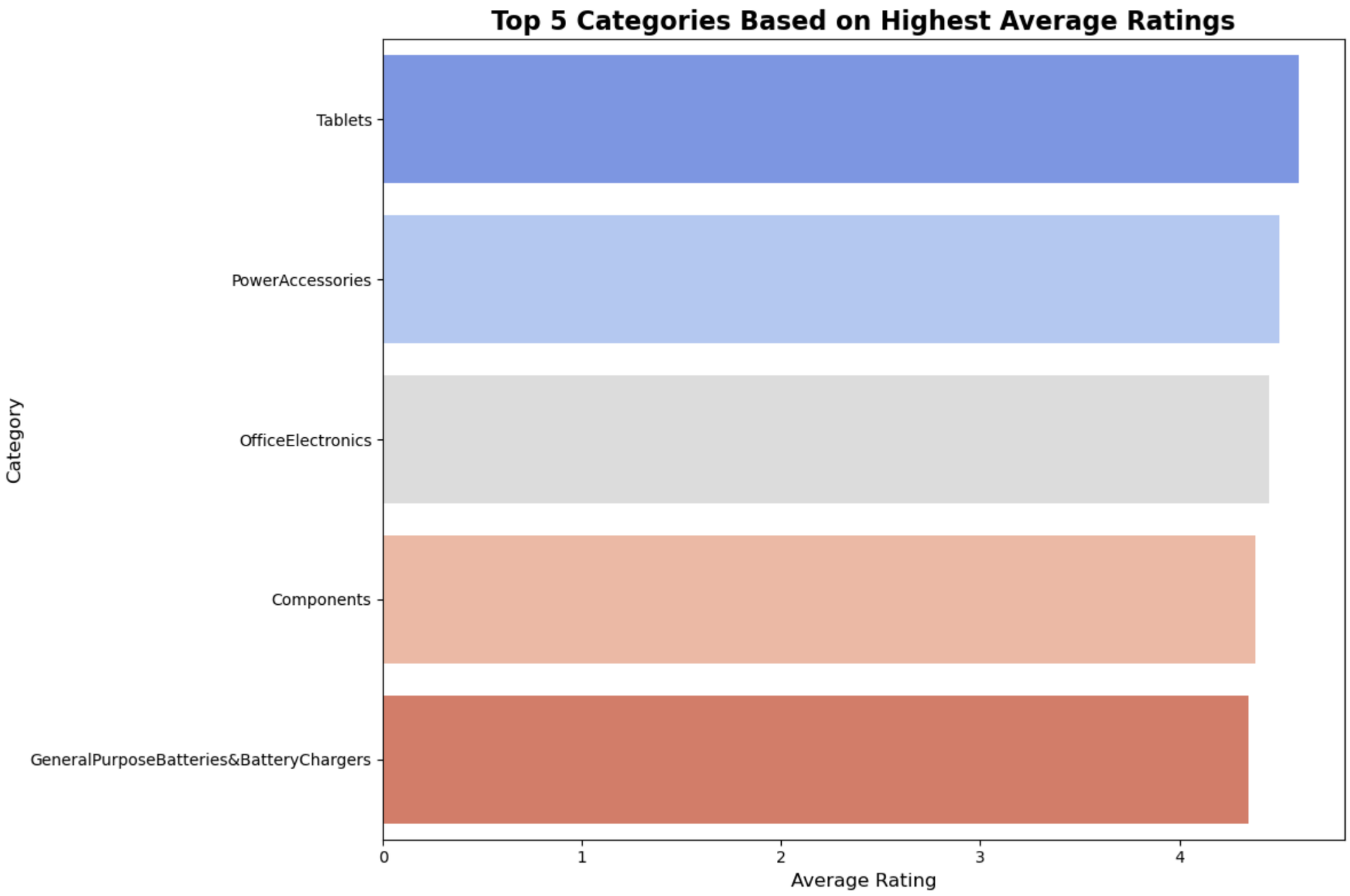
In [88]:
```
# Plot the top 5 categories based on average rating
plt.figure(figsize=(12, 8))
sns.barplot(x='rating', y='Category', data=top_5_categories, palette='coolwarm')

# Add title and labels
plt.title('Top 5 Categories Based on Highest Average Ratings', fontsize=16, fontweight='bold')
plt.xlabel('Average Rating', fontsize=12)
plt.ylabel('Category', fontsize=12)

# Show the plot
plt.tight_layout()
plt.show()
```



In [95]:
```
# Calculate the average rating for each category and subcategory
avg_ratings_by_category = df.groupby(['Category','Sub category'])['rating'].mean().reset_index()

# Sort categories by average rating in descending order
avg_ratings_by_category = avg_ratings_by_category.sort_values(by='rating', ascending=False)

# Select the top 5 categories
top_5_Sub_categories = avg_ratings_by_category.head(5)

# Print the top 5 categories with highest ratings
print(top_5_Sub_categories)
```

	Category	Sub category	rating
8	Computers & Accessories	Tablets	4.60
16	Electronics	PowerAccessories	4.50
26	Office Products	OfficeElectronics	4.45
2	Computers & Accessories	Components	4.38
11	Electronics	GeneralPurposeBatteries&BatteryChargers	4.35

In [98]:
```
# Plot the top 5 categories and Sub category based on average rating
plt.figure(figsize=(12, 8))
sns.barplot(x='rating', y='Sub category', data=top_5_Sub_categories, palette='coolwarm')

# Add title and labels
plt.title('Top 5 Categories Based on Highest Average Ratings', fontsize=16, fontweight='bold')
plt.xlabel('Average Rating', fontsize=12)
plt.ylabel('Category', fontsize=12)

# Show the plot
plt.tight_layout()
plt.show()
```



10. Identify any potential areas for improvement or optimization based on the data analysis.

In [101…

```
# Correlation between discounted price and rating
correlation = df['discounted_price'].corr(df['rating'])
print(f"Correlation between discounted price and rating: {correlation:.2f}")

# Average rating by category
avg_ratings_by_category = df.groupby('Category')['rating'].mean().reset_index()

# Identify categories with low average ratings
low_rated_categories = avg_ratings_by_category[avg_ratings_by_category['rating'] < avg_ratings_by_category['rating'].mean()]
print("Categories with below-average ratings:")
print(low_rated_categories)

# Analyzing review content for keywords (optional)
from sklearn.feature_extraction.text import CountVectorizer

# Initialize CountVectorizer
vectorizer = CountVectorizer(stop_words='english')
X = vectorizer.fit_transform(df['review_content'].fillna(''))
word_counts = X.sum(axis=0).A1
words = vectorizer.get_feature_names_out()
keyword_counts = pd.DataFrame({'Keyword': words, 'Count': word_counts})
keyword_counts = keyword_counts.sort_values(by='Count', ascending=False)
print("Most common review keywords:")
print(keyword_counts.head(10))
```

```
Correlation between discounted price and rating: 0.12
Categories with below-average ratings:
               Category    rating
0        Car & Motorbike  3.800000
2            Electronics  4.081749
3  Health & PersonalCare  4.000000
4         Home & Kitchen  4.040625
6    Musical Instruments  3.900000
Most common review keywords:
        Keyword  Count
5817       good   6455
9601    product   3947
9825    quality   2417
12825       use   1730
9507      price   1393
2756      cable   1380
7386       like   1243
9144      phone   1094
3027   charging   1010
12855     using   1006
```

- Address Product Issues: Focus on improving products with low ratings and leverage successful products for marketing and development.
- Optimize Pricing: Adjust pricing strategies based on the relationship between price, discount, and customer satisfaction.
- Improve Categories: Invest in high-performing categories and improve or reevaluate underperforming ones.
- Enhance Data Quality: Ensure data accuracy and completeness for more reliable insights.

In [ ]: