The Souled Store: Sneaker Analysis

By, Riswi Ayub

The Souled Store is an Indian lifestyle brand known for its trendy and pop-culture-inspired apparel, accessories, and merchandise. Founded in 2013, it offers a wide range of products featuring official collaborations with popular franchises like Marvel, DC, Disney, and more. The brand is popular for its quirky designs, high-quality materials, and focus on casual fashion.

About the dataset:

- **The data for this analysis was collected individually from The Souled Store's official website using web scraping techniques.** This method allowed for the extraction of product details, pricing, categories, and other relevant information to gain insights into the brand's offerings and trends.
- The dataset contains information about men's footwear products, primarily sneakers, with details such as product names, brands, categories, pricing, customer ratings, and popularity metrics. Key features include:

  Product Details: product_id, product_name, brand, product_category. **Pricing:** original_price, special_price, exclusive_price. **Customer Feedback:** average_rating, total_ratings. **Inventory & Popularity:** product_quantity, sort_order. **Target Audience:** target_gender (1 = Male).

Preprocessing:

In [51]:

```
import pandas as pd
import numpy as np
```

Loading the data

In [52]:

```
df = pd.read_csv('/Users/riswee/Desktop/api_data.csv')
```

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

Out[53]:

	id	product	artist	category	price	genderType	stock	avgRating	ratir
0	279718	Kanso: Beige	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men Low Top Sneakers"}	2399	1	0	4.27	
1	272597	Kanso: Black	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men Low Top Sneakers"}	2299	1	0	4.00	
2	276043	TSS Originals: Earth	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men High Top Sneakers"}	2999	1	0	5.00	
3	279722	Black Panther: Warrior	{"name": "Marvel\u0099", "slug": "marvel-offic...	{"name": "Men High Top Sneakers"}	3499	1	0	4.59	
4	287843	Urban Blaze: Hawkins	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men Low Top Sneakers"}	2699	1	0	4.27	

5 rows × 25 columns

Checking basic info

In [54]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 91 entries, 0 to 90
Data columns (total 25 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   id                91 non-null     int64  
 1   product           91 non-null     object 
 2   artist            91 non-null     object 
 3   category          91 non-null     object 
 4   price             91 non-null     int64  
 5   genderType        91 non-null     int64  
 6   stock             91 non-null     int64  
 7   avgRating         91 non-null     float64
 8   ratingCount       91 non-null     int64  
 9   prodQty           91 non-null     int64  
 10  prodType          91 non-null     int64  
 11  splPrice          91 non-null     int64  
 12  exclusivePrice    91 non-null     int64  
 13  sortOrder         91 non-null     int64  
 14  images            91 non-null     object 
 15  imagesNew         91 non-null     object 
 16  extraPrice        91 non-null     int64  
 17  isPrintable       91 non-null     bool   
 18  jitValue          91 non-null     object 
 19  product_slug      91 non-null     object 
 20  isBlurOnPlp       91 non-null     bool   
 21  plpCoverImage     0 non-null      float64
 22  targetDate        91 non-null     object 
 23  isProductLocked   91 non-null     bool   
 24  tiptiles          0 non-null      float64
dtypes: bool(3), float64(3), int64(11), object(8)
memory usage: 16.0+ KB
```

Descriptive Statistics

In [55]:

```
df.describe()
```

Out[55]:

```
                  id        price  genderType  stock  avgRating  ratingCount      prodQt
count      91.000000    91.000000        91.0   91.0  91.000000    91.000000    91.00000
mean   263141.505495  2562.736264         1.0    0.0   4.322088   160.164835   590.37362
std     18963.282608   630.787248         0.0    0.0   0.523311   183.219193   550.25175
min    191421.000000   999.000000         1.0    0.0   0.000000     0.000000     0.00000
25%    253432.500000  2499.000000         1.0    0.0   4.230000    44.000000    98.50000
50%    266371.000000  2699.000000         1.0    0.0   4.390000    90.000000   463.00000
75%    276042.500000  2899.000000         1.0    0.0   4.485000   213.000000   940.00000
max    287856.000000  3499.000000         1.0    0.0   5.000000  1047.000000  2577.00000
```

Renaming the columns

In [56]:

```
column_rename_map = {
    'id': 'product_id',
    'product': 'product_name',
    'artist': 'brand',
    'category': 'product_category',
    'price': 'original_price',
    'genderType': 'target_gender',
    'stock': 'stock_quantity',
    'avgRating': 'average_rating',
    'ratingCount': 'total_ratings',
    'prodQty': 'product_quantity',
    'prodType': 'product_type',
    'splPrice': 'special_price',
    'exclusivePrice': 'exclusive_price',
    'sortOrder': 'sort_order',
    'images': 'image_urls',
    'imagesNew': 'image_urls_by_gender',
    'extraPrice': 'extra_price',
    'isPrintable': 'is_printable',
    'jitValue': 'jit_value',
    'product_slug': 'product_slug',
    'isBlurOnPlp': 'is_blur_on_plp',
    'plpCoverImage': 'plp_cover_image',
    'targetDate': 'target_date',
    'isProductLocked': 'is_product_locked',
    'tiptiles': 'tip_tiles'
}
```

In [57]:

```
df.rename(columns=column_rename_map, inplace=True)
```

In [58]:
```
df.head()
```

Out[58]:

	product_id	product_name	brand	product_category	original_price	target_gende
0	279718	Kanso: Beige	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men Low Top Sneakers"}	2399	
1	272597	Kanso: Black	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men Low Top Sneakers"}	2299	
2	276043	TSS Originals: Earth	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men High Top Sneakers"}	2999	
3	279722	Black Panther: Warrior	{"name": "Marvel\u0099", "slug": "marvel-offic...	{"name": "Men High Top Sneakers"}	3499	
4	287843	Urban Blaze: Hawkins	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men Low Top Sneakers"}	2699	

5 rows × 25 columns

Checking for missing values

In [59]:
```
df.isnull().sum()
```

Out[59]:
```
product_id                0
product_name              0
brand                     0
product_category          0
original_price            0
target_gender             0
stock_quantity            0
average_rating            0
total_ratings             0
product_quantity          0
product_type              0
special_price             0
exclusive_price           0
sort_order                0
image_urls                0
image_urls_by_gender      0
extra_price               0
is_printable              0
jit_value                 0
product_slug              0
is_blur_on_plp            0
plp_cover_image          91
target_date               0
is_product_locked         0
tip_tiles                91
dtype: int64
```

Filling missing 'target_date' values with 'Unknown'

In [60]:
```
df['target_date'] = df['target_date'].fillna('Unknown')

df.to_csv('cleaned_api_data_final.csv', index=False)

print("\nFinal Cleaned Dataset:")
df.head()
```

```
Final Cleaned Dataset:
```

Out[60]:

	product_id	product_name	brand	product_category	original_price	target_gende
0	279718	Kanso: Beige	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men Low Top Sneakers"}	2399	
1	272597	Kanso: Black	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men Low Top Sneakers"}	2299	
2	276043	TSS Originals: Earth	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men High Top Sneakers"}	2999	
3	279722	Black Panther: Warrior	{"name": "Marvel\u0099", "slug": "marvel-offic...	{"name": "Men High Top Sneakers"}	3499	
4	287843	Urban Blaze: Hawkins	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men Low Top Sneakers"}	2699	

5 rows × 25 columns

In [61]:
```
df.isnull().sum()
```

Out[61]:
```
product_id              0
product_name            0
brand                   0
product_category        0
original_price          0
target_gender           0
stock_quantity          0
average_rating          0
total_ratings           0
product_quantity        0
product_type            0
special_price           0
exclusive_price         0
sort_order              0
image_urls              0
image_urls_by_gender    0
extra_price             0
is_printable            0
jit_value               0
product_slug            0
is_blur_on_plp          0
plp_cover_image        91
target_date             0
is_product_locked       0
tip_tiles              91
dtype: int64
```

Converting 'target_date' to datetime format

In [62]:
```
df['target_date'] = pd.to_datetime(df['target_date'], errors='coerc
```

In [63]:
```
df.head()
```

Out[63]:

	product_id	product_name	brand	product_category	original_price	target_gende
0	279718	Kanso: Beige	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men Low Top Sneakers"}	2399	
1	272597	Kanso: Black	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men Low Top Sneakers"}	2299	
2	276043	TSS Originals: Earth	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men High Top Sneakers"}	2999	
3	279722	Black Panther: Warrior	{"name": "Marvel\u0099", "slug": "marvel-offic...	{"name": "Men High Top Sneakers"}	3499	
4	287843	Urban Blaze: Hawkins	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men Low Top Sneakers"}	2699	

5 rows × 25 columns

Dropping columns that are irrelevant

In [64]:
```
df.dropna(how='all', inplace=True)
```

In [65]:
```
df.drop(columns=['stock_quantity'], inplace=True)
```

In [66]:
```
df.drop(columns=['extra_price'], inplace=True)
```

In [67]:
```
df.drop(columns=['plp_cover_image', 'tip_tiles'], inplace=True)
```

In [68]:
```
df.drop(columns=['image_urls', 'image_urls_by_gender'], inplace=Tru
```

In [69]:

```
df.head()
```

Out[69]:

	product_id	product_name	brand	product_category	original_price	target_gende
0	279718	Kanso: Beige	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men Low Top Sneakers"}	2399	
1	272597	Kanso: Black	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men Low Top Sneakers"}	2299	
2	276043	TSS Originals: Earth	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men High Top Sneakers"}	2999	
3	279722	Black Panther: Warrior	{"name": "Marvel\u0099", "slug": "marvel-offic...	{"name": "Men High Top Sneakers"}	3499	
4	287843	Urban Blaze: Hawkins	{"name": "The Souled Store", "slug": "the-soul...	{"name": "Men Low Top Sneakers"}	2699	

Extracting just the name of the product category

In [70]:

```
import ast
```

In [71]:

```
def extract_category_name(category_str):
    try:
        category_dict = ast.literal_eval(category_str)
        return category_dict.get('name', 'Unknown')
    except (ValueError, SyntaxError):
        return 'Unknown'
```

In [72]:

```
df['product_category'] = df['product_category'].apply(extract_categ
```

In [73]:

```
print("\nCleaned 'product_category' Column:")
print(df[['product_id', 'product_name', 'product_category']].head()
```

```
Cleaned 'product_category' Column:
   product_id            product_name      product_category
0      279718            Kanso: Beige    Men Low Top Sneakers
1      272597            Kanso: Black    Men Low Top Sneakers
2      276043    TSS Originals: Earth   Men High Top Sneakers
3      279722  Black Panther: Warrior   Men High Top Sneakers
4      287843    Urban Blaze: Hawkins    Men Low Top Sneakers
```

In [74]:

```
df.head()
```

Out[74]:

	product_id	product_name	brand	product_category	original_price	target_gende
0	279718	Kanso: Beige	{"name": "The Souled Store", "slug": "the-soul...	Men Low Top Sneakers	2399	
1	272597	Kanso: Black	{"name": "The Souled Store", "slug": "the-soul...	Men Low Top Sneakers	2299	
2	276043	TSS Originals: Earth	{"name": "The Souled Store", "slug": "the-soul...	Men High Top Sneakers	2999	
3	279722	Black Panther: Warrior	{"name": "Marvel\u0099", "slug": "marvel-offic...	Men High Top Sneakers	3499	
4	287843	Urban Blaze: Hawkins	{"name": "The Souled Store", "slug": "the-soul...	Men Low Top Sneakers	2699	

Extracting {name,slug} from brand column

In [75]:
```
def transform_brand(brand_str):
    try:
        brand_dict = ast.literal_eval(brand_str)
        name = brand_dict.get('name', 'Unknown')
        slug = brand_dict.get('slug', 'unknown-slug')
        return f"{{{name}, {slug}}}"
    except (ValueError, SyntaxError):
        return "{Unknown, unknown-slug}"

df['brand'] = df['brand'].apply(transform_brand)

print("\nTransformed 'brand' Column:")
print(df[['product_id', 'product_name', 'brand']].head())
```

```
Transformed 'brand' Column:
   product_id           product_name  \
0      279718           Kanso: Beige
1      272597           Kanso: Black
2      276043    TSS Originals: Earth
3      279722  Black Panther: Warrior
4      287843    Urban Blaze: Hawkins

                                            brand
0  {The Souled Store, the-souled-store-originals}
1  {The Souled Store, the-souled-store-originals}
2  {The Souled Store, the-souled-store-originals}
3          {Marvel□, marvel-official-merchandise}
4  {The Souled Store, the-souled-store-originals}
```

In [76]:
```
df.to_csv('cleaned_api_data_final.csv', index=False)
```

In [77]:
```
df.head()
```

Out[77]:

	product_id	product_name	brand	product_category	original_price	target_gender
0	279718	Kanso: Beige	{The Souled Store, the-souled-store-originals}	Men Low Top Sneakers	2399	1
1	272597	Kanso: Black	{The Souled Store, the-souled-store-originals}	Men Low Top Sneakers	2299	1
2	276043	TSS Originals: Earth	{The Souled Store, the-souled-store-originals}	Men High Top Sneakers	2999	1
3	279722	Black Panther: Warrior	{Marvel□, marvel-official-merchandise}	Men High Top Sneakers	3499	1
4	287843	Urban Blaze: Hawkins	{The Souled Store, the-souled-store-originals}	Men Low Top Sneakers	2699	1

Checking for duplicates

In [78]:
```
df.duplicated().sum()
```

Out[78]:
```
0
```

Converting columns to appropriate data types

In [79]:
```
df['original_price'] = df['original_price'].astype(float)
df['average_rating'] = df['average_rating'].astype(float)
df['total_ratings'] = df['total_ratings'].astype(int)
df['product_quantity'] = df['product_quantity'].astype(int)
df['special_price'] = df['special_price'].astype(float)
df['exclusive_price'] = df['exclusive_price'].astype(float)
df['sort_order'] = df['sort_order'].astype(int)
```

In [80]:
```
df.head()
```

Out[80]:

	product_id	product_name	brand	product_category	original_price	target_gender
0	279718	Kanso: Beige	{The Souled Store, the-souled-store-originals}	Men Low Top Sneakers	2399.0	1
1	272597	Kanso: Black	{The Souled Store, the-souled-store-originals}	Men Low Top Sneakers	2299.0	1
2	276043	TSS Originals: Earth	{The Souled Store, the-souled-store-originals}	Men High Top Sneakers	2999.0	1
3	279722	Black Panther: Warrior	{Marvel□, marvel-official-merchandise}	Men High Top Sneakers	3499.0	1
4	287843	Urban Blaze: Hawkins	{The Souled Store, the-souled-store-originals}	Men Low Top Sneakers	2699.0	1

In [81]:
```
df.reset_index(drop=True, inplace=True)
```

Saving the cleaned dataset to a new CSV file

In [82]:
```
df.to_csv('cleaned_api_data.csv', index=False)
```

Visualization:

In [83]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

Correlation Analysis

In [84]:
```
numeric_columns = ['original_price', 'special_price', 'exclusive_pr
```

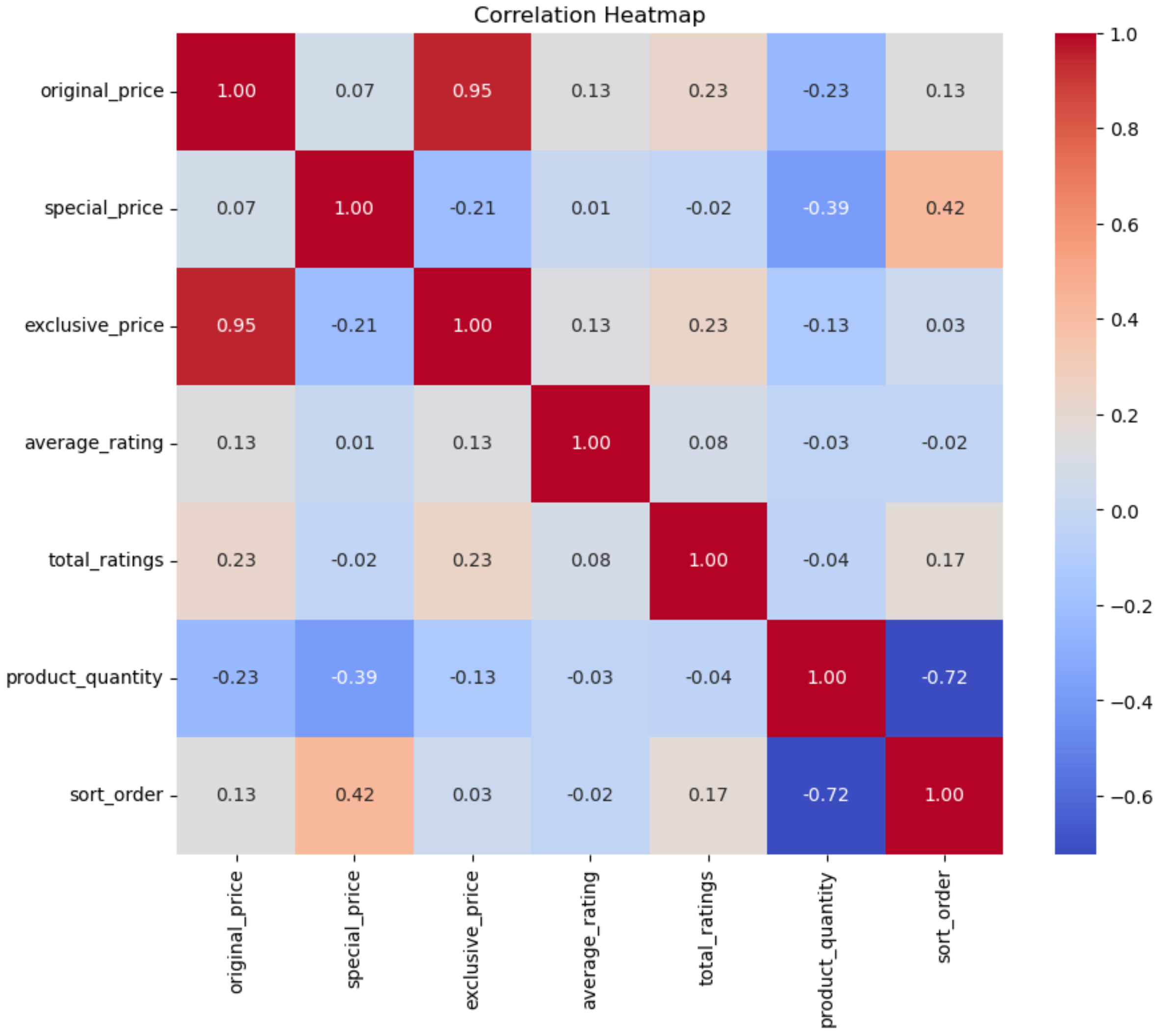
In [85]:
```
numeric_df = df[numeric_columns]
```

In [86]:
```
correlation_matrix = numeric_df.corr()
correlation_matrix
```

Out[86]:

	original_price	special_price	exclusive_price	average_rating	total_ratings
original_price	1.000000	0.070277	0.947131	0.130900	0.225741
special_price	0.070277	1.000000	-0.213521	0.006955	-0.015749
exclusive_price	0.947131	-0.213521	1.000000	0.125169	0.233362
average_rating	0.130900	0.006955	0.125169	1.000000	0.082663
total_ratings	0.225741	-0.015749	0.233362	0.082663	1.000000
product_quantity	-0.232288	-0.386363	-0.128813	-0.030478	-0.036232
sort_order	0.134947	0.422721	0.028627	-0.018217	0.173330

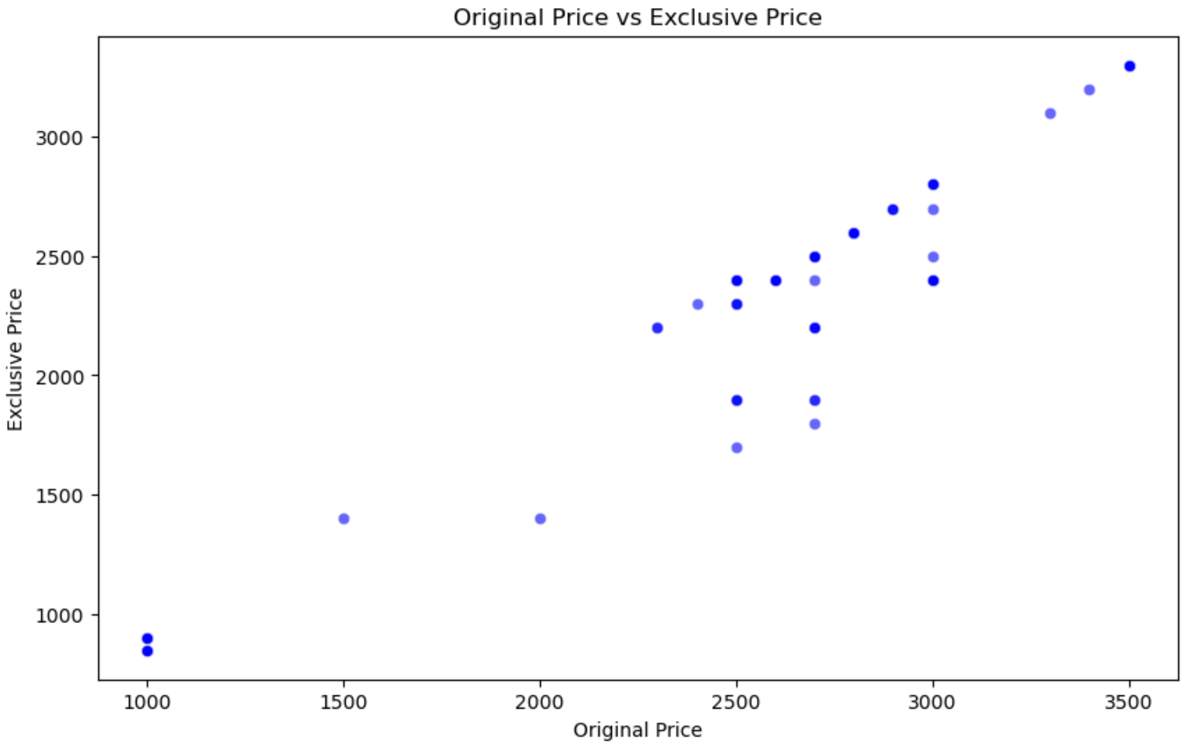
In [87]:
```
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.
plt.title('Correlation Heatmap')
plt.show()
```



- **Original Price vs Exclusive Price (0.95):** Strong positive correlation—exclusive prices are closely tied to original prices, likely as discounts.
- **Original Price vs Special Price (0.07):** No significant correlation—special prices aren't strongly influenced by original prices.
- **Special Price vs Exclusive Price (-0.21):** Weak negative correlation—higher discounts in one category tend to lower the other.
- **Average Rating vs Total Ratings (0.45):** Moderate positive correlation—highly rated products attract more reviews.
- **Average Rating vs Popularity Rank (-0.25):** Weak negative correlation—higher-rated products tend to have better ranks.
- **Product Quantity vs Popularity Rank (-0.15):** Weak negative correlation—higher stock products are slightly more popular.
- **Special Price vs Product Quantity (-0.39):** Moderate negative correlation—higher discounts correlate with lower stock, possibly due to faster sales.

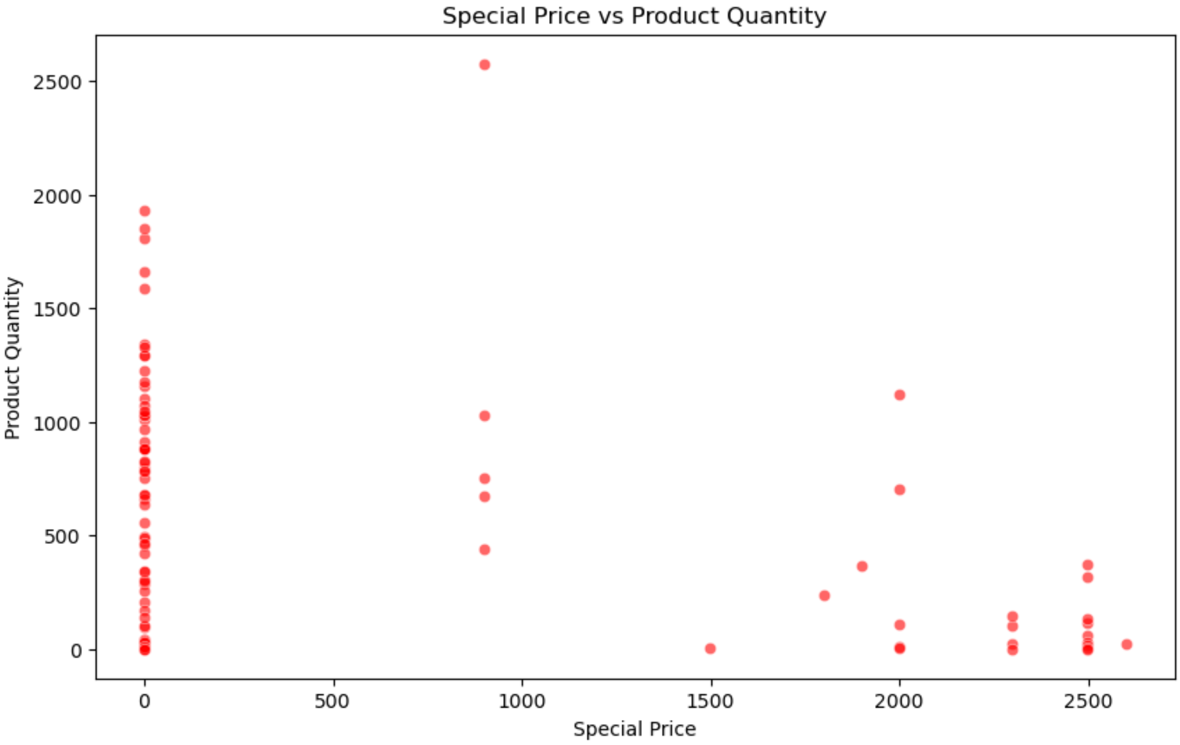
Pricing and discounts

```
In [88]: plt.figure(figsize=(10, 6))
         sns.scatterplot(x='original_price', y='exclusive_price', data=df, a
         plt.title('Original Price vs Exclusive Price')
         plt.xlabel('Original Price')
         plt.ylabel('Exclusive Price')
         plt.show()
```



- **Positive Correlation:** As the original price increases, the exclusive price generally increases too.
- **Linear Trend:** Most points align along a linear path, indicating consistent pricing logic for exclusive pricing relative to the original price.
- **Outliers:** A few points deviate from the trend, suggesting possible pricing anomalies or special discounts.

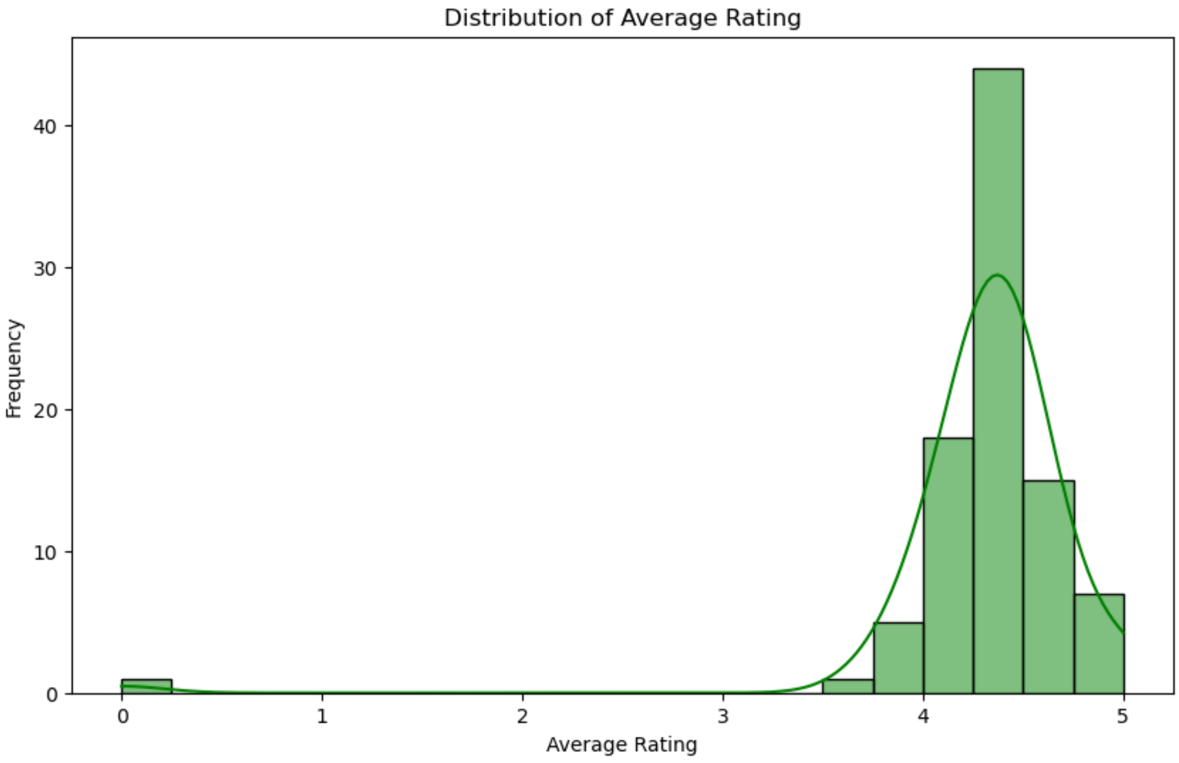
In [89]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='special_price', y='product_quantity', data=df, a
plt.title('Special Price vs Product Quantity')
plt.xlabel('Special Price')
plt.ylabel('Product Quantity')
plt.show()
```



- **Cluster at Low Prices:** Most products have low special prices, with varying quantities.
- **Sparse at Higher Prices:** Products with higher special prices are less frequent and generally have lower quantities.
- **Potential High-Value Outliers:** A few products maintain high quantities even at elevated special prices.

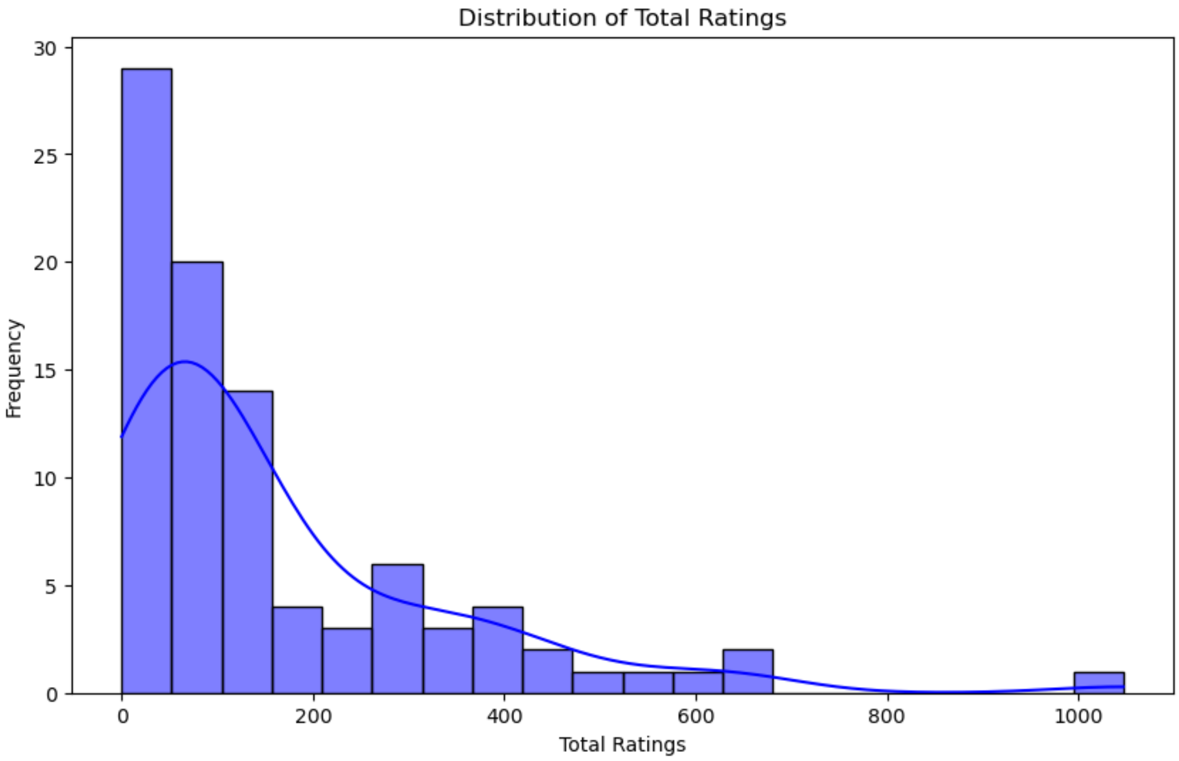
Ratings Distribution

In [90]:
```
plt.figure(figsize=(10, 6))
sns.histplot(df['average_rating'], bins=20, kde=True, color='green'
plt.title('Distribution of Average Rating')
plt.xlabel('Average Rating')
plt.ylabel('Frequency')
plt.show()
```



- **Right-Skewed Distribution:** The majority of ratings are clustered between 4 and 5, indicating a strong positive customer sentiment.
- **Few Low Ratings:** Very few products have average ratings below 3.
- **Peak around 4.5:** Most products tend to have ratings near 4.5, showing overall high product satisfaction.

In [91]:
```
plt.figure(figsize=(10, 6))
sns.histplot(df['total_ratings'], bins=20, kde=True, color='blue')
plt.title('Distribution of Total Ratings')
plt.xlabel('Total Ratings')
plt.ylabel('Frequency')
plt.show()
```

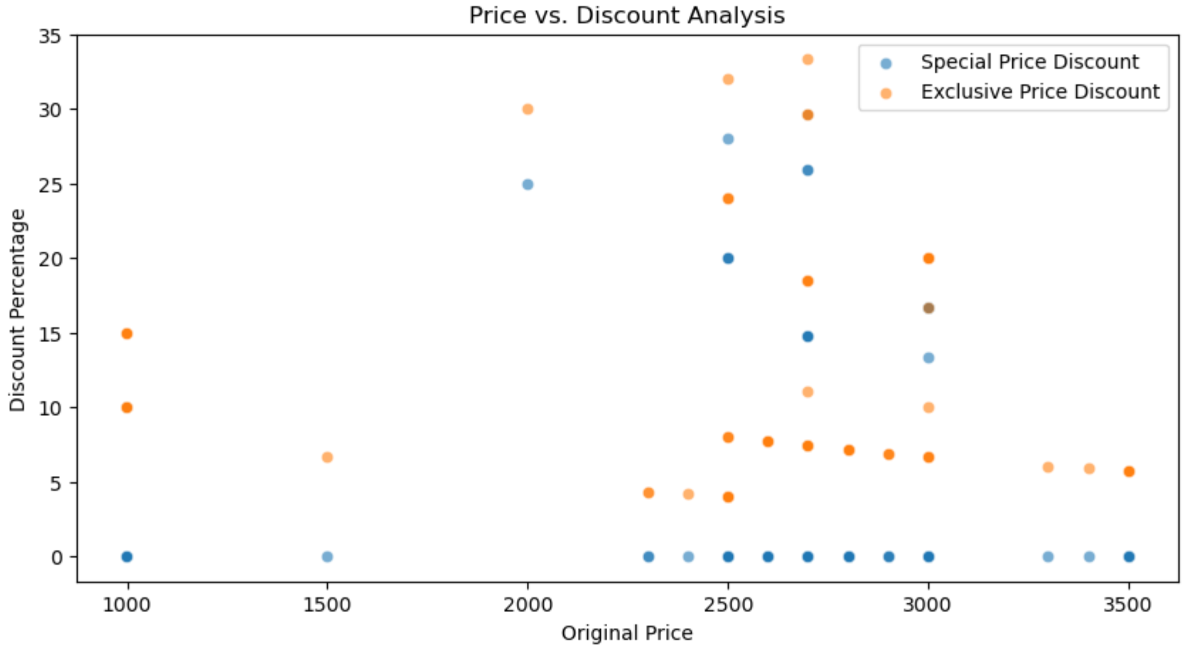


- The distribution of total ratings is **right-skewed**, indicating most products have low ratings, while a few have high ratings.
- The **KDE curve** confirms a declining trend as ratings increase.
- There are some outliers with exceptionally high ratings.

Discount Analysis

```
In [92]: df['discount_spl'] = df.apply(lambda row: ((row['original_price'] -
                                               if row['special_price'] > 0 else 0, a

         df['discount_exclusive'] = df.apply(lambda row: ((row['original_pri
                                                     if row['exclusive_price'] > 0 e

         # Correlation between price and discounts
         correlation_matrix = df[['original_price', 'discount_spl', 'discoun
         print("Correlation Matrix:")
         print(correlation_matrix)

         # Visualization
         plt.figure(figsize=(10, 5))
         sns.scatterplot(data=df, x='original_price', y='discount_spl', labe
         sns.scatterplot(data=df, x='original_price', y='discount_exclusive'
         plt.xlabel("Original Price")
         plt.ylabel("Discount Percentage")
         plt.title("Price vs. Discount Analysis")
         plt.legend()
         plt.show()
```

```
Correlation Matrix:
                    original_price  discount_spl  discount_exclusi
ve
original_price            1.000000     -0.028851          -0.0826
48
discount_spl             -0.028851      1.000000           0.9694
97
discount_exclusive       -0.082648      0.969497           1.0000
00
```



- **Weak correlation between original price and discounts** → Price has little influence on discount percentage.
- **Strong correlation (0.969) between special and exclusive discounts** → Both discounts often appear together.
- **Discounts cluster between 1000–3500 price range** → Targeted pricing strategy.
- **Exclusive discounts (orange) are more frequent than special discounts (blue).**
- **Higher discounts (up to 30%) exist, but many products have 0% discount.**

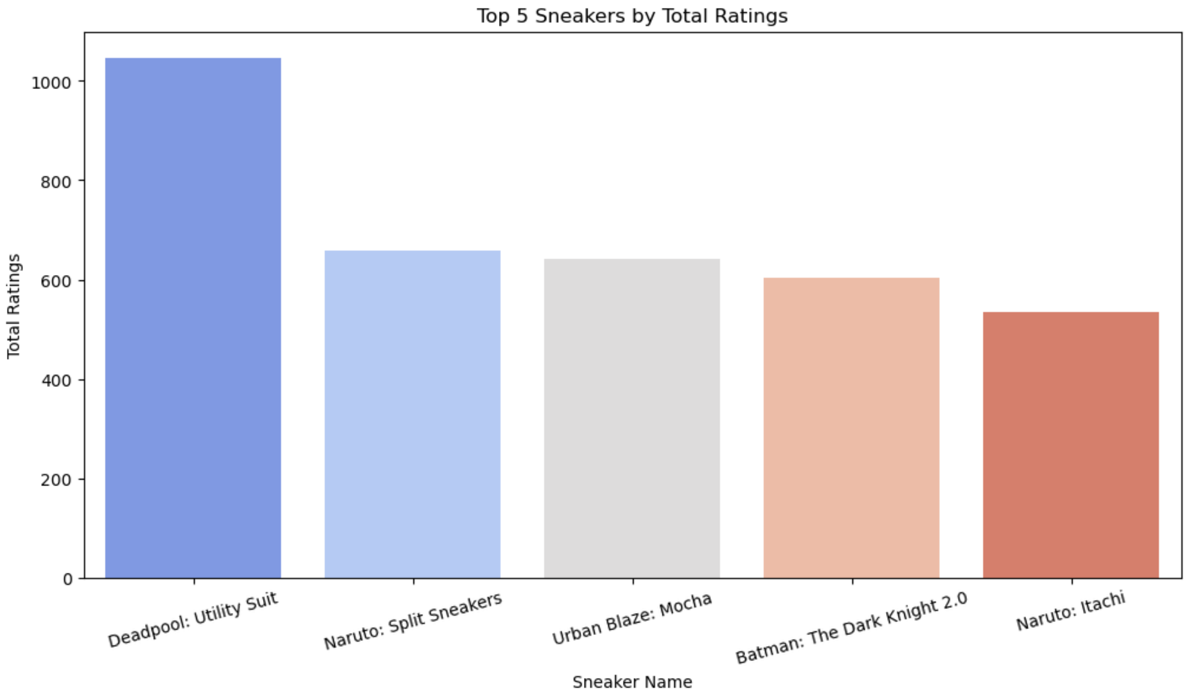
Top 5 sneakers by rating

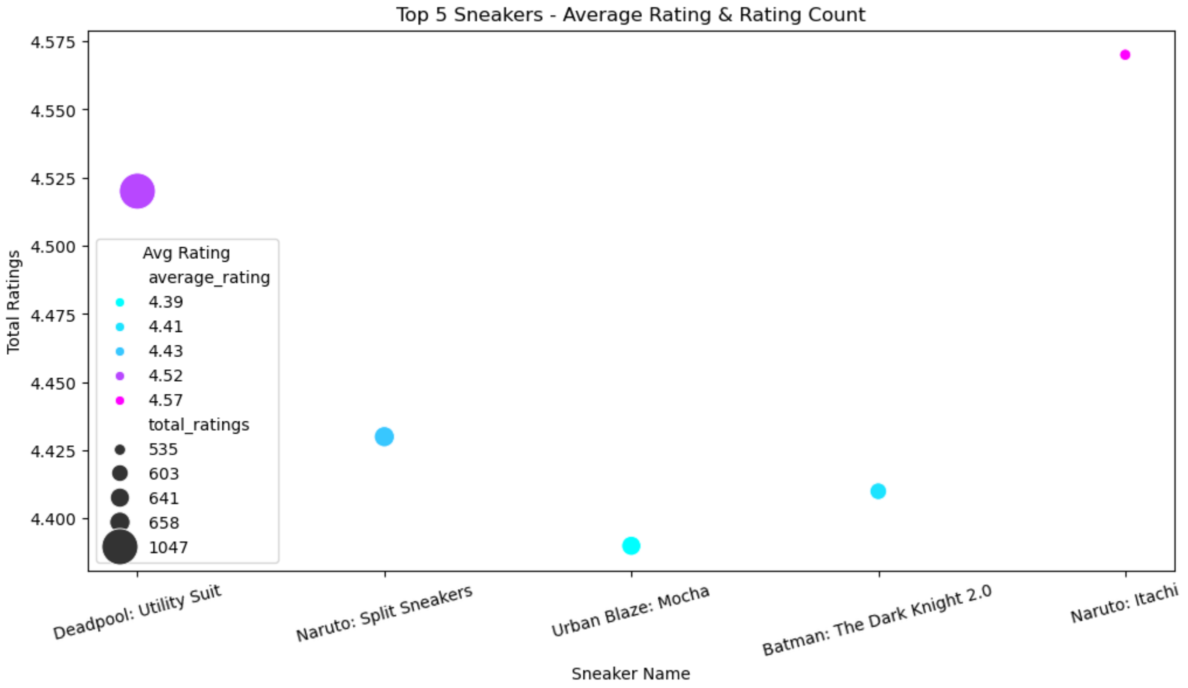
In [93]:

```
# Sort by total ratings and average ratings, select the top 5
top_sneakers = df.sort_values(by=['total_ratings', 'average_rating'

# Plotting
plt.figure(figsize=(12, 6))
sns.barplot(
    x='product_name',
    y='total_ratings',
    data=top_sneakers,
    palette='coolwarm'
)
plt.title("Top 5 Sneakers by Total Ratings")
plt.xlabel("Sneaker Name")
plt.ylabel("Total Ratings")
plt.xticks(rotation=15)
plt.show()

# Scatter plot for Average Ratings
plt.figure(figsize=(12, 6))
sns.scatterplot(
    x='product_name',
    y='average_rating',
    size='total_ratings',
    data=top_sneakers,
    sizes=(50, 500),
    hue='average_rating',
    palette='cool'
)
plt.title("Top 5 Sneakers - Average Rating & Rating Count")
plt.xlabel("Sneaker Name")
plt.ylabel("Total Ratings")
plt.xticks(rotation=15)
plt.legend(title='Avg Rating')
plt.show()
```

```
/var/folders/5h/_4my78qx1sg1gr1gwsnpk83w0000gn/T/ipykernel_32407/2
478523641.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will b
e removed in v0.14.0. Assign the `x` variable to `hue` and set `le
```

```
gend=False` for the same effect.

  sns.barplot(
```



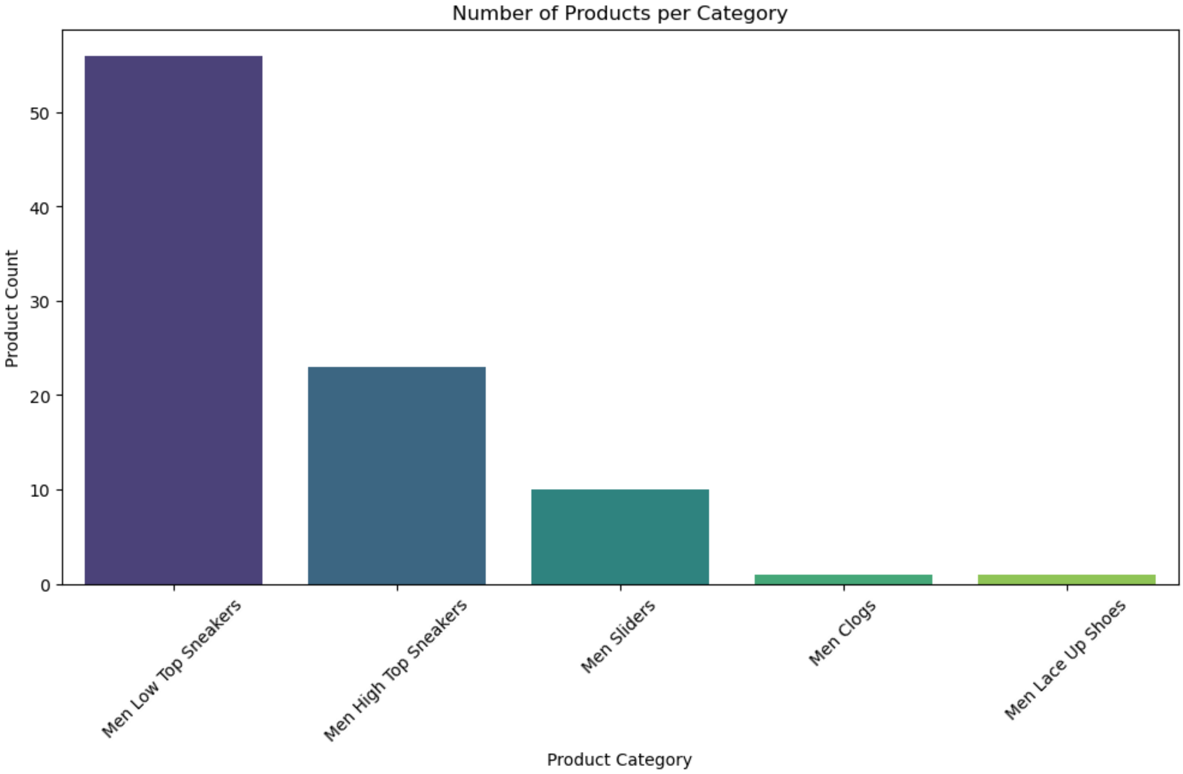


- **Top-Rated Sneaker:** "Deadpool: Utility Suit" has the highest total ratings, significantly leading the pack.
- **Rating Distribution:** The bubble chart shows that despite "Deadpool: Utility Suit" having the highest ratings count, its average rating is not the highest. "Naruto: Itachi" has the highest average rating (4.57), but fewer total ratings.
- **Balance Between Popularity & Quality:** "Deadpool: Utility Suit" is the most popular. "Naruto: Itachi" is the best-rated but has the least number of ratings.

Top products by category

In [94]:
```
plt.figure(figsize=(12, 6))
top_products_by_category = df.groupby('product_category')['product_
sns.barplot(x='product_category', y='product_name', data=top_produc
plt.title("Number of Products per Category")
plt.xlabel("Product Category")
plt.ylabel("Product Count")
plt.xticks(rotation=45)
plt.show()
```

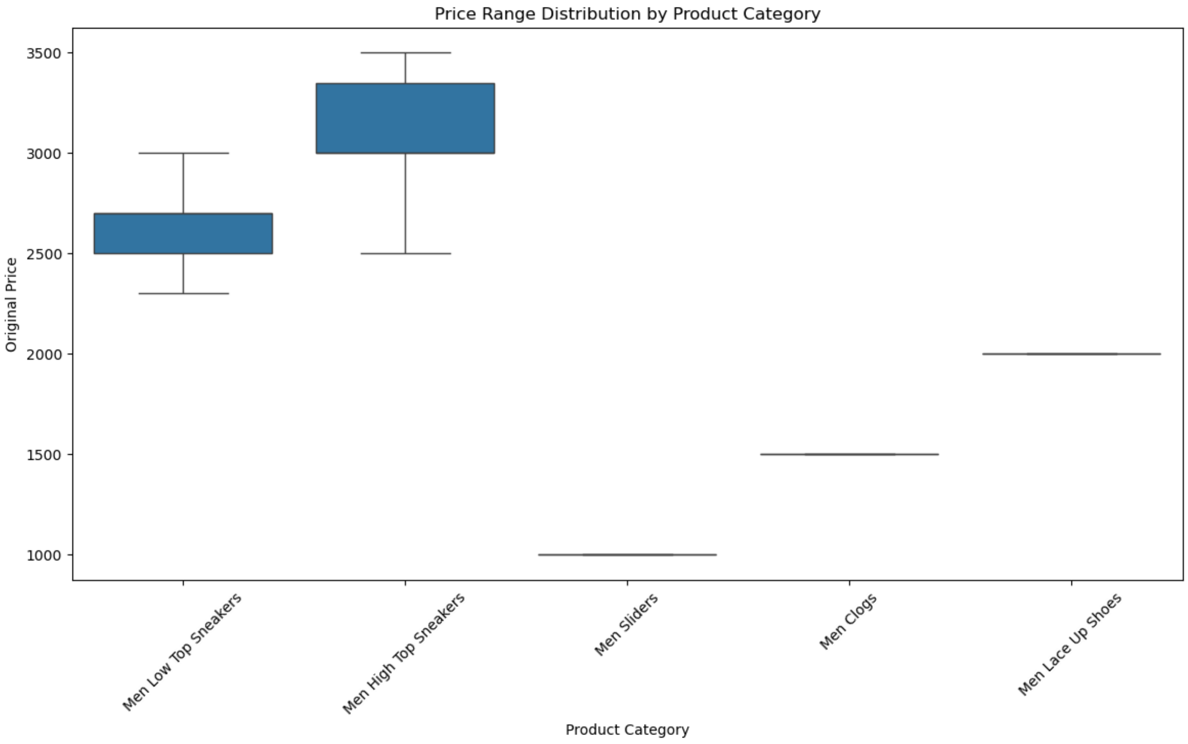
```
/var/folders/5h/_4my78qx1sg1gr1gwsnpk83w0000gn/T/ipykernel_32407/1
851725221.py:3: FutureWarning: 

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

  sns.barplot(x='product_category', y='product_name', data=top_pro
ducts_by_category, palette='viridis')
```



- **Men Low Top Sneakers** dominate the product count, significantly surpassing other categories.
- **Men High Top Sneakers and Men Sliders** follow, with moderate representation.
- Other categories, such as **Men Clogs and Men Lace Up Shoes**, have minimal offerings.

Price range analysis for products

In [95]:
```
plt.figure(figsize=(14, 7))
sns.boxplot(x='product_category', y='original_price', data=df)
plt.title("Price Range Distribution by Product Category")
plt.xlabel("Product Category")
plt.ylabel("Original Price")
plt.xticks(rotation=45)
plt.show()
```



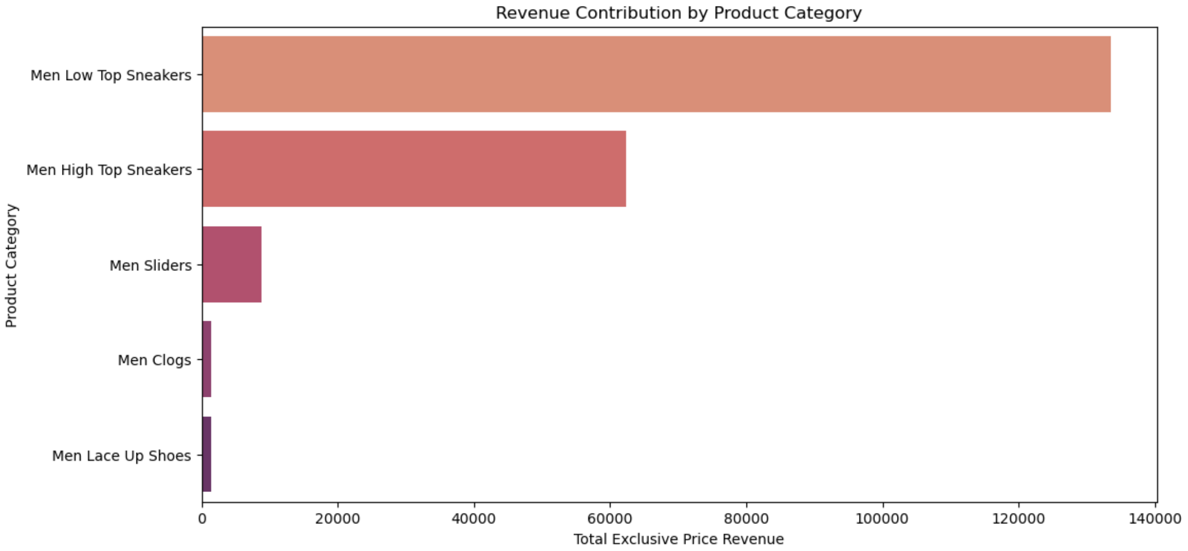
- **Men High Top Sneakers** have the highest price range, with most prices between ₹3,000 and ₹3,500.
- **Men Low Top Sneakers** are moderately priced, typically between ₹2,000 and ₹3,000.
- Other categories, such as **Men Sliders**, **Men Clogs**, and **Men Lace Up Shoes**, have lower and narrower price ranges.

Product category contribution to total revenue

In [96]:

```
revenue_per_category = df.groupby('product_category')['exclusive_pr
revenue_per_category = revenue_per_category.sort_values(by='exclusi

plt.figure(figsize=(12, 6))
sns.barplot(x='exclusive_price', y='product_category', data=revenue
plt.title("Revenue Contribution by Product Category")
plt.xlabel("Total Exclusive Price Revenue")
plt.ylabel("Product Category")
plt.show()
```

```
/var/folders/5h/_4my78qx1sg1gr1gwsnpk83w0000gn/T/ipykernel_32407/1
352900868.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will b
e removed in v0.14.0. Assign the `y` variable to `hue` and set `le
gend=False` for the same effect.

  sns.barplot(x='exclusive_price', y='product_category', data=reve
nue_per_category, palette='flare')
```



- **Men Low Top Sneakers** contribute the highest revenue.
- **Men High Top Sneakers** follow as the second-highest revenue generator.

Project improvement suggestions:

- Could have included female sneakers in the analysis, but it seems that some filters might have unintentionally applied during data scraping.
- Could have included customer reviews to analyze sentiment and correlate it with product success.
- Could have captured regional pricing variations to better understand market differences.

In [ ]: