

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import plotly.express as px
import matplotlib.pyplot as plt
```

```
In [2]: df = pd.read_csv('BigBasket Products.csv')
```

```
In [3]: df
```

Out[3]:

	index	product	category	sub_category	brand	sale_price	market_price	
0	1	Garlic Oil - Vegetarian Capsule 500 mg	Beauty & Hygiene	Hair Care	Sri Sri Ayurveda	220.00	220.0	Hair S
1	2	Water Bottle - Orange	Kitchen, Garden & Pets	Storage & Accessories	Mastercook	180.00	180.0	Wa F B
2	3	Brass Angle Deep - Plain, No.2	Cleaning & Household	Pooja Needs	Trm	119.00	250.0	Lar Lam
3	4	Cereal Flip Lid Container/Storage Jar - Assort...	Cleaning & Household	Bins & Bathroom Ware	Nakoda	149.00	176.0	Lau Sto Ba
4	5	Creme Soft Soap - For Hands & Body	Beauty & Hygiene	Bath & Hand Wash	Nivea	162.00	162.0	Ba B
...								
27550	27551	Wottagirl! Perfume Spray - Heaven, Classic	Beauty & Hygiene	Fragrances & Deos	Layerr	199.20	249.0	Per
27551	27552	Rosemary	Gourmet & World Food	Cooking & Baking Needs	Puramate	67.50	75.0	Seaso &
27552	27553	Peri-Peri Sweet Potato Chips	Gourmet & World Food	Snacks, Dry Fruits, Nuts	FabBox	200.00	200.0	Nach
27553	27554	Green Tea - Pure Original	Beverages	Tea	Tetley	396.00	495.0	Tea
27554	27555	United Dreams Go Far Deodorant	Beauty & Hygiene	Men's Grooming	United Colors Of Benetton	214.53	390.0	Deodc

27555 rows × 10 columns

In [4]: `df.head()`

Out[4]:

	index	product	category	sub_category	brand	sale_price	market_price	type
0	1	Garlic Oil - Vegetarian Capsule 500 mg	Beauty & Hygiene	Hair Care	Sri Sri Ayurveda	220.0	220.0	Hair Oil & Serum
1	2	Water Bottle - Orange	Kitchen, Garden & Pets	Storage & Accessories	Mastercook	180.0	180.0	Water & Fridge Bottles
2	3	Brass Angle Deep - Plain, No.2	Cleaning & Household	Pooja Needs	Trm	119.0	250.0	Lamp & Lamp Oil
3	4	Cereal Flip Lid Container/Storage Jar - Assort...	Cleaning & Household	Bins & Bathroom Ware	Nakoda	149.0	176.0	Laundry, Storage Baskets
4	5	Creme Soft Soap - For Hands & Body	Beauty & Hygiene	Bath & Hand Wash	Nivea	162.0	162.0	Bathing Bars & Soaps

In [5]: `df.tail()`

Out[5]:		index	product	category	sub_category	brand	sale_price	market_price	type	rat
			Wottagirl! Perfume Spray - Heaven, Classic	Beauty & Hygiene	Fragrances & Deos	Layerr	199.20	249.0	Perfume	
	27551	27552	Rosemary	Gourmet & World Food	Cooking & Baking Needs	Puramate	67.50	75.0	Herbs, Seasonings & Rubs	
	27552	27553	Peri-Peri Sweet Potato Chips	Gourmet & World Food	Snacks, Dry Fruits, Nuts	FabBox	200.00	200.0	Nachos & Chips	
	27553	27554	Green Tea - Pure Original	Beverages	Tea	Tetley	396.00	495.0	Tea Bags	
	27554	27555	United Dreams Go Far Deodorant	Beauty & Hygiene	Men's Grooming	United Colors Of Benetton	214.53	390.0	Men's Deodorants	

◀ ▶

In [6]: `df.describe()`

Out[6]:	index	sale_price	market_price	rating
count	27555.00000	27549.00000	27555.00000	18919.00000
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	112475.00000	12500.00000	5.000000

In [7]: `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
```

In [8]: df.isnull().sum()

```
Out[8]: 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
```

In [9]: df

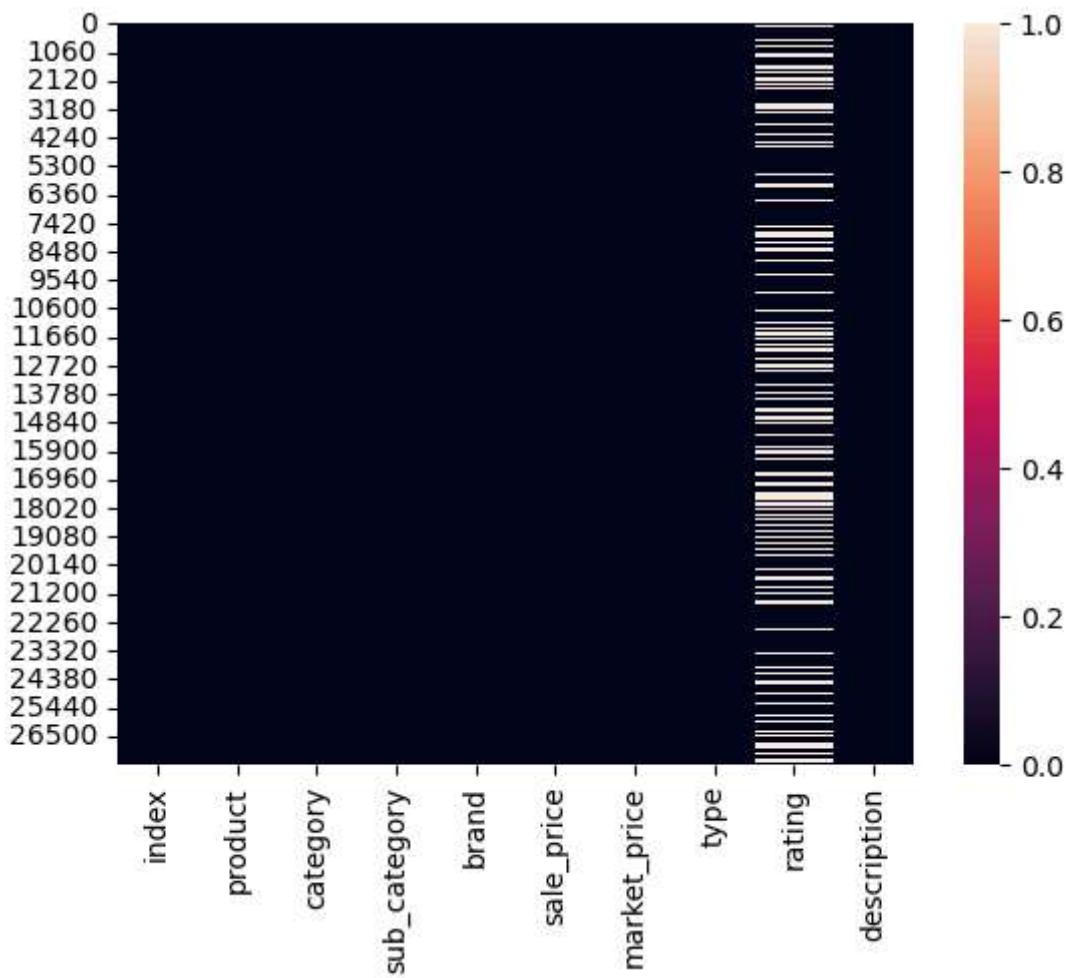
Out[9]:

	index	product	category	sub_category	brand	sale_price	market_price	
0	1	Garlic Oil - Vegetarian Capsule 500 mg	Beauty & Hygiene	Hair Care	Sri Sri Ayurveda	220.00	220.0	Hair S
1	2	Water Bottle - Orange	Kitchen, Garden & Pets	Storage & Accessories	Mastercook	180.00	180.0	Wa F B
2	3	Brass Angle Deep - Plain, No.2	Cleaning & Household	Pooja Needs	Trm	119.00	250.0	Lar Lam
3	4	Cereal Flip Lid Container/Storage Jar - Assort...	Cleaning & Household	Bins & Bathroom Ware	Nakoda	149.00	176.0	Lau Sto Ba
4	5	Creme Soft Soap - For Hands & Body	Beauty & Hygiene	Bath & Hand Wash	Nivea	162.00	162.0	Ba B
...								
27550	27551	Wottagirl! Perfume Spray - Heaven, Classic	Beauty & Hygiene	Fragrances & Deos	Layerr	199.20	249.0	Per
27551	27552	Rosemary	Gourmet & World Food	Cooking & Baking Needs	Puramate	67.50	75.0	Seaso &
27552	27553	Peri-Peri Sweet Potato Chips	Gourmet & World Food	Snacks, Dry Fruits, Nuts	FabBox	200.00	200.0	Nach
27553	27554	Green Tea - Pure Original	Beverages	Tea	Tetley	396.00	495.0	Tea
27554	27555	United Dreams Go Far Deodorant	Beauty & Hygiene	Men's Grooming	United Colors Of Benetton	214.53	390.0	Deodc

27555 rows × 10 columns

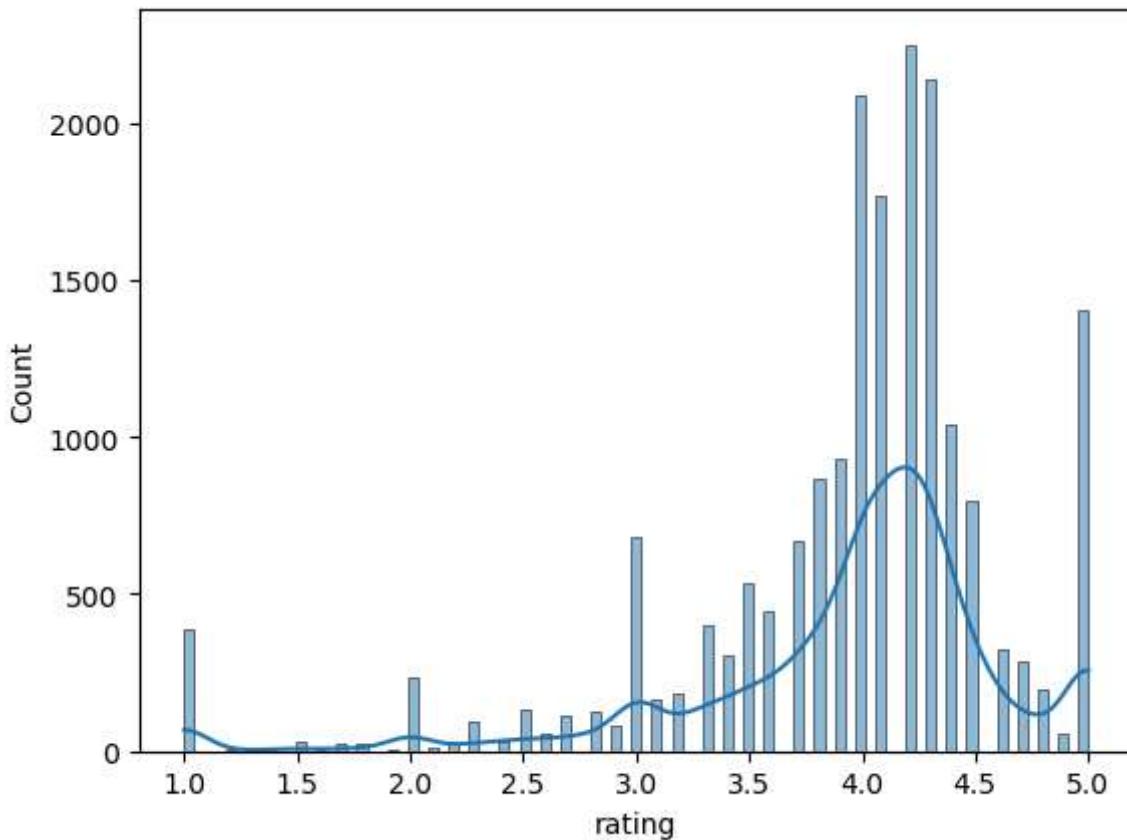
```
In [10]: sns.heatmap(df.isnull())
```

```
Out[10]: <AxesSubplot:>
```



```
In [11]: sns.histplot(df.rating, kde=True)
```

```
Out[11]: <AxesSubplot:xlabel='rating', ylabel='Count'>
```

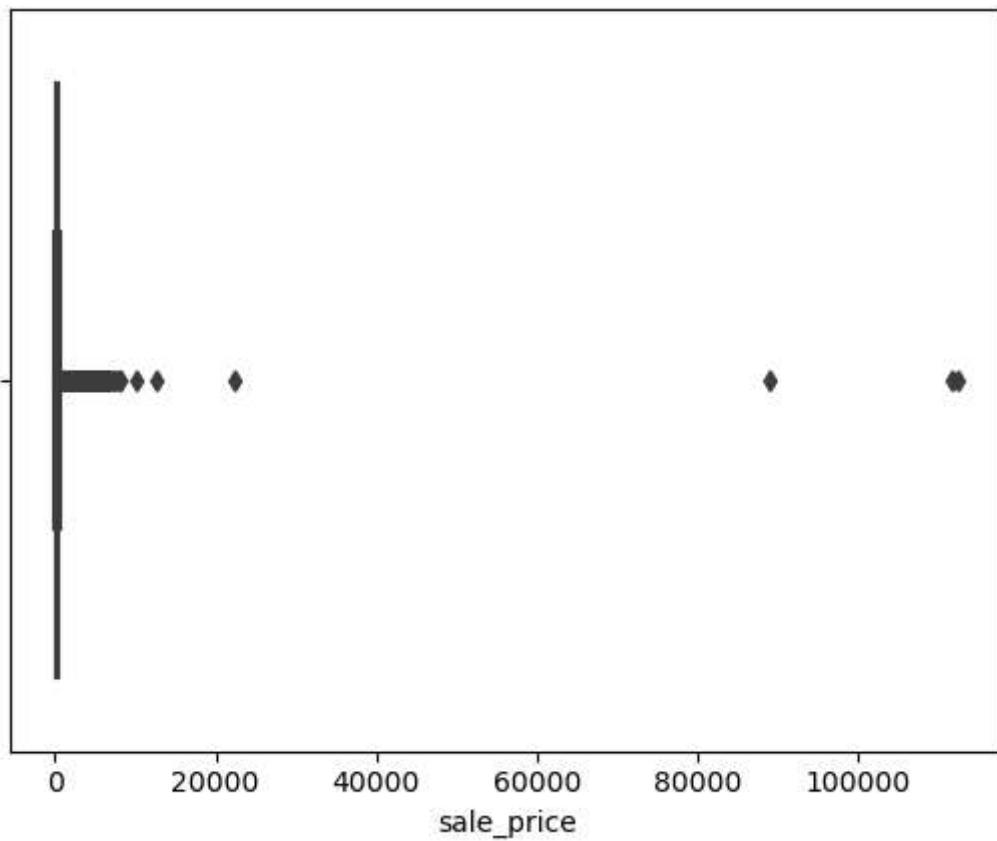


```
In [12]: sns.boxplot(df.sale_price)
```

```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
```

```
    warnings.warn(
```

```
Out[12]: <AxesSubplot:xlabel='sale_price'>
```



```
In [13]: Q1,Q3 = df.sale_price.quantile([0.25,0.75])
```

```
In [14]: Q1,Q3
```

```
Out[14]: (95.0, 359.0)
```

```
In [15]: IQR = Q3-Q1
```

```
In [16]: IQR
```

```
Out[16]: 264.0
```

```
In [17]: LL = Q1-1.5*(IQR)  
UL = Q3+1.5*(IQR)
```

```
In [18]: LL,UL
```

```
Out[18]: (-301.0, 755.0)
```

```
In [19]: df1 = df[df['sale_price']>UL]
```

```
In [20]: df1
```

Out[20]:

	index	product	category	sub_category	brand	sale_price	market_price	type
8	9	Biotin & Collagen Volumizing Hair Shampoo + Bi...	Beauty & Hygiene	Hair Care	StBotanica	1098.00	1098.0	Shampoo & Conditioner
47	48	Colour Catcher Sheets	Cleaning & Household	All Purpose Cleaners	Dylon	799.00	799.0	Imported Cleaners
51	52	Peach Syrup	Gourmet & World Food	Drinks & Beverages	Pekers	850.00	850.0	Gourmet Juices & Drinks
69	70	Padded Harness - 3/4 inch, Grey Colour	Kitchen, Garden & Pets	Pet Food & Accessories	Glenand	840.00	840.0	Pet Collars & Leashes
91	92	Hard Anodised Ezee-Pour Saucepan With Lid - L88	Kitchen, Garden & Pets	Cookware & Non Stick	Hawkins Futura	864.50	910.0	Tawa & Sauce Pan
...
27498	27499	Juicer - Fruit & Vegetable, Light Green	Kitchen, Garden & Pets	Kitchen Accessories	Ganesh	1071.00	1071.0	Kitchen Tools & Other Accessories
27505	27506	Virgin Coconut Oil	Foodgrains, Oil & Masala	Edible Oils & Ghee	Merkera	875.00	875.0	Other Edible Oils
27514	27515	Verge & Sheer Perfume For Pair	Beauty & Hygiene	Fragrances & Deos	Skinn by Titan	1615.50	1795.0	Perfume
27538	27539	Quista Pro Advanced Whey Protein Formula fort...	Beauty & Hygiene	Health & Medicine	Himalaya	4500.00	4500.0	Supplements & Proteins

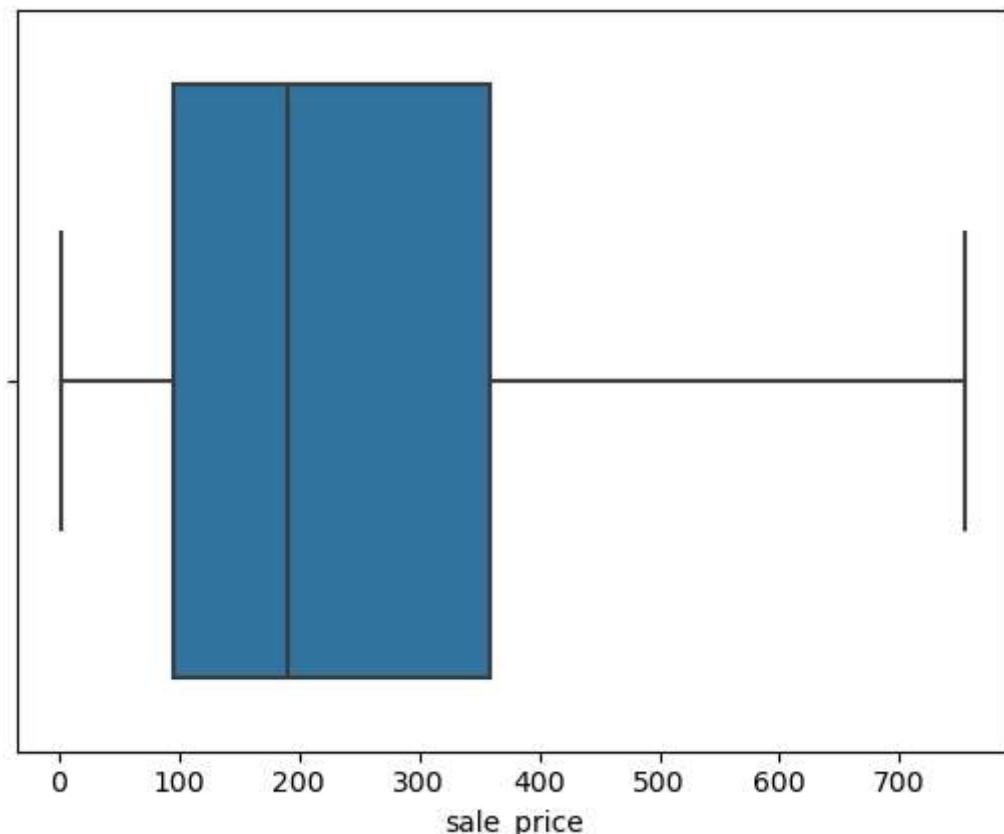
	index	product	category	sub_category	brand	sale_price	market_price	type
27542	27543	Embellish Skin Lightening Serum	Beauty & Hygiene	Skin Care	Organic Harvest	1525.75	1795.0	Face Care

2205 rows × 10 columns

In [21]: `df.sale_price = np.where(df['sale_price']>UL,UL,df['sale_price'])`In [22]: `sns.boxplot(df.sale_price)`

C:\Users\DELL\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Out[22]: `<AxesSubplot:xlabel='sale_price'>`In [23]: `df`

Out[23]:

	index	product	category	sub_category	brand	sale_price	market_price	
0	1	Garlic Oil - Vegetarian Capsule 500 mg	Beauty & Hygiene	Hair Care	Sri Sri Ayurveda	220.00	220.0	Hair S
1	2	Water Bottle - Orange	Kitchen, Garden & Pets	Storage & Accessories	Mastercook	180.00	180.0	Wa F B
2	3	Brass Angle Deep - Plain, No.2	Cleaning & Household	Pooja Needs	Trm	119.00	250.0	Lar Lam
3	4	Cereal Flip Lid Container/Storage Jar - Assort...	Cleaning & Household	Bins & Bathroom Ware	Nakoda	149.00	176.0	Lau Sto Ba
4	5	Creme Soft Soap - For Hands & Body	Beauty & Hygiene	Bath & Hand Wash	Nivea	162.00	162.0	Ba B
...								
27550	27551	Wottagirl! Perfume Spray - Heaven, Classic	Beauty & Hygiene	Fragrances & Deos	Layerr	199.20	249.0	Per
27551	27552	Rosemary	Gourmet & World Food	Cooking & Baking Needs	Puramate	67.50	75.0	Seaso &
27552	27553	Peri-Peri Sweet Potato Chips	Gourmet & World Food	Snacks, Dry Fruits, Nuts	FabBox	200.00	200.0	Nach
27553	27554	Green Tea - Pure Original	Beverages	Tea	Tetley	396.00	495.0	Tea
27554	27555	United Dreams Go Far Deodorant	Beauty & Hygiene	Men's Grooming	United Colors Of Benetton	214.53	390.0	Deodc

27555 rows × 10 columns

```
In [24]: Data = df.category.value_counts()
```

```
In [25]: Data
```

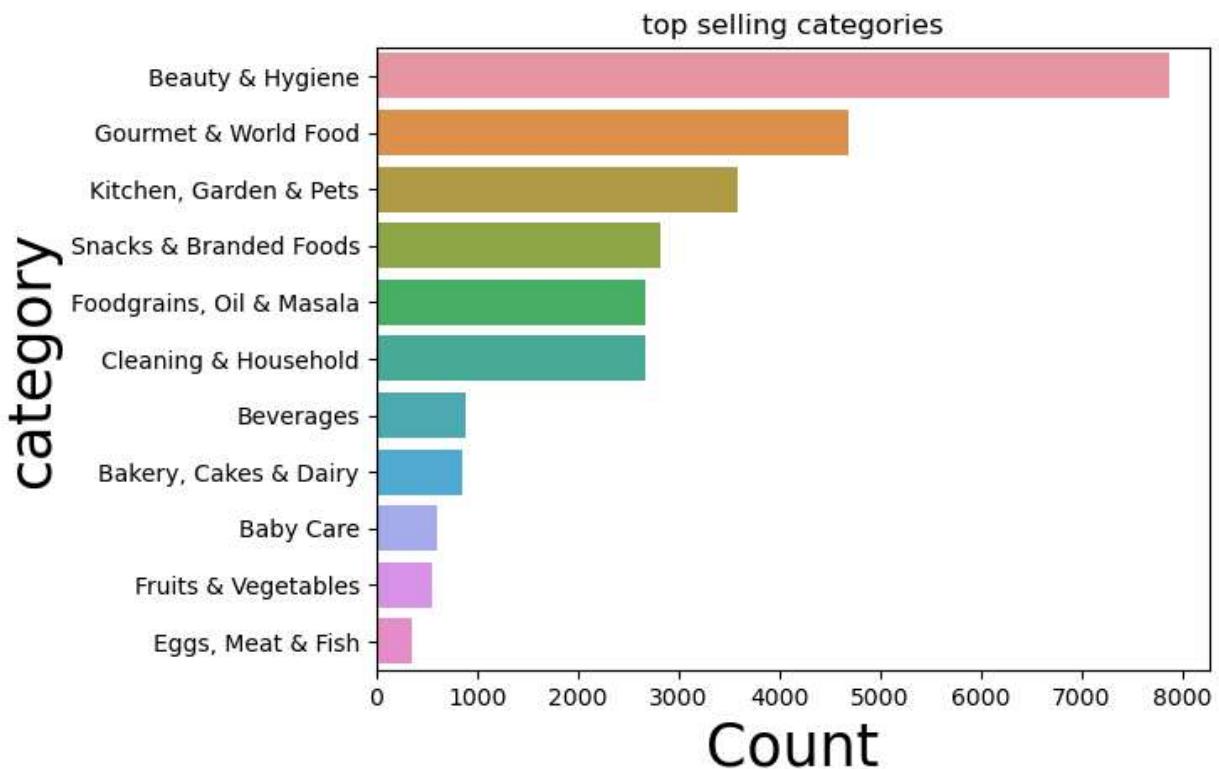
```
Out[25]:
```

Beauty & Hygiene	7867
Gourmet & World Food	4690
Kitchen, Garden & Pets	3580
Snacks & Branded Foods	2814
Foodgrains, Oil & Masala	2676
Cleaning & Household	2675
Beverages	885
Bakery, Cakes & Dairy	851
Baby Care	610
Fruits & Vegetables	557
Eggs, Meat & Fish	350

Name: category, dtype: int64

```
In [39]: sns.barplot(x=Data,y=Data.index)
plt.xlabel('Count',fontdict={'fontsize':25})
plt.ylabel('category',fontdict={'fontsize':25})
plt.title('top selling categories')
```

```
Out[39]: Text(0.5, 1.0, 'top selling categories')
```



```
In [27]: Q1,Q3 = df.market_price.quantile([0.25,0.75])
```

```
In [28]: Q1,Q3
```

```
Out[28]: (100.0, 425.0)
```

```
In [29]: IQR = Q3-Q1
```

```
In [30]: IQR
```

```
Out[30]: 325.0
```

```
In [31]: LL = Q1-1.5*(IQR)  
UL = Q3+1.5*(IQR)
```

```
In [32]: LL,UL
```

```
Out[32]: (-387.5, 912.5)
```

```
In [33]: df1 = df[df['market_price']>UL]
```

```
In [34]: df1
```

Out[34]:

	index	product	category	sub_category	brand	sale_price	market_price	type	r
8	9	Biotin & Collagen Volumizing Hair Shampoo + Bi...	Beauty & Hygiene	Hair Care	StBotanica	755.0	1098.0	Shampoo & Conditioner	
34	35	Pet Solitaire Container Set - Silver	Kitchen, Garden & Pets	Storage & Accessories	Steelo	499.0	969.0	Containers Sets	
96	97	Hair Remover Spray - Foam Lemon	Beauty & Hygiene	Feminine Hygiene	Dimples	755.0	1200.0	Hair Removal	
99	100	Wonder Diaper Pants - Xtra Large, 12-17 Kg	Baby Care	Diapers & Wipes	Huggies	755.0	1398.0	Diapers	
118	119	Adult Dry Cat Food - +1 Year, Ocean Fish	Kitchen, Garden & Pets	Pet Food & Accessories	Maxi Persian	755.0	999.0	Pet Meals & Treats	
...
27488	27489	Cranberry - Dried, Sliced	Gourmet & World Food	Snacks, Dry Fruits, Nuts	Rostaa	755.0	980.0	Dry Fruits & Berries	
27498	27499	Juicer - Fruit & Vegetable, Light Green	Kitchen, Garden & Pets	Kitchen Accessories	Ganesh	755.0	1071.0	Kitchen Tools & Other Accessories	
27514	27515	Verge & Sheer Perfume For Pair	Beauty & Hygiene	Fragrances & Deos	Skinn by Titan	755.0	1795.0	Perfume	
27538	27539	Quista Pro Advanced Whey Protein Formula fort...	Beauty & Hygiene	Health & Medicine	Himalaya	755.0	4500.0	Supplements & Proteins	

	index	product	category	sub_category	brand	sale_price	market_price	type	r
27542	27543	Embellish Skin Lightening Serum	Beauty & Hygiene	Skin Care	Organic Harvest	755.0	1795.0	Face Care	

2147 rows × 10 columns

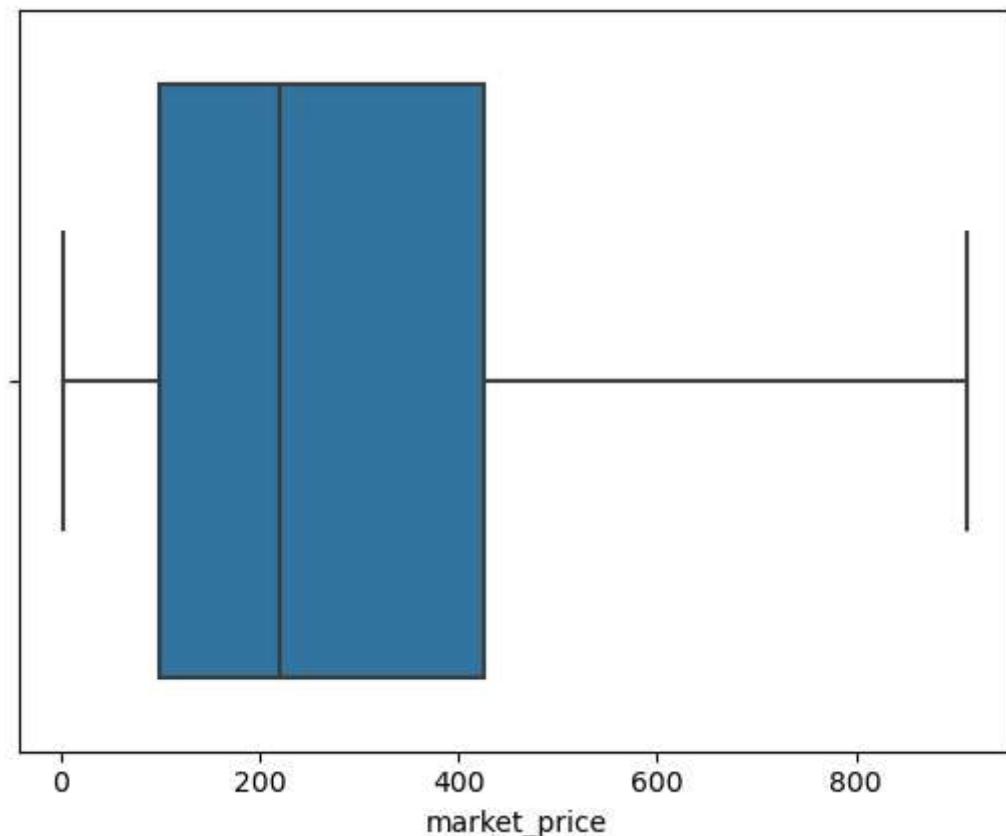
```
In [35]: df.market_price = np.where(df['market_price'] > UL, UL, df['market_price'])
```

```
In [36]: sns.boxplot(df.market_price)
```

C:\Users\DELL\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

```
Out[36]: <AxesSubplot:xlabel='market_price'>
```



```
In [ ]:
```