BigBasket Analytics: Unlocking the Power of Data in E-Commerce

BigBasket Overview

- **BigBasket** is India's top online grocery retailer, providing a vast range of products such as fresh fruits and vegetables, meat, dairy, bakery items, and other household essentials. With a strong supply chain and partnerships with local farmers.
- BigBasket guarantees fresh and high-quality products.
- Customers can shop conveniently through its website or mobile app, with options for various payment methods and flexible delivery services.
- BigBasket also offers exclusive benefits through its BB Star membership, making it a preferred choice for online grocery shopping in India.

Problem Statement

- The rapid growth of e-commerce platforms like BigBasket has introduced challenges in managing large-scale operations, inventory optimization, and customer satisfaction. The objective is to analyze BigBasket's operational data to uncover trends, improve efficiency, and enhance decision-making for better business outcomes.

Goal

- To conduct an in-depth analysis of BigBasket's data to identify areas for process improvement, optimize inventory management, and ensure high customer satisfaction through data-driven strategies.

Dataset Summary

- **index**: A unique integer identifier for each observation.
- **product**: The name of the product (text format).
- **category**: The primary category or department the product belongs to (e.g., "Beauty & Hygiene," "Kitchen, Garden & Pets").
- **sub_category**: A more specific classification within the main category (e.g., "Hair Care," "Storage & Accessories").

- **brand**: The brand name of the product (text format).
- **sale_price**: The selling price of the product (numerical).
- **market_price**: The original or market-listed price of the product (numerical).
- **rating**: The customer satisfaction score for the product (numerical).
- **description**: Additional details or description of the product (text format).

Importing Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.io as pio
```

Step 1: Load DataSet

```
df = pd.read_csv("C:/Users/DELL-/Downloads/BigBasket Products.csv")
df
```

```
       index                                            product  \
0          1             Garlic Oil - Vegetarian Capsule 500 mg
1          2                           Water Bottle - Orange
2          3                    Brass Angle Deep - Plain, No.2
3          4  Cereal Flip Lid Container/Storage Jar - Assort...
4          5                 Creme Soft Soap - For Hands & Body
...      ...                                              ...
27550  27551         Wottagirl! Perfume Spray - Heaven, Classic
27551  27552                                         Rosemary
27552  27553                     Peri-Peri Sweet Potato Chips
27553  27554                        Green Tea - Pure Original
27554  27555                    United Dreams Go Far Deodorant

                     category              sub_category  \
0           Beauty & Hygiene                 Hair Care
1      Kitchen, Garden & Pets     Storage & Accessories
2        Cleaning & Household              Pooja Needs
3        Cleaning & Household      Bins & Bathroom Ware
4           Beauty & Hygiene          Bath & Hand Wash
...                      ...                       ...
27550       Beauty & Hygiene         Fragrances & Deos
27551   Gourmet & World Food    Cooking & Baking Needs
27552   Gourmet & World Food  Snacks, Dry Fruits, Nuts
27553              Beverages                       Tea
27554       Beauty & Hygiene            Men's Grooming

                          brand  sale_price  market_price  \
0              Sri Sri Ayurveda       220.00         220.0
1                     Mastercook      180.00         180.0
2                           Trm       119.00         250.0
```

```
3                          Nakoda      149.00         176.0
4                           Nivea      162.00         162.0
...                           ...         ...           ...
27550                     Layerr      199.20         249.0
27551                   Puramate       67.50          75.0
27552                     FabBox      200.00         200.0
27553                     Tetley      396.00         495.0
27554  United Colors Of Benetton       214.53         390.0

                            type  rating  \
0              Hair Oil & Serum     4.1
1        Water & Fridge Bottles     2.3
2               Lamp & Lamp Oil     3.4
3      Laundry, Storage Baskets     3.7
4          Bathing Bars & Soaps     4.4
...                         ...     ...
27550                  Perfume     3.9
27551  Herbs, Seasonings & Rubs     4.0
27552            Nachos & Chips     3.8
27553                 Tea Bags     4.2
27554          Men's Deodorants     4.5

                                                 description
0      This Product contains Garlic Oil that is known...
1      Each product is microwave safe (without lid), ...
2      A perfect gift for all occasions, be it your m...
3      Multipurpose container with an attractive desi...
4      Nivea Creme Soft Soap gives your skin the best...
...                                                  ...
27550  Layerr brings you Wottagirl Classic fragrant b...
27551  Puramate rosemary is enough to transform a dis...
27552  We have taken the richness of Sweet Potatoes (...
27553  Tetley Green Tea with its refreshing pure, ori...
27554  The new mens fragrance from the United Dreams ...

[27555 rows x 10 columns]
```

Step 2:look for first 12 rows.

```
df.head(10)
```

```
   index                                            product  \
0      1             Garlic Oil - Vegetarian Capsule 500 mg
1      2                           Water Bottle - Orange
2      3                    Brass Angle Deep - Plain, No.2
3      4  Cereal Flip Lid Container/Storage Jar - Assort...
4      5                 Creme Soft Soap - For Hands & Body
5      6                  Germ - Removal Multipurpose Wipes
6      7                                      Multani Mati
7      8                  Hand Sanitizer - 70% Alcohol Base
```

```
8      9  Biotin & Collagen Volumizing Hair Shampoo + Bi...
9     10               Scrub Pad - Anti- Bacterial, Regular

                  category            sub_category
brand  \
0        Beauty & Hygiene               Hair Care  Sri Sri Ayurveda

1  Kitchen, Garden & Pets   Storage & Accessories         Mastercook

2    Cleaning & Household             Pooja Needs                Trm

3    Cleaning & Household    Bins & Bathroom Ware              Nakoda

4        Beauty & Hygiene        Bath & Hand Wash               Nivea

5    Cleaning & Household    All Purpose Cleaners     Nature Protect

6        Beauty & Hygiene               Skin Care           Satinance

7        Beauty & Hygiene        Bath & Hand Wash             Bionova

8        Beauty & Hygiene               Hair Care          StBotanica

9    Cleaning & Household  Mops, Brushes & Scrubs       Scotch brite


   sale_price  market_price                           type  rating  \
0       220.0         220.0              Hair Oil & Serum     4.1
1       180.0         180.0        Water & Fridge Bottles     2.3
2       119.0         250.0                Lamp & Lamp Oil     3.4
3       149.0         176.0       Laundry, Storage Baskets     3.7
4       162.0         162.0           Bathing Bars & Soaps     4.4
5       169.0         199.0  Disinfectant Spray & Cleaners     3.3
6        58.0          58.0                      Face Care     3.6
7       250.0         250.0         Hand Wash & Sanitizers     4.0
8      1098.0        1098.0          Shampoo & Conditioner     3.5
9        20.0          20.0       Utensil Scrub-Pad, Glove     4.3

                                        description
0  This Product contains Garlic Oil that is known...
1  Each product is microwave safe (without lid), ...
2  A perfect gift for all occasions, be it your m...
3  Multipurpose container with an attractive desi...
4  Nivea Creme Soft Soap gives your skin the best...
5  Stay protected from contamination with Multipu...
6  Satinance multani matti is an excellent skin t...
7  70%Alcohol based is gentle of hand leaves skin...
8  An exclusive blend with Vitamin B7 Biotin, Hyd...
9  Scotch Brite Anti- Bacterial Scrub Pad thoroug...
```

Step 3: Get Description of the data in the DataFrame.

```
df.describe()
```

```
              index     sale_price  market_price        rating
count  27555.00000   27549.000000  27555.000000  18919.000000
mean   13778.00000     334.648391    382.056664      3.943295
std     7954.58767    1202.102113    581.730717      0.739217
min        1.00000       2.450000      3.000000      1.000000
25%     6889.50000      95.000000    100.000000      3.700000
50%    13778.00000     190.320000    220.000000      4.100000
75%    20666.50000     359.000000    425.000000      4.300000
max    27555.00000  112475.000000  12500.000000      5.000000
```

Step 4: Find Information about the DataFrame

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27555 entries, 0 to 27554
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   index         27555 non-null  int64  
 1   product       27554 non-null  object 
 2   category      27555 non-null  object 
 3   sub_category  27555 non-null  object 
 4   brand         27554 non-null  object 
 5   sale_price    27549 non-null  float64
 6   market_price  27555 non-null  float64
 7   type          27555 non-null  object 
 8   rating        18919 non-null  float64
 9   description   27440 non-null  object 
dtypes: float64(3), int64(1), object(6)
memory usage: 2.1+ MB
```

Step 5: Find out Top & least sold products

```
#top product
df[['product','category','sub_category']].value_counts().head(1)
```

```
product                         category                   sub_category

Turmeric Powder/Arisina Pudi  Foodgrains, Oil & Masala  Masalas &
Spices    25
Name: count, dtype: int64
```

```
#least sold product
df[['product','category','sub_category']].value_counts().tail(1)
```

```
product                                category              sub_category

1 To 1  Baking Flour - Gluten Free  Gourmet & World Food  Cooking &
Baking Needs    1
Name: count, dtype: int64
```

Step 6: Measuring discount on a certain item.

```
discount=(df['market_price']-df['sale_price'])/df['market_price']*100
discount
```

```
0          0.000000
1          0.000000
2         52.400000
3         15.340909
4          0.000000
             ...
27550     20.000000
27551     10.000000
27552      0.000000
27553     20.000000
27554     44.992308
Length: 27555, dtype: float64
```

Step 7: Find out the Missing Values from the Dataset.

```
missing_values=df.isnull().sum()
missing_values
```

```
index               0
product             1
category            0
sub_category        0
brand               1
sale_price          6
market_price        0
type                0
rating           8636
description       115
dtype: int64
```

```
df[df['product'].isna()]
```

```
      product   category sub_category          brand  sale_price  \
index
14364     NaN  Beverages       Coffee  Cothas Coffee       200.0

       market_price           type  rating  \
index
14364         240.0  Ground Coffee     4.2
```

```
                                                  description
index
14364  Cothas Specialty Blend Coffee and Chicory incl...
```

```
df[df['brand'].isna()]
```

```
                      product               category
sub_category  \
index

9766   Food Package - Medium  Cleaning & Household  Disposables,
Garbage Bag

      brand  sale_price  market_price                          type
rating  \
index

9766    NaN        50.0          50.0  Aluminium Foil, Clingwrap
3.956482

      description
index
9766          NaN
```

Find out the percentage of missing values of dataset

```
total_missing_values=missing_values.sum()
total_cell=np.prod(df.shape)
percent_missing = (total_missing_values/total_cell) * 100
print("Percentage: {:.2f}%".format(percent_missing))
```

```
Percentage: 0.42%
```

Cleaning missing values

```
df2=pd.DataFrame(df)

df2.loc[df2['product'].isna(),'product']='Unknown'
df2['product']
```

```
0                    Garlic Oil - Vegetarian Capsule 500 mg
1                                    Water Bottle - Orange
2                           Brass Angle Deep - Plain, No.2
3        Cereal Flip Lid Container/Storage Jar - Assort...
4                      Creme Soft Soap - For Hands & Body
                                ...
27550           Wottagirl! Perfume Spray - Heaven, Classic
27551                                             Rosemary
27552                          Peri-Peri Sweet Potato Chips
27553                             Green Tea - Pure Original
```

```
27554                         United Dreams Go Far Deodorant
Name: product, Length: 27555, dtype: object
```

```
df2.loc[df2['description'].isna(),'description']='Unknown'
df2['description']
```

```
0        This Product contains Garlic Oil that is known...
1        Each product is microwave safe (without lid), ...
2        A perfect gift for all occasions, be it your m...
3        Multipurpose container with an attractive desi...
4        Nivea Creme Soft Soap gives your skin the best...
                               ...
27550    Layerr brings you Wottagirl Classic fragrant b...
27551    Puramate rosemary is enough to transform a dis...
27552    We have taken the richness of Sweet Potatoes (...
27553    Tetley Green Tea with its refreshing pure, ori...
27554    The new mens fragrance from the United Dreams ...
Name: description, Length: 27555, dtype: object
```

```
df2.loc[df2['brand'].isna(),'brand']='Unknown'
df2['brand']
```

```
0                  Sri Sri Ayurveda
1                        Mastercook
2                               Trm
3                            Nakoda
4                             Nivea
                    ...
27550                        Layerr
27551                      Puramate
27552                        FabBox
27553                        Tetley
27554    United Colors Of Benetton
Name: brand, Length: 27555, dtype: object
```

```
df2['rating'].median()
```

```
np.float64(4.1)
```

```
df2['sale_price'] =
np.where(df2['sale_price'].isna(),df2['sale_price'].median(),
df2['sale_price'])
df2['sale_price']
```

```
0        220.00
1        180.00
2        119.00
3        149.00
4        162.00
          ...
27550    199.20
```

```
27551     67.50
27552    200.00
27553    396.00
27554    214.53
Name: sale_price, Length: 27555, dtype: float64
```

```
df2['rating'] = np.where(df2['rating'].isna(),df2['rating'].median(),
df2['rating'])
df2['rating']
```

```
0        4.1
1        2.3
2        3.4
3        3.7
4        4.4
        ...
27550    3.9
27551    4.0
27552    3.8
27553    4.2
27554    4.5
Name: rating, Length: 27555, dtype: float64
```

Data after cleaning

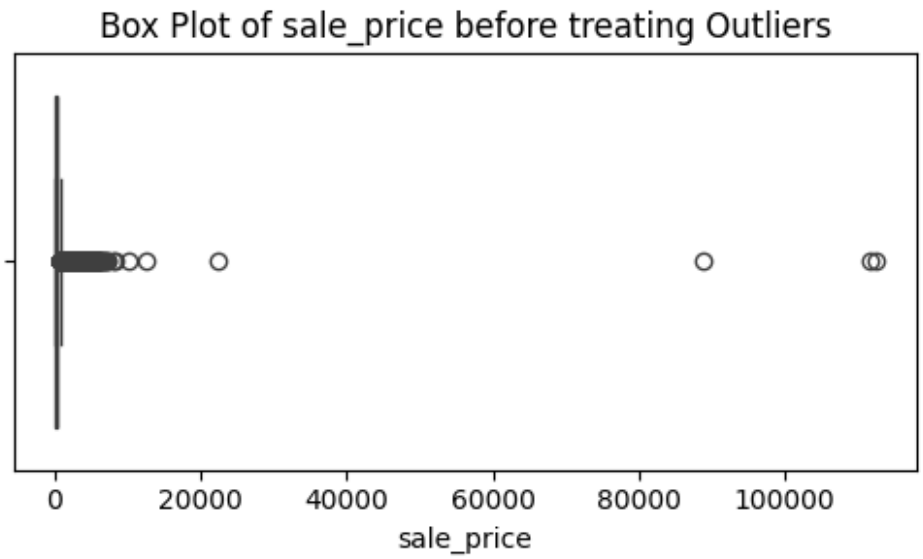
```
df2.isna().sum()
```

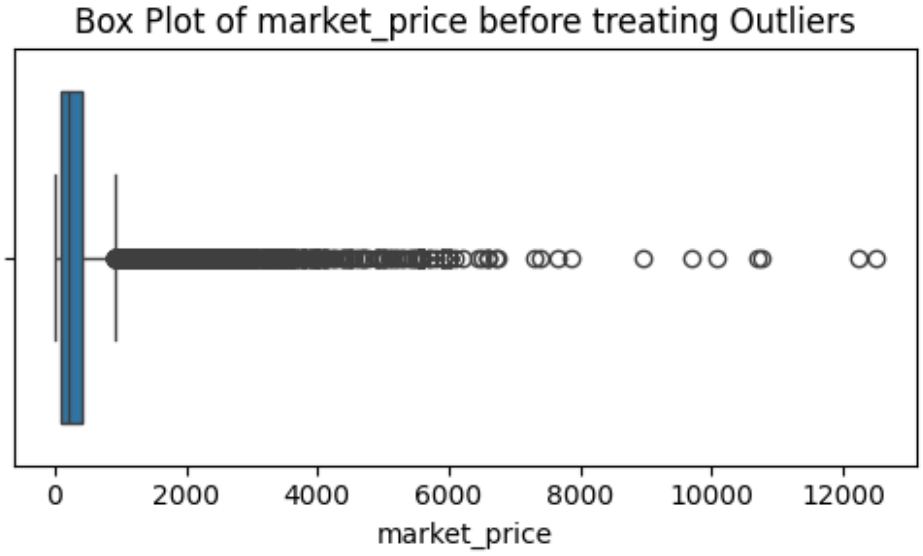
```
index           0
product         0
category        0
sub_category    0
brand           0
sale_price      0
market_price    0
type            0
rating          0
description     0
dtype: int64
```

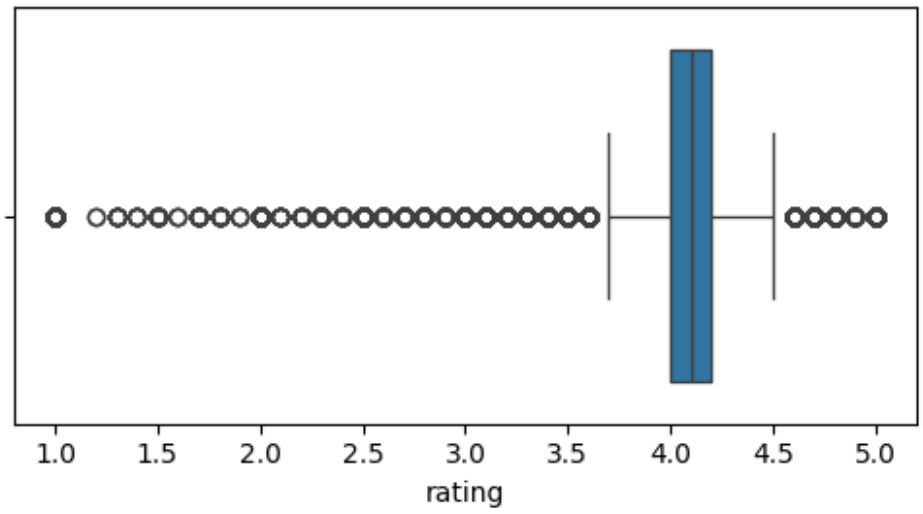
Step 8: Find out the outliers from the dataset according to the columns

```
# List of specific columns to plot
columns_to_plot = ['sale_price', 'market_price', 'rating']

# Plot box plots for the selected columns
plt.figure(figsize=(5, len(columns_to_plot) * 3))  # Adjust figure
size
for i, column in enumerate(columns_to_plot):
    plt.subplot(len(columns_to_plot), 1, i + 1)
```

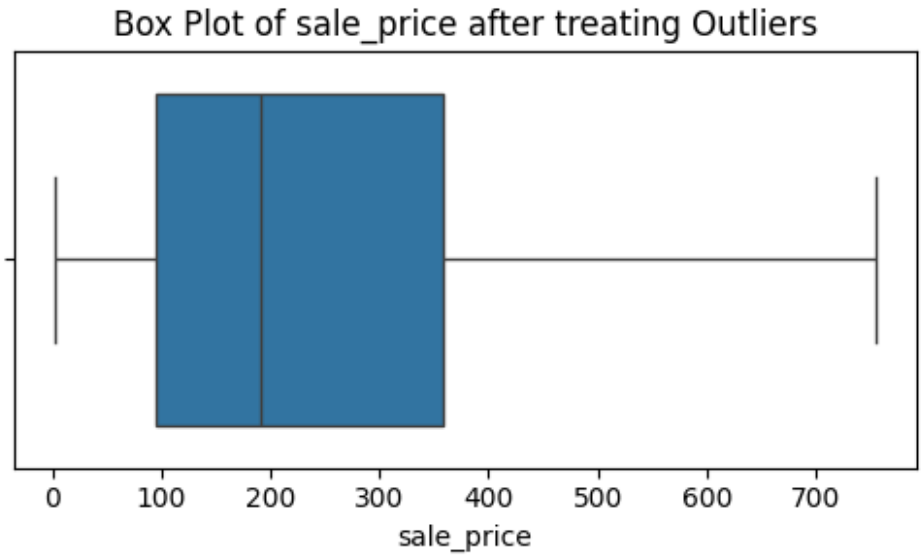
```
    sns.boxplot(x=df2[column])
    plt.title(f'Box Plot of {column} before treating Outliers')
    plt.tight_layout()

plt.show()
```

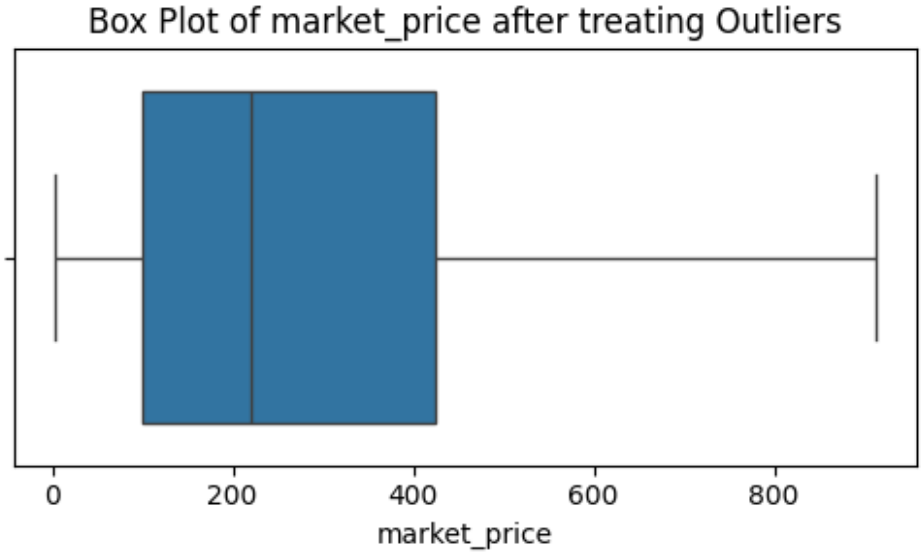

Box Plot of sale_price before treating Outliers
0
20000
40000
60000
80000
100000
sale_price

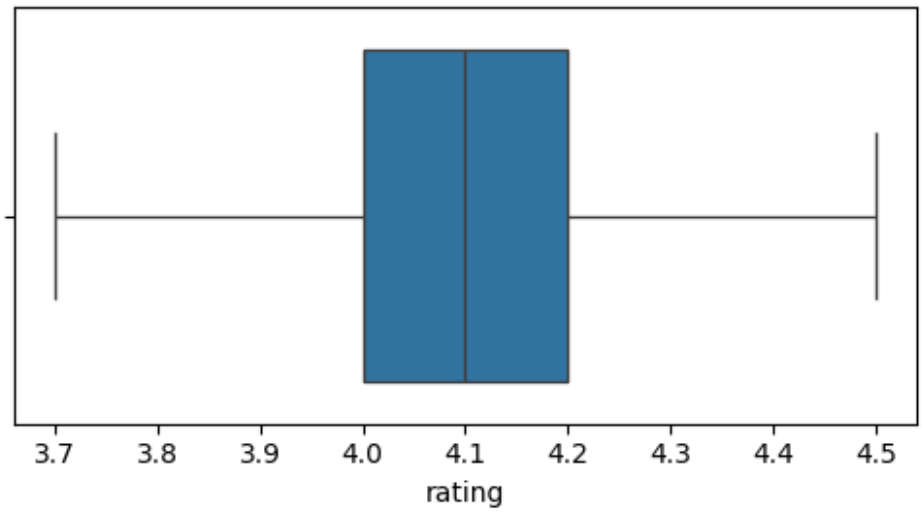

Box Plot of market_price before treating Outliers
0
2000
4000
6000
8000
10000
12000
market_price


Box Plot of rating before treating Outliers
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
rating

```
def replace_outliers(data, column):
    Q1 = data[column].quantile(0.25)
    Q3 = data[column].quantile(0.75)

    IQR = Q3 - Q1

    lower_bound = max(Q1 - 1.5 * IQR,0)
    upper_bound = Q3 + 1.5 * IQR

    # Replace values below lower bound with lower bound and values
above upper bound with upper bound
    data[column] = data[column].apply(lambda x: lower_bound if x <
lower_bound else (upper_bound if x > upper_bound else x))

    return data

data = replace_outliers(df2, 'sale_price')
data = replace_outliers(df2, 'market_price')
data = replace_outliers(df2, 'rating')

# List of specific columns to plot
columns_to_plot = ['sale_price', 'market_price', 'rating']

# Plot box plots for the selected columns
plt.figure(figsize=(5, len(columns_to_plot) * 3))  # Adjust figure
size
for i, column in enumerate(columns_to_plot):
    plt.subplot(len(columns_to_plot), 1, i + 1)
    sns.boxplot(x=df2[column])
    plt.title(f'Box Plot of {column} after treating Outliers')
    plt.tight_layout()

plt.show()
```

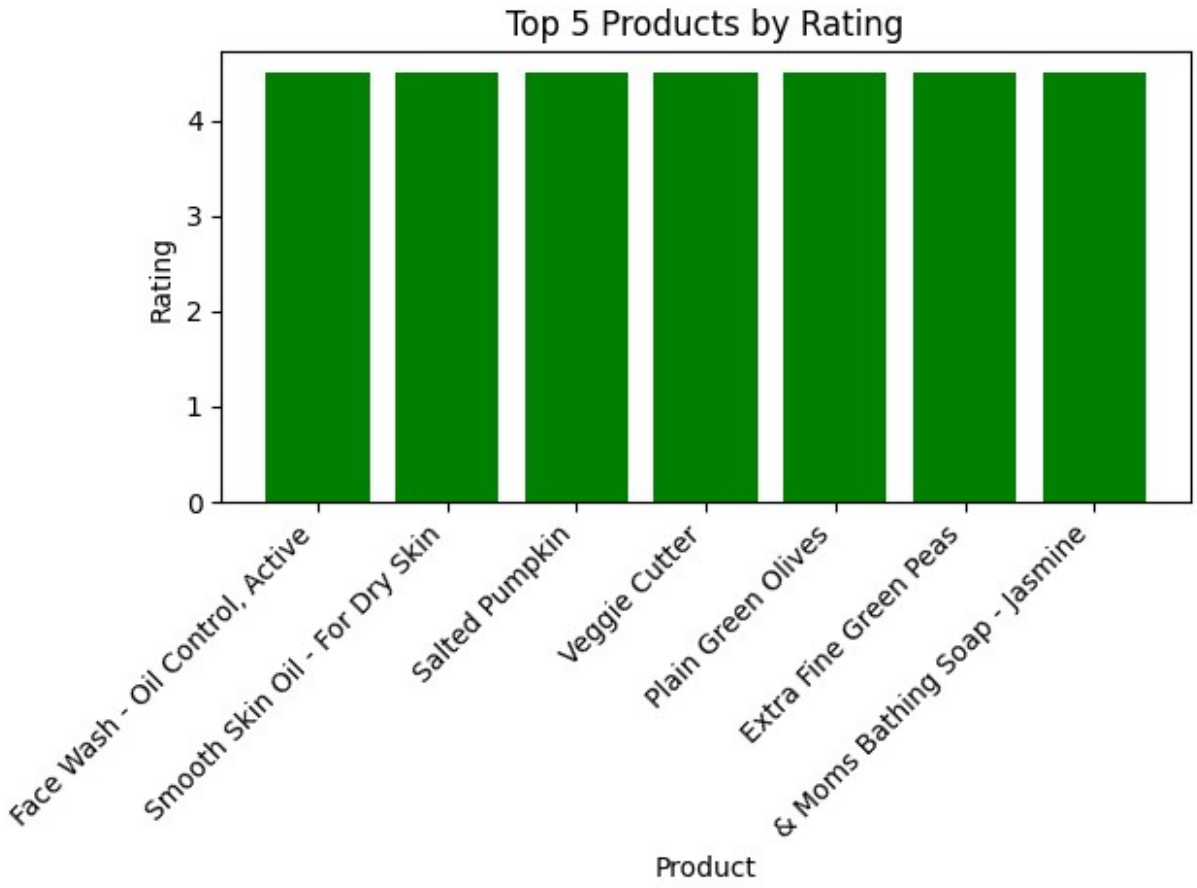

Box Plot of sale_price after treating Outliers
0
100
200
300
400
500
600
700
sale_price


Box Plot of market_price after treating Outliers
0
200
400
600
800
market_price


Box Plot of rating after treating Outliers
3.7
3.8
3.9
4.0
4.1
4.2
4.3
4.4
4.5
rating

Step 9: Create Plots or visualizations.

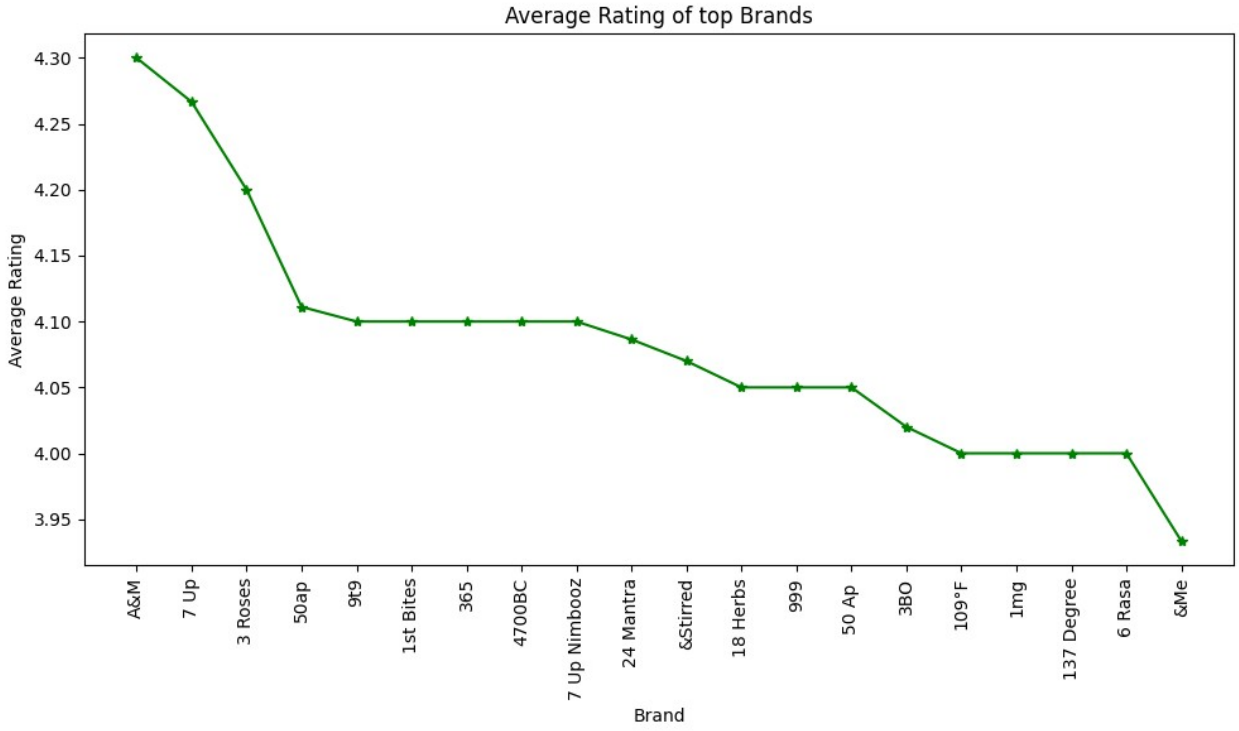
```
top_5_products = df2.nlargest(7, 'rating')[['product', 'rating']]
plt.bar(top_5_products['product'], top_5_products['rating'],
color='green')
plt.xlabel('Product')
plt.ylabel('Rating')
plt.title('Top 5 Products by Rating')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



Brand Ratings

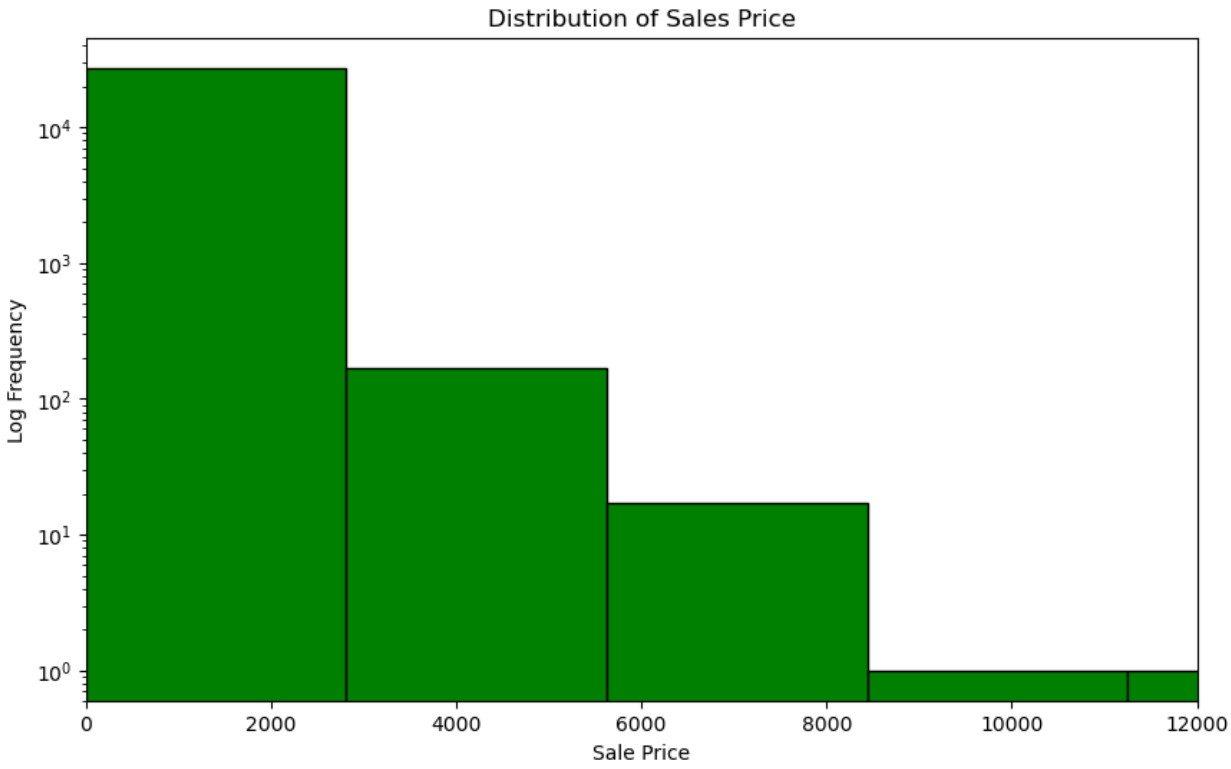
```
brand_ratings = df2.groupby('brand')
['rating'].mean().reset_index().head(20)
brand_ratings_sorted = brand_ratings.sort_values(by='rating',
ascending=False)
plt.figure(figsize=(10,6))
plt.plot(brand_ratings_sorted['brand'],
brand_ratings_sorted['rating'], marker='*', color='green')
plt.xticks(rotation=90)
```

```
plt.xlabel('Brand')
plt.ylabel('Average Rating')
plt.title('Average Rating of top Brands')
plt.tight_layout()
plt.show()
```



Price Distribution (histogram)

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

# Increase the number of bins, e.g., to 40
plt.hist(df2['sale_price'], bins=40, color='green', edgecolor='black')

plt.yscale('log')  # Use logarithmic scale for the y-axis
plt.title('Distribution of Sales Price')
plt.xlabel('Sale Price')
plt.ylabel('Log Frequency')

plt.xlim(0, 12000)
plt.xticks(range(0, 13000, 2000))

plt.show()
```

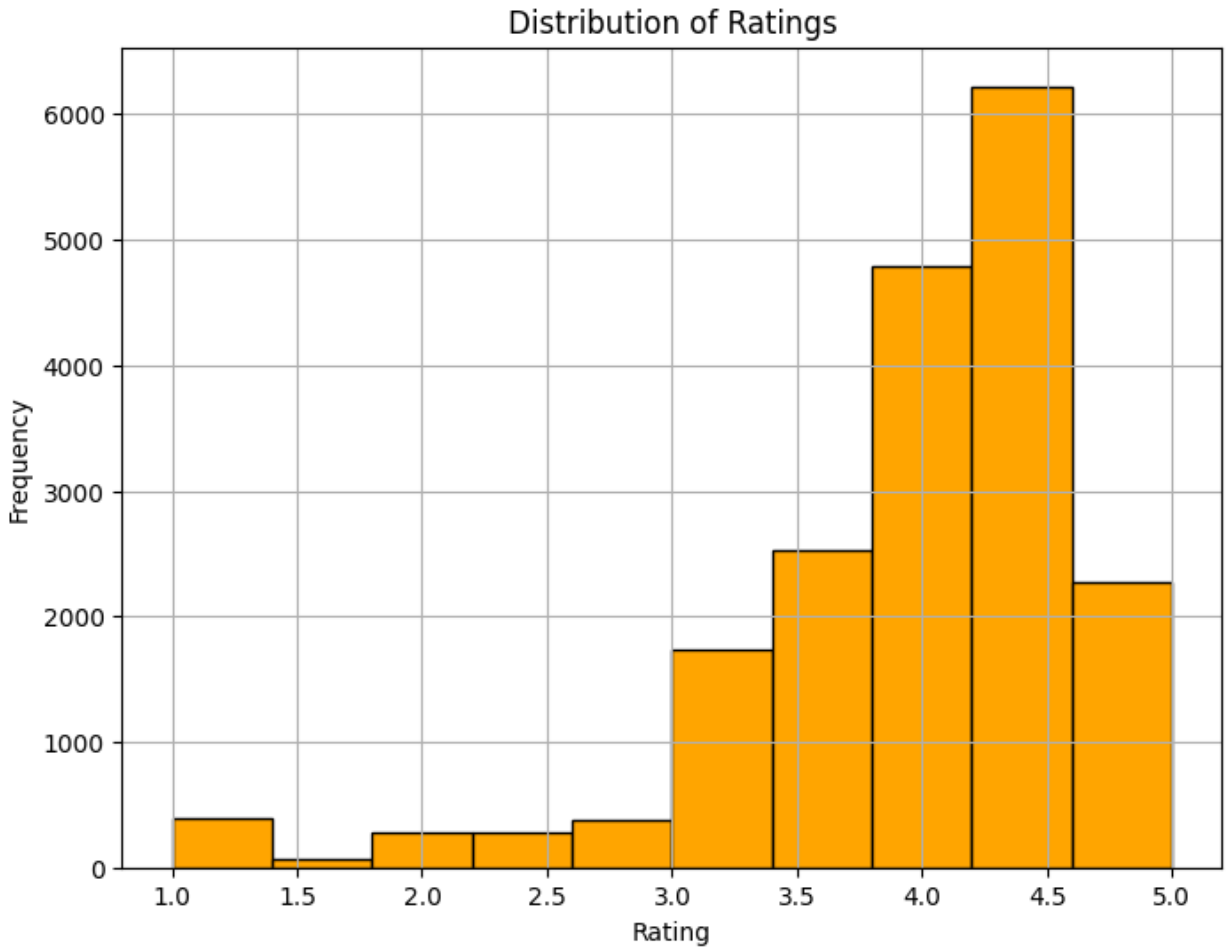


Scatter plot of sale price vs rating

```
plt.figure(figsize=(8,6))
plt.scatter(df2['sale_price'], df2['rating'], color='blue', alpha=0.6)
plt.title('Sale Price vs Rating')
plt.xlabel('Sale Price')
plt.ylabel('Rating')
plt.grid(True)
plt.show()
```

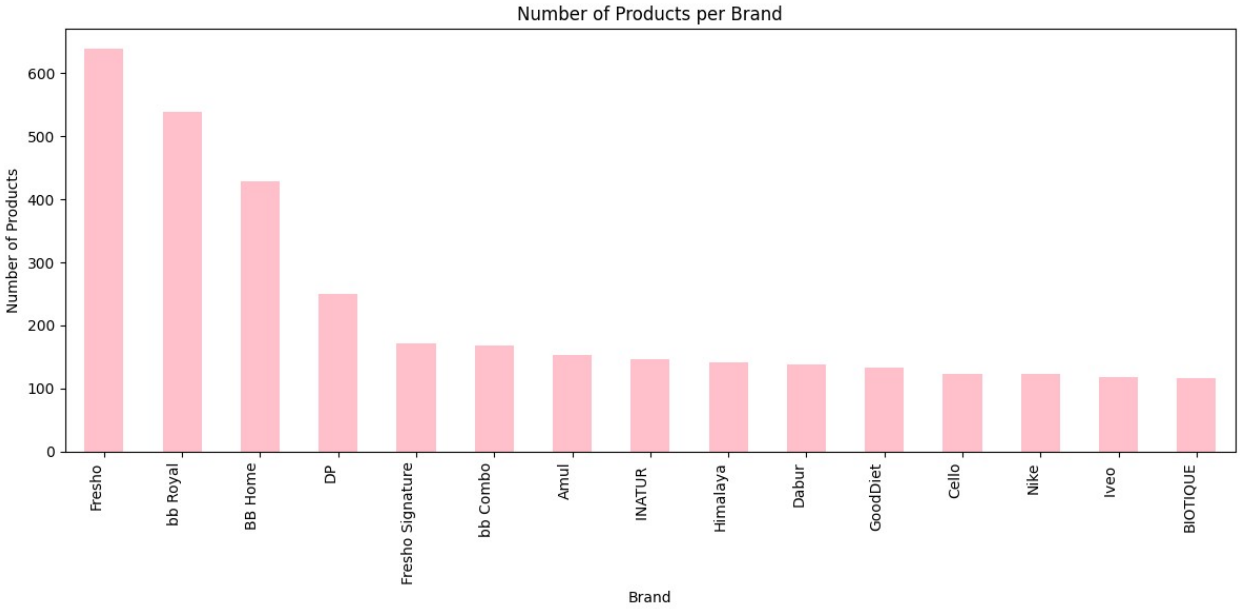
Plot histogram of ratings

```
plt.figure(figsize=(8,6))
plt.hist(df2['rating'], bins=10, color='orange', edgecolor='black')
plt.title('Distribution of Ratings')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```



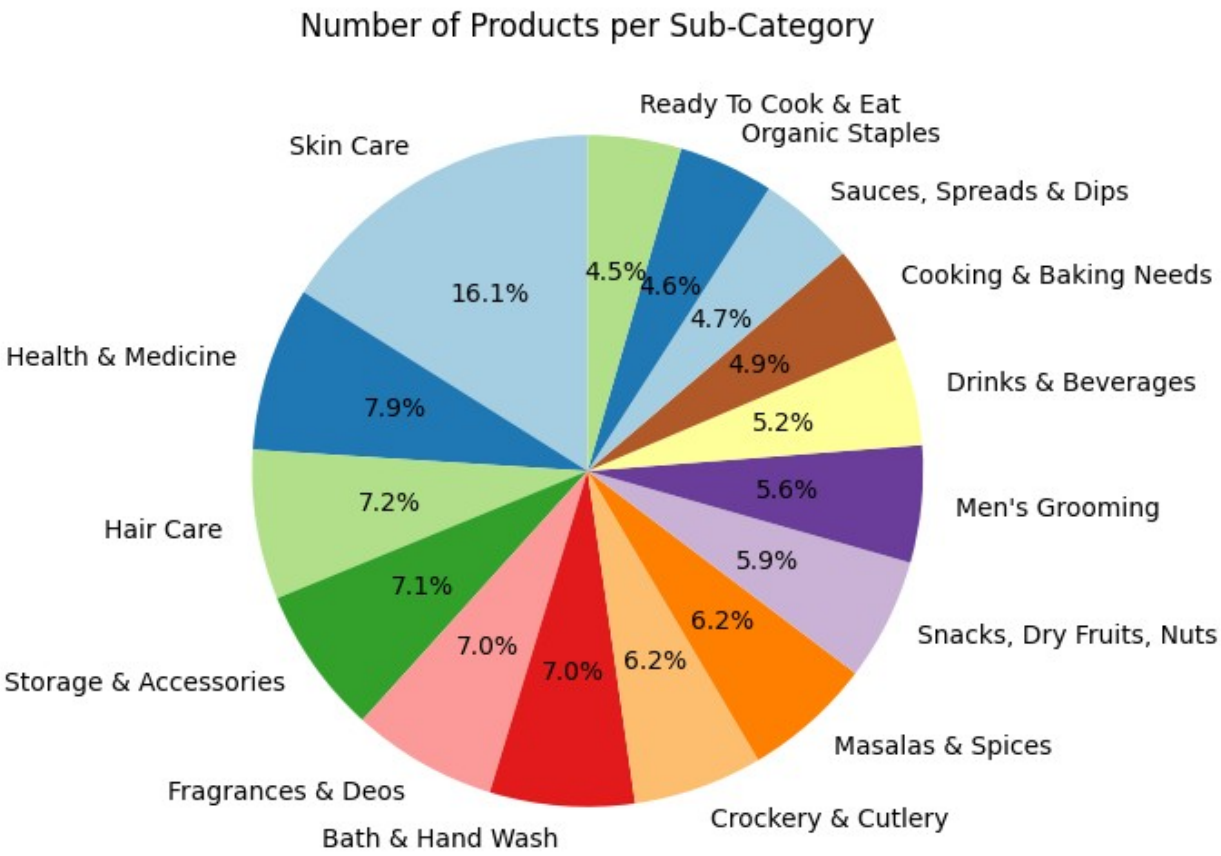
Count the number of products for top 15 brands

```
brand_counts = df2['brand'].value_counts().head(15)
# Plotting the number of products per brand
plt.figure(figsize=(12,6))
brand_counts.plot(kind='bar', color='pink')
plt.title('Number of Products per Brand')
plt.xlabel('Brand')
plt.ylabel('Number of Products')
plt.xticks(rotation=90, ha='right')
plt.tight_layout()
plt.show()
```



Count the number of products for top 15 sub-category

```
sub_category_counts = df2['sub_category'].value_counts().head(15)

# Plotting the number of products per sub-category as a pie chart
plt.figure(figsize=(12,6))
sub_category_counts.plot(kind='pie', autopct='%1.1f%%',
colors=plt.cm.Paired.colors, startangle=90)
plt.title('Number of Products per Sub-Category')
plt.ylabel('')  # Hides the 'sub-category' label
plt.show()
```



Key Insights

Recommendations

Conclusion

The analysis highlights critical patterns in customer preferences, operational bottlenecks, and inventory flow, providing actionable insights to streamline BigBasket's e-commerce operations. These findings will enable better resource allocation, reduced wastage, and an improved shopping experience for customers.