```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data=pd.read_csv("BigBasket Products.csv")
data
```

	index	product	category	sub_category	brand	sale_price	market_price	type	rating	description
0	1	Garlic Oil - Vegetarian Capsule 500 mg	Beauty & Hygiene	Hair Care	Sri Sri Ayurveda	220.00	220.0	Hair Oil & Serum	4.1	This Product contains Garlic Oil that is known...
1	2	Water Bottle - Orange	Kitchen, Garden & Pets	Storage & Accessories	Mastercook	180.00	180.0	Water & Fridge Bottles	2.3	Each product is microwave safe (without lid), ...
2	3	Brass Angle Deep - Plain, No.2	Cleaning & Household	Pooja Needs	Trm	119.00	250.0	Lamp & Lamp Oil	3.4	A perfect gift for all occasions, be it your m...
3	4	Cereal Flip Lid Container/Storage Jar - Assort...	Cleaning & Household	Bins & Bathroom Ware	Nakoda	149.00	176.0	Laundry, Storage Baskets	3.7	Multipurpose container with an attractive desi...
4	5	Creme Soft Soap - For Hands & Body	Beauty & Hygiene	Bath & Hand Wash	Nivea	162.00	162.0	Bathing Bars & Soaps	4.4	Nivea Creme Soft Soap gives your skin the best...
...	...	...	...	...	...	...	...	...	...	...
27550	27551	Wottagirl! Perfume Spray - Heaven, Classic	Beauty & Hygiene	Fragrances & Deos	Layerr	199.20	249.0	Perfume	3.9	Layerr brings you Wottagirl Classic fragrant b...
27551	27552	Rosemary	Gourmet & World	Cooking & Baking Needs	Puramate	67.50	75.0	Herbs, Seasonings	4.0	Puramate rosemary is enough to

Next steps: Generate code with data View recommended plots New interactive sheet

```
data.head(12)
```

	index	product	category	sub_category	brand	sale_price	market_price	type	rating	description
0	1	Garlic Oil - Vegetarian Capsule 500 mg	Beauty & Hygiene	Hair Care	Sri Sri Ayurveda	220.0	220.0	Hair Oil & Serum	4.1	This Product contains Garlic Oil that is known...
1	2	Water Bottle - Orange	Kitchen, Garden & Pets	Storage & Accessories	Mastercook	180.0	180.0	Water & Fridge Bottles	2.3	Each product is microwave safe (without lid), ...
2	3	Brass Angle Deep - Plain, No.2	Cleaning & Household	Pooja Needs	Trm	119.0	250.0	Lamp & Lamp Oil	3.4	A perfect gift for all occasions, be it your m...
3	4	Cereal Flip Lid Container/Storage Jar - Assort...	Cleaning & Household	Bins & Bathroom Ware	Nakoda	149.0	176.0	Laundry, Storage Baskets	3.7	Multipurpose container with an attractive desi...
4	5	Creme Soft Soap - For Hands & Body	Beauty & Hygiene	Bath & Hand Wash	Nivea	162.0	162.0	Bathing Bars & Soaps	4.4	Nivea Creme Soft Soap gives your skin the best...
5	6	Germ - Removal Multipurpose Wipes	Cleaning & Household	All Purpose Cleaners	Nature Protect	169.0	199.0	Disinfectant Spray & Cleaners	3.3	Stay protected from contamination with Multipu...
6	7	Multani Mati	Beauty & Hygiene	Skin Care	Satinance	58.0	58.0	Face Care	3.6	Satinance multani matti is an excellent skin t...
								Hand Wash		70%Alcohol

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
data.tail(12)
```

	index	product	category	sub_category	brand	sale_price	market_price	type	rating	description
27543	27544	Popcorn - French Butter & Pink Salt	Gourmet & World Food	Snacks, Dry Fruits, Nuts	4700BC	31.50	35.0	Gourmet Popcorn	4.1	High-quality mushroom corn popped in olive oil...
27544	27545	Brass Dhoop Aarti With Wooden Handle,No.3	Cleaning & Household	Pooja Needs	Trm	239.00	495.0	Other Pooja Needs	4.0	Dhoobakal is a vessel used for burning the Dho...
27545	27546	Toilet Cleaning Brush - Round With Holder (Big)	Cleaning & Household	Mops, Brushes & Scrubs	Liao	189.00	349.0	Toilet & Other Brushes	3.8	This round toilet brush is made up of virgin q...
27546	27547	Organic Powder - Garam Masala	Foodgrains, Oil & Masala	Organic Staples	Organic Tattva	152.00	160.0	Organic Masalas & Spices	4.2	Organic Tattva Garam masala is a famous spice ...
27547	27548	Powder - Baking	Snacks & Branded Foods	Ready To Cook & Eat	Kwality	38.00	38.0	Home Baking	3.7	Kwality Baking Powder Directions for use: Use ...
27548	27549	Apple Cider Vinegar Shampoo	Beauty & Hygiene	Hair Care	Morpheme Remedies	499.00	499.0	Shampoo & Conditioner	5.0	Say no to dull, lifeless, dry and damaged hair...
27549	27550	Papad - Garlic Disco	Snacks & Branded Foods	Ready To Cook & Eat	Atish	61.00	61.0	Papads, Ready To Fry	4.0	Papads are prepared from urad dal flour and sp...
		Wottagirl!								Laverr brings

```
data.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

```
data.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
```

```
data.shape
```

```
(27555, 10)
```

```
data.size
```

```
275550
```

```
data.sample()
```

	index	product	category	sub_category	brand	sale_price	market_price	type	rating	description
8939	8940	Dilli Dahi Kebab	Snacks & Branded Foods	Frozen Veggies & Snacks	ITC Master Chef	123.5	130.0	Frozen Veg Snacks	3.4	A spicy kebab with the goodness of curd combin...

```
data.columns
```

```
Index(['index', 'product', 'category', 'sub_category', 'brand', 'sale_price',
       'market_price', 'type', 'rating', 'description'],
      dtype='object')
```

```
data.dtypes
```

	0
index	int64
product	object
category	object
sub_category	object
brand	object
sale_price	float64
market_price	float64
type	object
rating	float64
description	object

dtype: object

```
data.values
```

Show hidden output

```
data.index
```

```
RangeIndex(start=0, stop=27555, step=1)
```

```
data.isnull().sum()
```

```
                 0
      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
```

```
unique_counts = data[['category', 'sub_category', 'brand', 'type']].nunique()
```

```
numerical_summary = data[['sale_price', 'market_price', 'rating']].describe()
```

```
data['rating']=data['rating'].fillna(data['rating'].mean()).inplace=True
```

```
data.isnull().sum()
```

```
                 0
      index      0
    product      1
   category      0
sub_category     0
      brand      1
 sale_price      6
market_price     0
       type      0
     rating      0
description    115

dtype: int64
```

```
data.notnull()
```

	index	product	category	sub_category	brand	sale_price	market_price	type	rating	description
0	True	True	True	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True	True	True	True
2	True	True	True	True	True	True	True	True	True	True
3	True	True	True	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True	True	True	True
...	...	...	...	...	...	...	...	...	...	...
27550	True	True	True	True	True	True	True	True	True	True
27551	True	True	True	True	True	True	True	True	True	True
27552	True	True	True	True	True	True	True	True	True	True
27553	True	True	True	True	True	True	True	True	True	True
27554	True	True	True	True	True	True	True	True	True	True

27555 rows × 10 columns

```
data.drop_duplicates()
```

	index	product	category	sub_category	brand	sale_price	market_price	type	rating	description
0	1	Garlic Oil - Vegetarian Capsule 500 mg	Beauty & Hygiene	Hair Care	Sri Sri Ayurveda	220.00	220.0	Hair Oil & Serum	True	This Product contains Garlic Oil that is known...
1	2	Water Bottle - Orange	Kitchen, Garden & Pets	Storage & Accessories	Mastercook	180.00	180.0	Water & Fridge Bottles	True	Each product is microwave safe (without lid), ...
2	3	Brass Angle Deep - Plain, No.2	Cleaning & Household	Pooja Needs	Trm	119.00	250.0	Lamp & Lamp Oil	True	A perfect gift for all occasions, be it your m...
3	4	Cereal Flip Lid Container/Storage Jar - Assort...	Cleaning & Household	Bins & Bathroom Ware	Nakoda	149.00	176.0	Laundry, Storage Baskets	True	Multipurpose container with an attractive desi...
4	5	Creme Soft Soap - For Hands & Body	Beauty & Hygiene	Bath & Hand Wash	Nivea	162.00	162.0	Bathing Bars & Soaps	True	Nivea Creme Soft Soap gives your skin the best...
...	...	...	...	...	...	...	...	...	...	...
27550	27551	Wottagirl! Perfume Spray - Heaven, Classic	Beauty & Hygiene	Fragrances & Deos	Layerr	199.20	249.0	Perfume	True	Layerr brings you Wottagirl Classic fragrant b...
27551	27552	Rosemary	Gourmet & World	Cooking & Baking Needs	Puramate	67.50	75.0	Herbs, Seasonings	True	Puramate rosemary is enough to

```
data['rating']=data['rating'].astype(int)
```

```
data['category'].value_counts()
```

category	count
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

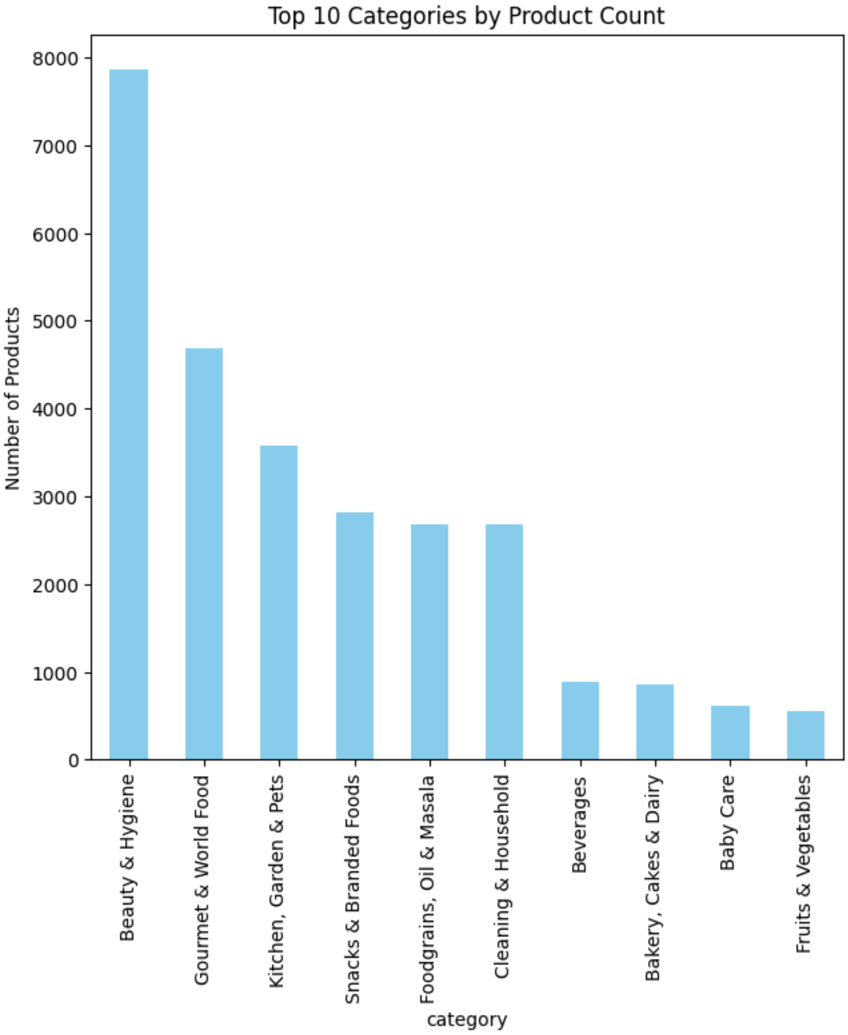
dtype: int64

```
top_categories = data['category'].value_counts().head(10)
```

Top 10 category by Product Count

```
plt.figure(figsize=(16, 24))
plt.subplot(3, 2, 1)
top_categories.plot(kind='bar', color='skyblue')
plt.title("Top 10 Categories by Product Count")
plt.ylabel("Number of Products")
```

```
Text(0, 0.5, 'Number of Products')
```

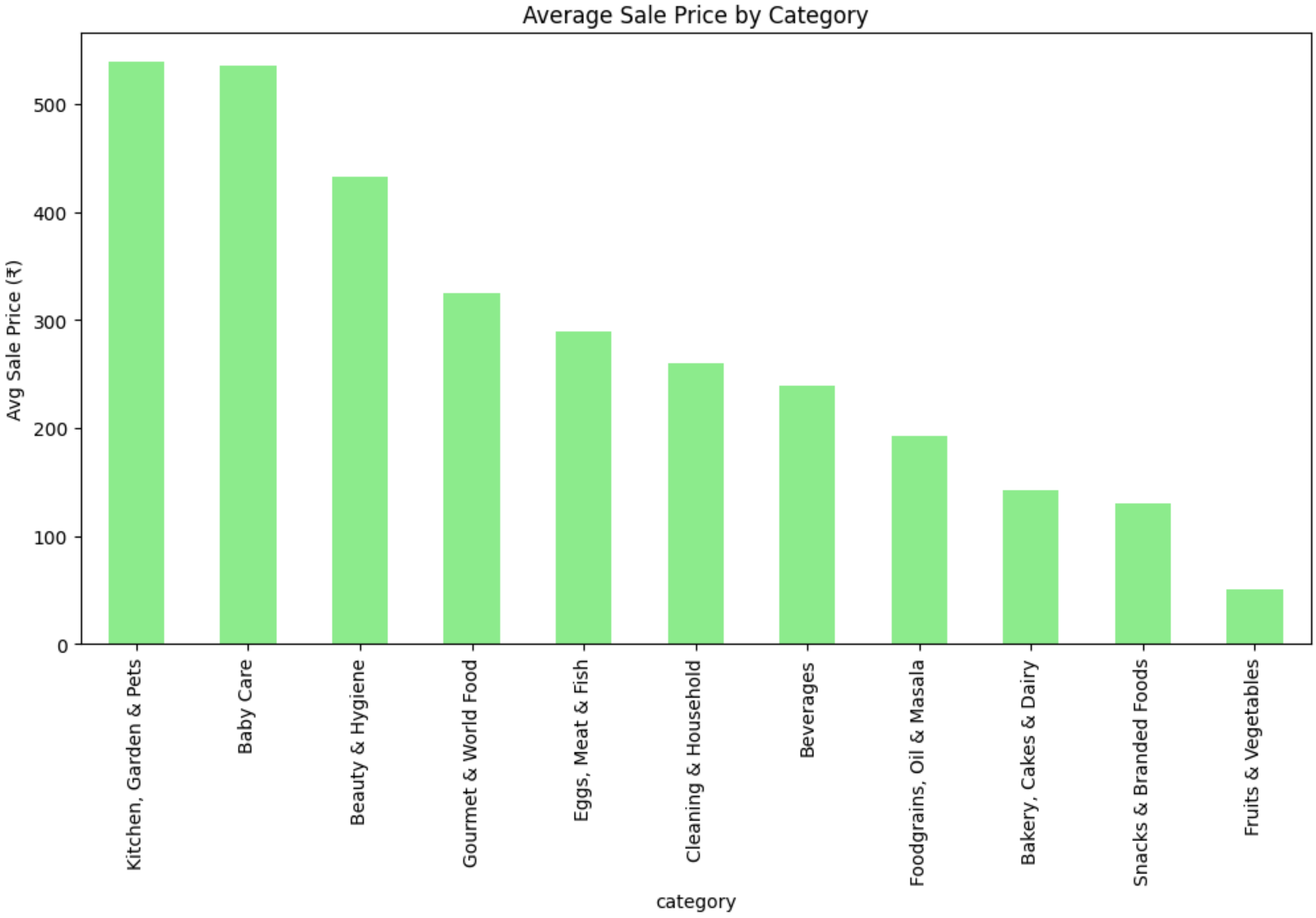


```
avg_price_by_category = data.groupby('category')['sale_price'].mean().sort_values(ascending=False)
```

Figure showing Average Sale Price by Category

```
plt.figure(figsize=(12, 6))
avg_price_by_category.plot(kind='bar', color='lightgreen')
plt.title("Average Sale Price by Category")
plt.ylabel("Avg Sale Price (₹)")
```

```
Text(0, 0.5, 'Avg Sale Price (₹)')
```

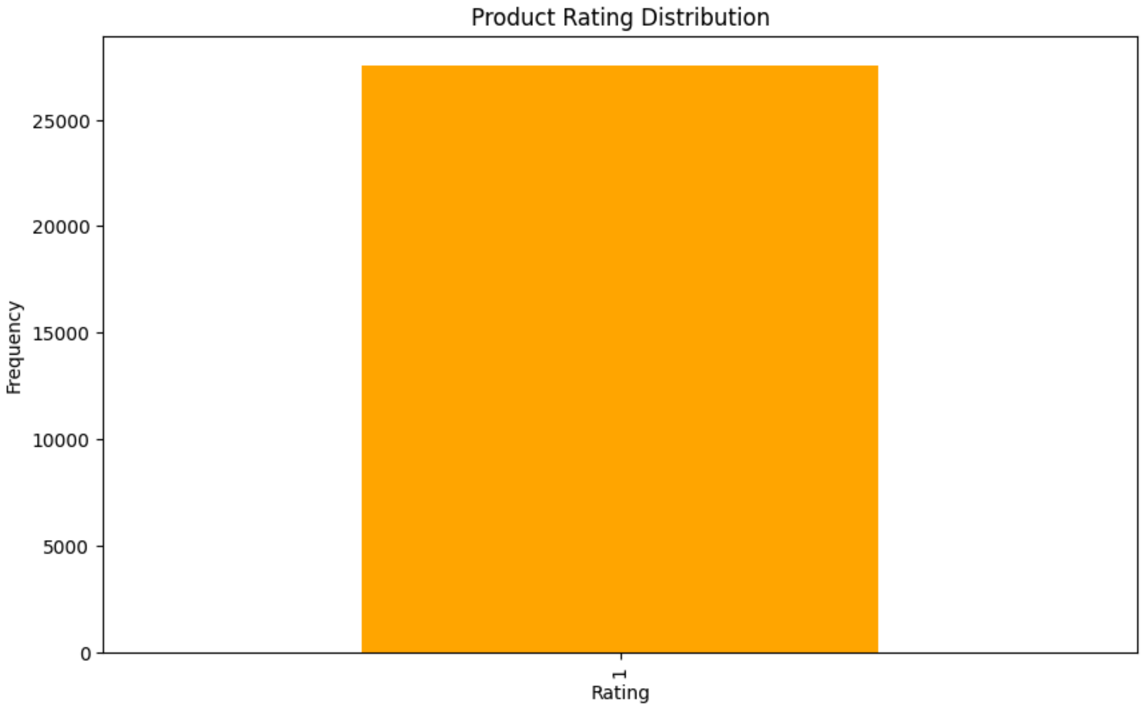


```
rating_distribution = data['rating'].value_counts().sort_index()
```

Figure showing Product Rating Distribution

```
plt.figure(figsize=(10, 6))
rating_distribution.plot(kind='bar', color='orange')
plt.title("Product Rating Distribution")
plt.xlabel("Rating")
plt.ylabel("Frequency")
```

```
Text(0, 0.5, 'Frequency')
```

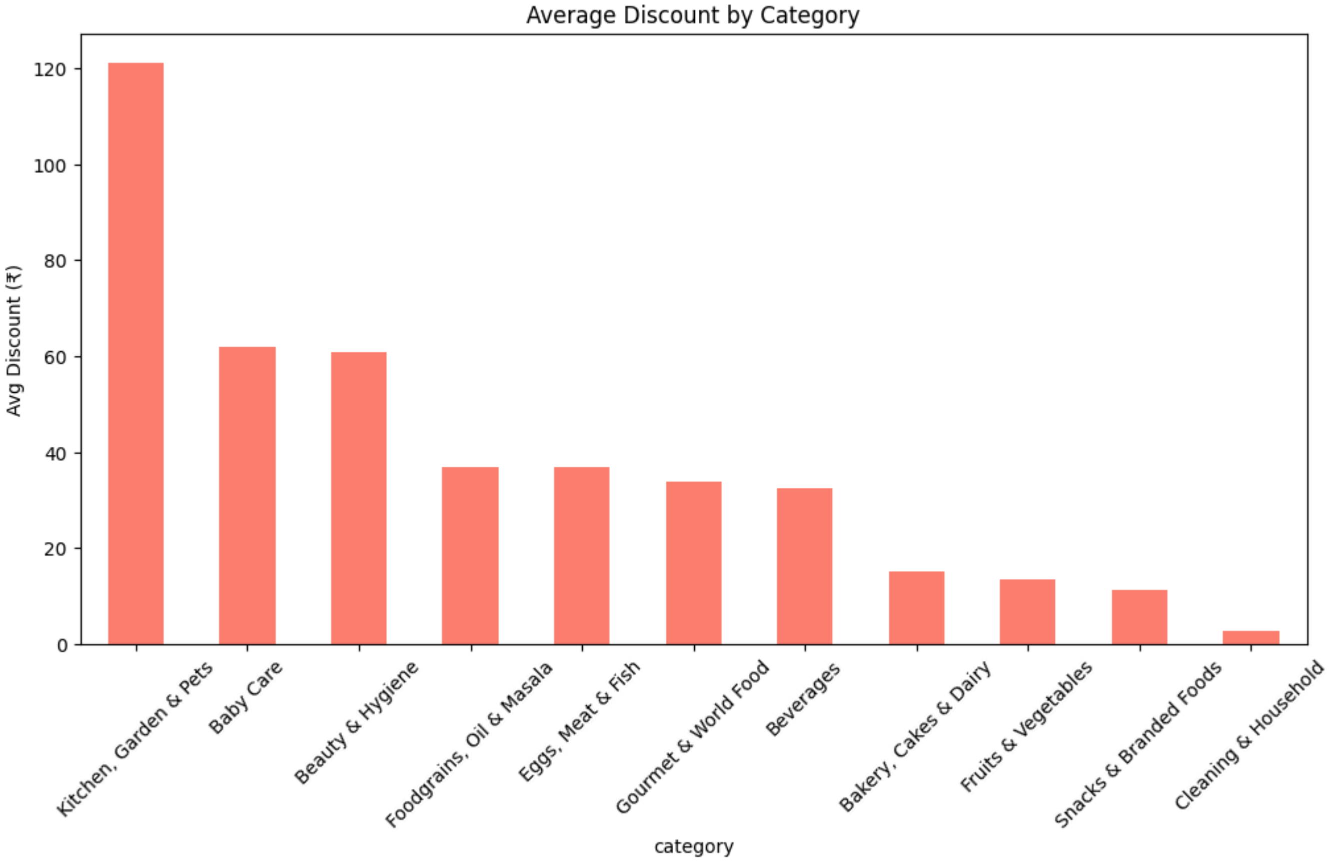


```
data['discount'] = data['market_price'] - data['sale_price']
avg_discount_by_category = data.groupby('category')['discount'].mean().sort_values(ascending=False)
```

Figure showing Average Discount by Category

```
plt.figure(figsize=(12, 6))
avg_discount_by_category.plot(kind='bar', color='salmon')
plt.title("Average Discount by Category")
plt.ylabel("Avg Discount (₹)")
plt.xticks(rotation=45)
```

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10]),
 [Text(0, 0, 'Kitchen, Garden & Pets'),
  Text(1, 0, 'Baby Care'),
  Text(2, 0, 'Beauty & Hygiene'),
  Text(3, 0, 'Foodgrains, Oil & Masala'),
  Text(4, 0, 'Eggs, Meat & Fish'),
  Text(5, 0, 'Gourmet & World Food'),
  Text(6, 0, 'Beverages'),
  Text(7, 0, 'Bakery, Cakes & Dairy'),
  Text(8, 0, 'Fruits & Vegetables'),
  Text(9, 0, 'Snacks & Branded Foods'),
  Text(10, 0, 'Cleaning & Household')])
```

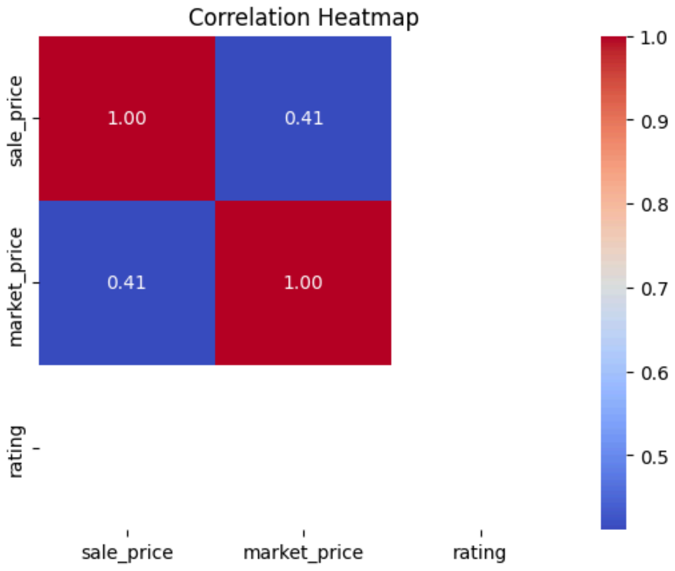


```
correlation_matrix = data[['sale_price', 'market_price', 'rating']].corr()
```

Correlation Heatmap

```
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
```

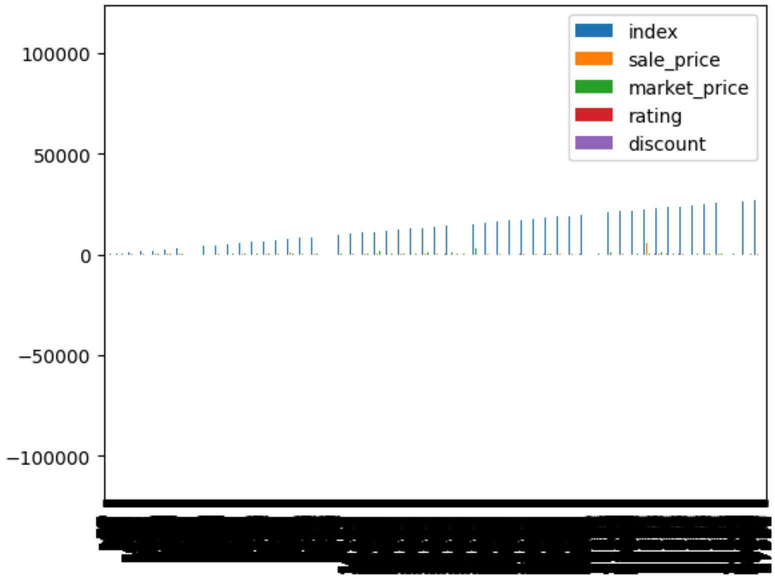
```
Text(0.5, 1.0, 'Correlation Heatmap')
```



BAR PLOT OF DATA

```
data.plot(kind='bar')
```

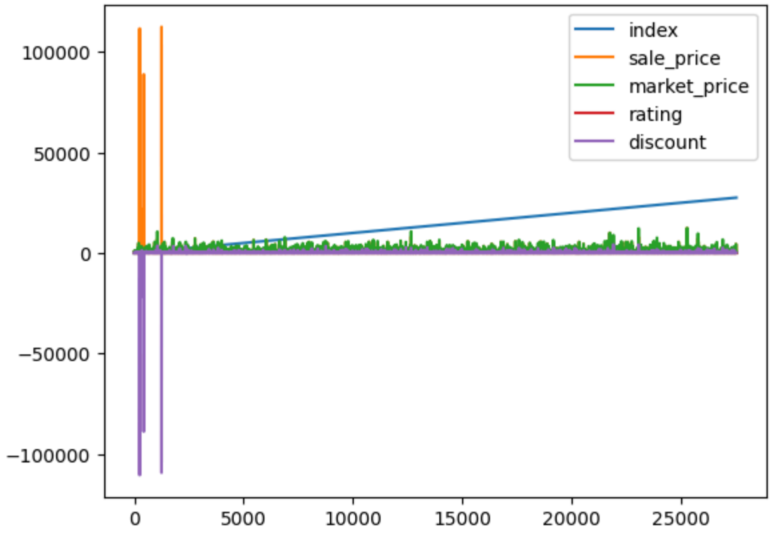
```
<Axes: >
```



LINE PLOT OF DATA

```
data.plot(kind='line')
```

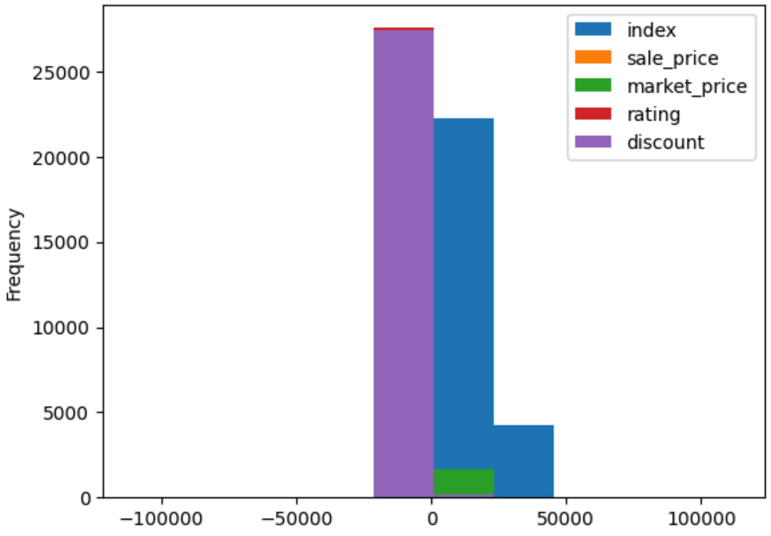
```
<Axes: >
```



HIST PLOT OF DATA

```
data.plot(kind='hist')
```

```
<Axes: ylabel='Frequency'>
```



BOX PLOT OF DATA

```
data.plot(kind='box')
```

Show hidden output

SCATTER PLOT OF RATING VS SALE PRICE

```
data.plot(x='rating', y='sale_price', kind='scatter')
```

Show hidden output

SCATTER PLOT OF RATING VS MARKET PRICE

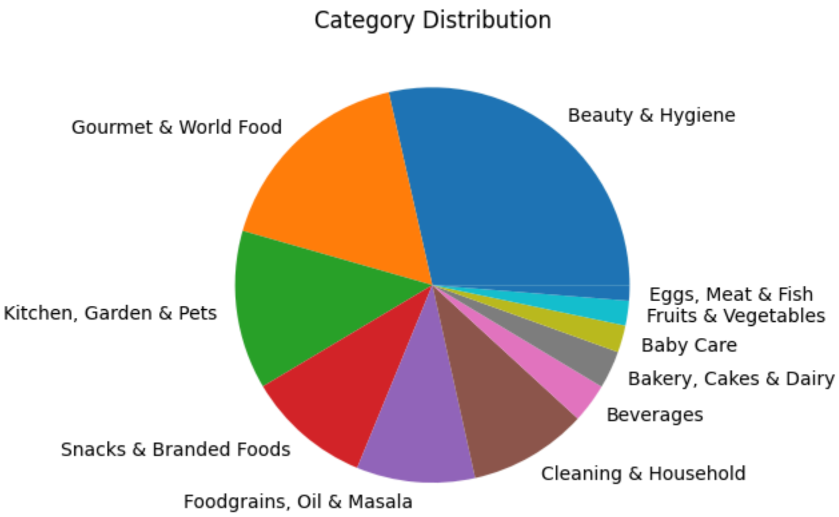
```
data.plot(x='rating', y='market_price', kind='scatter', color='red')
```

Show hidden output

PIE CHART OF CATEGORY DISTRIBUTION

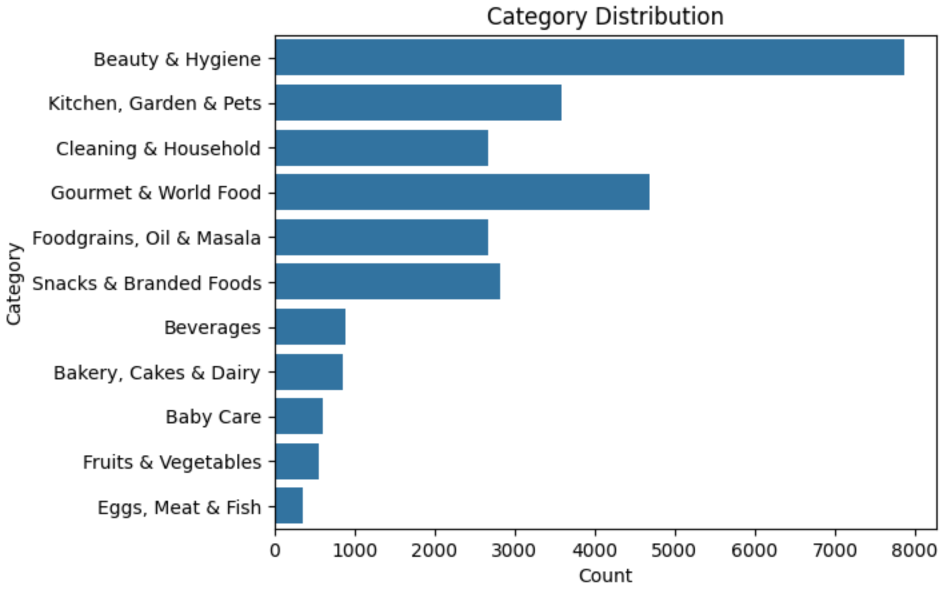
```
category_counts = data['category'].value_counts()

plt.pie(category_counts.values, labels=category_counts.index)
plt.title('Category Distribution')
plt.show()
```



COUNT CHART OF CATEGORY DISTRIBUTION

```
sns.countplot(data['category'])
plt.title('Category Distribution')
plt.xlabel('Count')
plt.ylabel('Category')
plt.show()
```



```
data.drop(['rating'], axis=1, inplace=True)
data.drop(['index'], axis=1, inplace=True)
```

TOP AND LEAST 5 SOLD PRODUCT

```
top_products = data.sort_values(by='sale_price', ascending=False).head(5)
print("Top 5 Sold Products:\n", top_products[['sub_category', 'sale_price']])


least_products = data.sort_values(by='sale_price', ascending=True).head(5)
print("Least 5 Sold Products:\n", least_products[['sub_category', 'sale_price']])
```

```
Top 5 Sold Products:
                  sub_category  sale_price
1249            Men's Grooming    112475.0
248         Cookware & Non Stick    111649.0
436        Party & Festive Needs     88899.0
288       Sauces, Spreads & Dips     22325.0
25301   Pet Food & Accessories     12500.0
Least 5 Sold Products:
              sub_category  sale_price
26976  Herbs & Seasonings        2.45
21312           Hair Care        3.00
27490  Biscuits & Cookies        5.00
27413    Cakes & Pastries        5.00
14603  Biscuits & Cookies        5.00
```

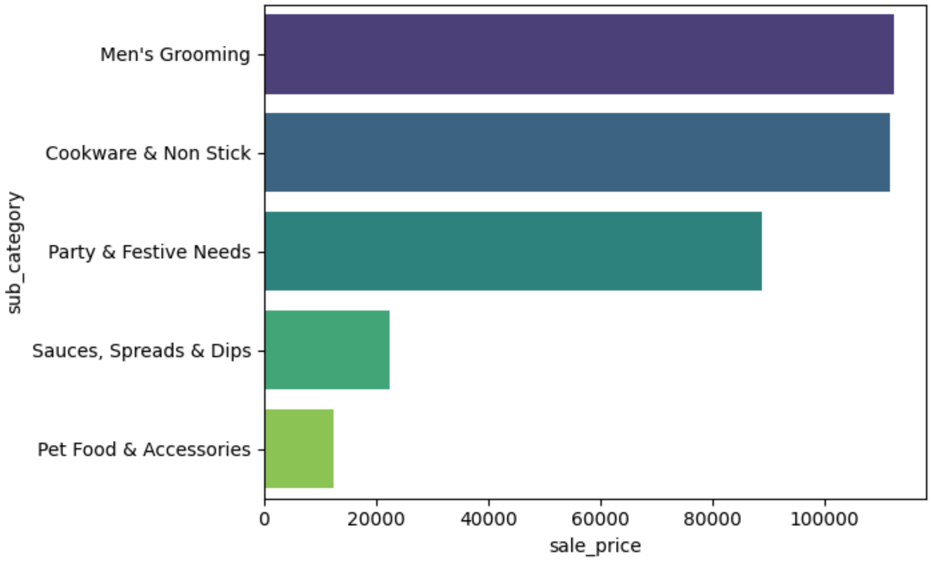
Top 5 Sold Product

```
sns.barplot(x='sale_price', y='sub_category',data=top_products,palette='viridis')
plt.show()
```

```
/tmp/ipython-input-16-4219147881.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue`

  sns.barplot(x='sale_price', y='sub_category',data=top_products,palette='viridis')
```



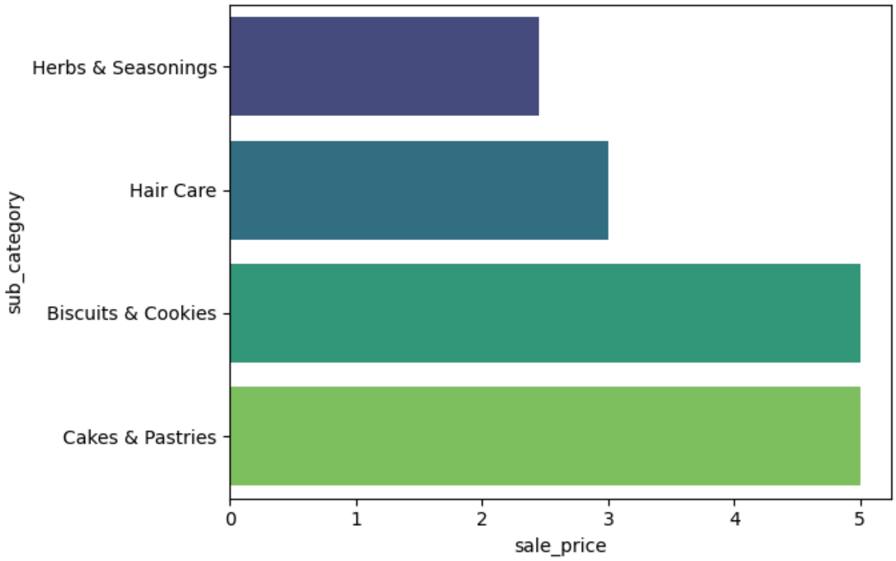
Top 4 Least Sold Product

```
sns.barplot(x='sale_price', y='sub_category',data=least_products,palette='viridis')
plt.show()
```

```
/tmp/ipython-input-18-4276360255.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue`

  sns.barplot(x='sale_price', y='sub_category',data=least_products,palette='viridis')
```



Measuring Discount on Certain Item

```
data['discount_percent'] = ((data['market_price'] - data['sale_price']) / data['market_price']) * 100

product_name = data['product'][0]
discount = data[data['product'] == product_name]['discount_percent'].iloc[0]

print(f"Discount for {product_name}: {discount:.2f}%")
```

```
Discount for Garlic Oil - Vegetarian Capsule 500 mg: 0.00%
```

LABEL ENCODING

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
a=['product','category','sub_category','brand','type','description']
for i in a:
  data[i]=le.fit_transform(data[i])
```

```
data.head()
```

	product	category	sub_category	brand	sale_price	market_price	type	description	discount	discount_percent
0	8277	2	49	1959	220.0	220.0	204	18758	0.0	0.000000
1	22935	9	86	1258	180.0	180.0	420	5402	0.0	0.000000
2	2957	4	73	2125	119.0	250.0	249	865	131.0	52.400000
3	3573	4	9	1386	149.0	176.0	250	11693	27.0	15.340909
4	5476	2	8	1455	162.0	162.0	39	12257	0.0	0.000000

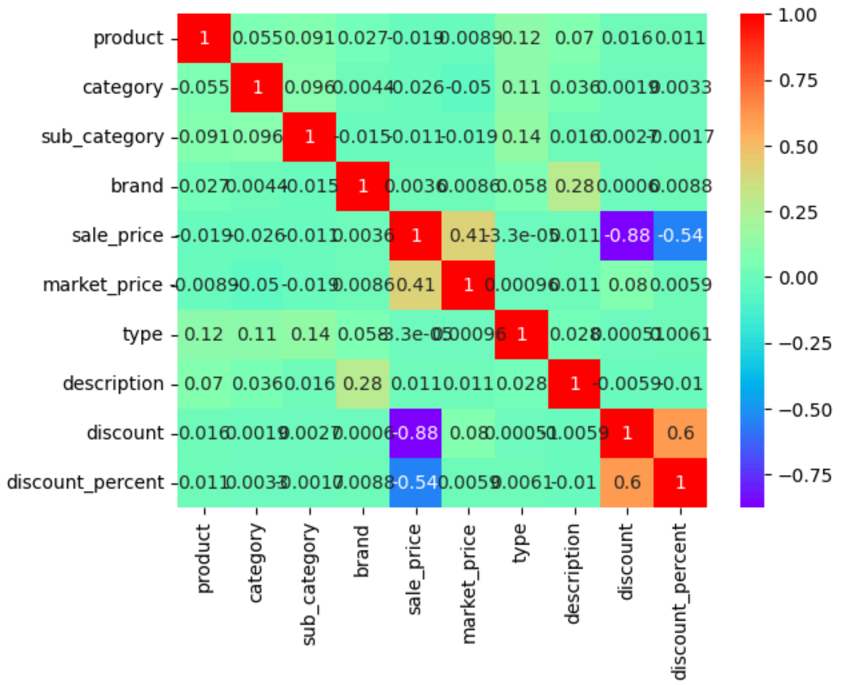
```
data.corr()
```

	product	category	sub_category	brand	sale_price	market_price	type	description	discount	disco
product	1.000000	0.054974	0.090789	0.027145	-0.018899	-0.008874	0.118095	0.070292	0.015992	
category	0.054974	1.000000	0.095718	0.004410	-0.025824	-0.049709	0.109048	0.036129	0.001946	
sub_category	0.090789	0.095718	1.000000	-0.014679	-0.011478	-0.018660	0.141230	0.015513	0.002673	
brand	0.027145	0.004410	-0.014679	1.000000	0.003587	0.008561	0.058388	0.280579	0.000595	
sale_price	-0.018899	-0.025824	-0.011478	0.003587	1.000000	0.411104	-0.000033	0.010581	-0.875935	
market_price	-0.008874	-0.049709	-0.018660	0.008561	0.411104	1.000000	0.000965	0.010701	0.079676	
type	0.118095	0.109048	0.141230	0.058388	-0.000033	0.000965	1.000000	0.028036	0.000510	
description	0.070292	0.036129	0.015513	0.280579	0.010581	0.010701	0.028036	1.000000	-0.005924	
discount	0.015992	0.001946	0.002673	0.000595	-0.875935	0.079676	0.000510	-0.005924	1.000000	
discount_percent	0.011336	0.003295	-0.001748	0.008768	-0.543388	0.005936	0.006112	-0.010373	0.597336	

HEATMAP OF CORRELATION DATA

```
sns.heatmap(data.corr(), cmap="rainbow", annot=True)
```

```
<Axes: >
```



JOINT PLOT OF SALE PRICE VS MARET PRICE

```
sns.jointplot(data=data, x='sale_price', y='market_price')
```

Show hidden output

SCATTER PLOT OF SALE VS MARKET PRICE

```
sns.scatterplot(data=data, x='market_price', y='sale_price')
```

Show hidden output

DIST PLOT OF SALES PRICE

```
sns.distplot(data["sale_price"])
```

Show hidden output

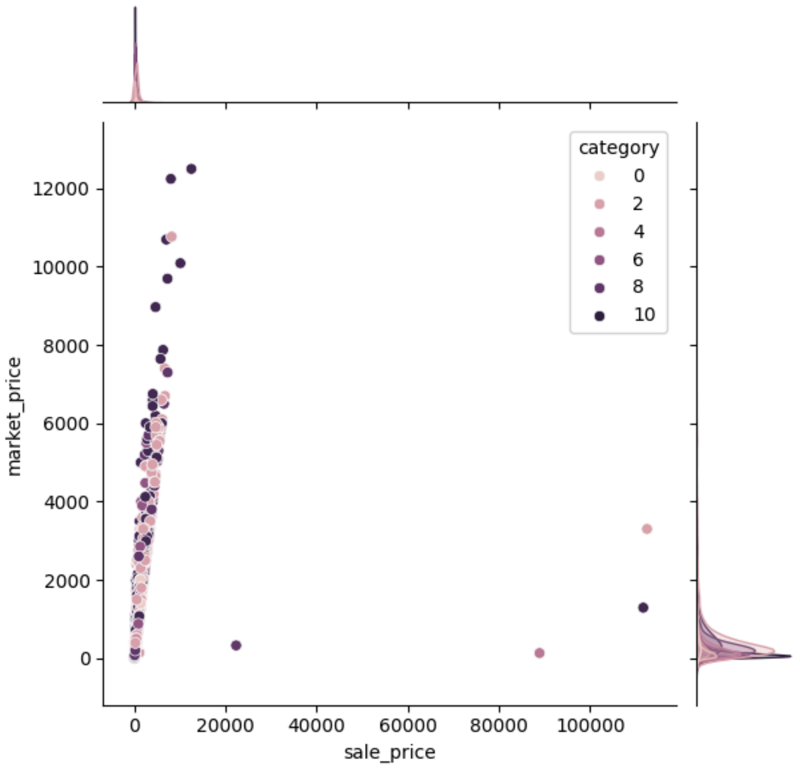
PAIR PLOT OF DATA

```
sns.pairplot(data)
plt.show()
```

Show hidden output

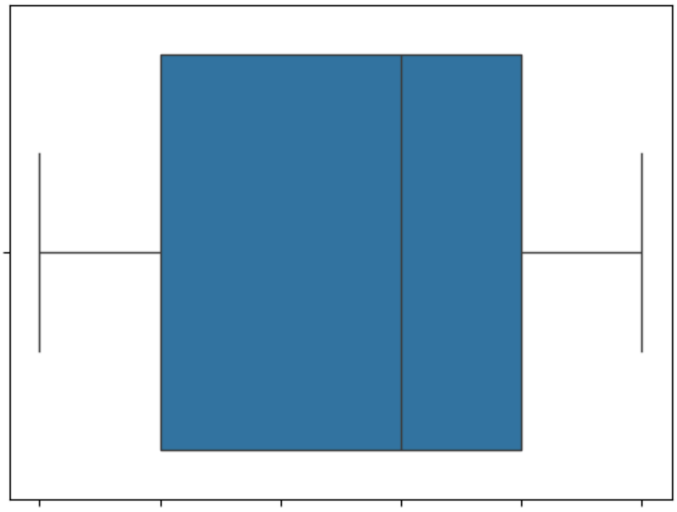
JOINT PLOT OF SALE VS MARKET PRICE

```
sns.jointplot(data=data, x='sale_price', y='market_price', hue="category")
plt.show()
```



BOXPLOT OF CATEGORY

```
sns.boxplot(data=data, x='category')
plt.show()
```



```
data.head()
```

```
     product  category  sub category  brand  sale price  market price  type  description  discount  discount percent
```