```
import pandas as pd  # For data manipulation
import numpy as np   # For numerical operations
import matplotlib.pyplot as plt  # For visualization
import seaborn as sns
```

Step1. Load Dataset

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

	index	product	category	sub_category	brand	sale_price	m
0	1	Garlic Oil - Vegetarian Capsule 500 mg	Beauty & Hygiene	Hair Care	Sri Sri Ayurveda	220.00	
1	2	Water Bottle - Orange	Kitchen, Garden & Pets	Storage & Accessories	Mastercook	180.00	
2	3	Brass Angle Deep - Plain, No.2	Cleaning & Household	Pooja Needs	Trm	119.00	
3	4	Cereal Flip Lid Container/Storage Jar - Assort...	Cleaning & Household	Bins & Bathroom Ware	Nakoda	149.00	
4	5	Creme Soft Soap - For Hands & Body	Beauty & Hygiene	Bath & Hand Wash	Nivea	162.00	
...	...	...	...	...	...	...	
27550	27551	Wottagirl! Perfume Spray - Heaven, Classic	Beauty & Hygiene	Fragrances & Deos	Layerr	199.20	

```
data.head()
```

	index	product	category	sub_category	brand	sale_price	marke
0	1	Garlic Oil - Vegetarian Capsule 500 mg	Beauty & Hygiene	Hair Care	Sri Sri Ayurveda	220.0	
1	2	Water Bottle - Orange	Kitchen, Garden & Pets	Storage & Accessories	Mastercook	180.0	
2	3	Brass Angle Deep - Plain, No.2	Cleaning & Household	Pooja Needs	Trm	119.0	

```
data.tail()
```

	index	product	category	sub_category	brand	sale_price	market_p
27550	27551	Wottagirl! Perfume Spray - Heaven, Classic	Beauty & Hygiene	Fragrances & Deos	Layerr	199.20	
27551	27552	Rosemary	Gourmet & World Food	Cooking & Baking Needs	Puramate	67.50	
27552	27553	Peri-Peri Sweet	Gourmet & World	Snacks, Dry	FabBox	200.00	

step 2: Use head function to look for first 12 rows.

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

	index	product	category	sub_category	brand	sale_price	mar
0	1	Garlic Oil - Vegetarian Capsule 500 mg	Beauty & Hygiene	Hair Care	Sri Sri Ayurveda	220.0	
1	2	Water Bottle - Orange	Kitchen, Garden & Pets	Storage & Accessories	Mastercook	180.0	
2	3	Brass Angle Deep - Plain, No.2	Cleaning & Household	Pooja Needs	Trm	119.0	
3	4	Cereal Flip Lid Container/Storage Jar - Assort...	Cleaning & Household	Bins & Bathroom Ware	Nakoda	149.0	
4	5	Creme Soft Soap - For Hands & Body	Beauty & Hygiene	Bath & Hand Wash	Nivea	162.0	
5	6	Germ - Removal Multipurpose Wipes	Cleaning & Household	All Purpose Cleaners	Nature Protect	169.0	
6	7	Multani Mati	Beauty & Hygiene	Skin Care	Satinance	58.0	

```
data.shape
```

```
(27555, 10)
```

```
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
```

STEP 3:GET DESCRIPTION OF THE DATA IN DATAFRAME

```
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.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

```
data.notnull().sum()
```

	0
index	27555
product	27554
category	27555
sub_category	27555
brand	27554
sale_price	27549
market_price	27555
type	27555
rating	18919
description	27440

dtype: int64

```
data["sale_price"][2:7]
```

	sale_price
2	119.0
3	149.0
4	162.0
5	169.0
6	58.0

dtype: float64

```
data["category"]= data["category"].map({"Beauty & Hygiene":1,"Cleaning & Househ
```

```
data.head(10)
```

	index	product	category	sub_category	brand	sale_price	marke
0	1	Garlic Oil - Vegetarian Capsule 500 mg	1.0	Hair Care	Sri Sri Ayurveda	220.0	
1	2	Water Bottle - Orange	NaN	Storage & Accessories	Mastercook	180.0	
2	3	Brass Angle Deep - Plain, No.2	2.0	Pooja Needs	Trm	119.0	
3	4	Cereal Flip Lid Container/Storage Jar - Assort...	2.0	Bins & Bathroom Ware	Nakoda	149.0	
4	5	Creme Soft Soap - For Hands & Body	1.0	Bath & Hand Wash	Nivea	162.0	
5	6	Germ - Removal Multipurpose Wipes	2.0	All Purpose Cleaners	Nature Protect	169.0	

```
data["category"].mean()
```

```
np.float64(1.2537469170935307)
```

```
data["category"] = data["category"].fillna(data["category"].mean())
```

```
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  float64
 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(4), int64(1), object(5)
memory usage: 2.1+ MB
```

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

	index	product	category	sub_category	brand	sale_price	mar
0	1	Garlic Oil - Vegetarian Capsule 500 mg	1.000000	Hair Care	Sri Sri Ayurveda	220.0	
1	2	Water Bottle - Orange	1.253747	Storage & Accessories	Mastercook	180.0	
2	3	Brass Angle Deep - Plain, No.2	2.000000	Pooja Needs	Trm	119.0	
3	4	Cereal Flip Lid Container/Storage Jar - Assort...	2.000000	Bins & Bathroom Ware	Nakoda	149.0	
4	5	Creme Soft Soap - For Hands & Body	1.000000	Bath & Hand Wash	Nivea	162.0	
5	6	Germ - Removal Multipurpose Wipes	2.000000	All Purpose Cleaners	Nature Protect	169.0	
6	7	Multani Mati	1.000000	Skin Care	Satinance	58.0	

```
data["sale_price"].mode()
```

	sale_price
0	99.0

dtype: float64

```
data["sale_price"].mode()[0]
```

```
np.float64(99.0)
```

```
data["sale_price"]=data["sale_price"].fillna(data["sale_price"].mode()[0])
```

```
data.isnull().sum()/len(data)*100
```

	0
index	0.000000
product	0.003629
category	0.000000
sub_category	0.000000
brand	0.003629
sale_price	0.000000
market_price	0.000000
type	0.000000
rating	31.340954
description	0.417347

dtype: float64

```
data["sale_price"]= data["sale_price"].astype("int64")
```

STEP 4 :Find out the Missing Values from the Dataset.

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

```
missing_values = missing_values[missing_values > 0]
print(missing_values)
```

```
product           1
brand             1
rating         8636
description     115
dtype: int64
```

```
df[df.isnull().any(axis=1)]
```

	index	product	category	sub_category	brand	sale_price	mar
55	56	Soothing Cucumber Facial Scrub With Apricot Seeds	1.000000	Skin Care	TJORI	299	
59	60	Corporate Planner Diary With Premium PU Leathe...	2.000000	Stationery	Prozo Plus	399	
65	66	Ayurvedic Anti-Tan Face Pack	1.000000	Skin Care	TJORI	269	
68	69	Organic Carom Seeds/Ajwain/Om Kalu	1.253747	Masalas & Spices	Earthon	72	
69	70	Padded Harness - 3/4 inch, Grey Colour	1.253747	Pet Food & Accessories	Glenand	840	
...	...	...	...	...	...	...	
27509	27510	Deluxe Crackers - Veg	1.253747	Chocolates & Biscuits	Kerk	150	
27511	27512	Specialist Stain Remover Pen & Marker	2.000000	All Purpose Cleaners	365	449	
27514	27515	Verge & Sheer Perfume For Pair	1.000000	Fragrances & Deos	Skinn by Titan	1615	
		Tick'et to					

```
df[df['rating'].isnull()]
```

	index	product	category	sub_category	brand	sale_price	mar
55	56	Soothing Cucumber Facial Scrub With Apricot Seeds	1.000000	Skin Care	TJORI	299	
59	60	Corporate Planner Diary With Premium PU Leathe...	2.000000	Stationery	Prozo Plus	399	
65	66	Ayurvedic Anti-Tan Face Pack	1.000000	Skin Care	TJORI	269	
68	69	Organic Carom Seeds/Ajwain/Om Kalu	1.253747	Masalas & Spices	Earthon	72	
69	70	Padded Harness - 3/4 inch, Grey Colour	1.253747	Pet Food & Accessories	Glenand	840	
...	...	...	...	...	...	...	
27509	27510	Deluxe Crackers - Veg	1.253747	Chocolates & Biscuits	Kerk	150	
27511	27512	Specialist Stain Remover Pen & Marker	2.000000	All Purpose Cleaners	365	449	
27514	27515	Verge & Sheer Perfume For Pair	1.000000	Fragrances & Deos	Skinn by Titan	1615	
		Tick'et to					

```
df['rating'] = df['rating'].fillna(df['rating'].mean()   # Mean
df['rating'] = df['rating'].fillna(df['rating'].median() # Median
```

```
df['category'] = df['category'].fillna(df['category'].mode()[0])
```

```
df.fillna(method='ffill', inplace=True)  # forward fill
df.fillna(method='bfill', inplace=True)  # backward fill
```

```
<ipython-input-71-bb1640f83259>:1: FutureWarning: DataFrame.fillna with 'me
  df.fillna(method='ffill', inplace=True)  # forward fill
<ipython-input-71-bb1640f83259>:2: FutureWarning: DataFrame.fillna with 'me
  df.fillna(method='bfill', inplace=True)  # backward fill
```

```
df.isnull().sum()  # to confirm they're gone
df.to_csv("filled_dataset.csv", index=False)  # save it
```

```
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       27555 non-null  object
 2   category      27555 non-null  float64
 3   sub_category  27555 non-null  object
 4   brand         27555 non-null  object
 5   sale_price    27555 non-null  int64
 6   market_price  27555 non-null  float64
 7   type          27555 non-null  object
 8   rating        18919 non-null  float64
 9   description   27555 non-null  object
dtypes: float64(3), int64(2), object(5)
memory usage: 2.1+ MB
```

STEP 5 :FIND INFO ABOUT THE DATAFRAME

```
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  float64
 3   sub_category  27555 non-null  object
 4   brand         27554 non-null  object
 5   sale_price    27555 non-null  int64
 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(2), object(5)
memory usage: 2.1+ MB
```

```
data["sale_price"]
```

	sale_price
0	220
1	180
2	119
3	149
4	162
...	...
27550	199
27551	67
27552	200
27553	396
27554	214

27555 rows × 1 columns

dtype: int64

```
data["sale_price"].unique()
```

```
array([ 220,  180,  119, ..., 1071, 4500, 1525])
```

```
data["sale_price"].nunique()
```

```
1443
```

```
data.type
```

	type
0	Hair Oil & Serum
1	Water & Fridge Bottles
2	Lamp & Lamp Oil
3	Laundry, Storage Baskets
4	Bathing Bars & Soaps
...	...
27550	Perfume
27551	Herbs, Seasonings & Rubs
27552	Nachos & Chips
27553	Tea Bags
27554	Men's Deodorants

27555 rows × 1 columns

dtype: object

```
data["rating"].value_counts()
```

rating	count
4.2	2249
4.3	2138
4.0	2089
4.1	1767
5.0	1407
4.4	1037
3.9	932
3.8	866
4.5	797

3.0	684
3.7	672
3.5	535
3.6	447
3.3	400
1.0	387
4.6	322
3.4	303
4.7	287
2.0	237
4.8	199
3.2	181
3.1	163
2.5	132
2.8	125
2.7	116
2.3	94
2.9	79
2.6	58
4.9	53
1.5	32
2.4	29
2.2	24
1.8	22
1.7	22
2.1	10
1.3	9
1.4	6
1.9	4
1.6	3
1.2	2

dtype: int64

```
data["sale_price"].mean()
```

```
np.float64(334.5102522228271)
```

```
data["market_price"].mean()
```

```
np.float64(382.05666448920346)
```

```
df = pd.DataFrame(data)
```

STEP 6 :MEASURING DISCOUNT ON A CERTIN ITEM

```
df['discount_percent'] = ((df['market_price'] - df['sale_price']) / df['market_

print(df[['product', 'market_price', 'sale_price', 'discount_percent']])
```

```
                                                product  market_price  \
0                  Garlic Oil - Vegetarian Capsule 500 mg         220.0
1                                     Water Bottle - Orange         180.0
2                         Brass Angle Deep - Plain, No.2         250.0
3      Cereal Flip Lid Container/Storage Jar - Assort...         176.0
4                     Creme Soft Soap - For Hands & Body         162.0
...                                                  ...           ...
27550         Wottagirl! Perfume Spray - Heaven, Classic         249.0
27551                                           Rosemary          75.0
27552                       Peri-Peri Sweet Potato Chips         200.0
27553                          Green Tea - Pure Original         495.0
27554                      United Dreams Go Far Deodorant         390.0

       sale_price  discount_percent
0             220          0.000000
1             180          0.000000
2             119         52.400000
3             149         15.340909
4             162          0.000000
...           ...               ...
27550         199         20.080321
27551          67         10.666667
27552         200          0.000000
27553         396         20.000000
27554         214         45.128205

[27555 rows x 4 columns]
```

```
df.columns
```

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

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27555 entries, 0 to 27554
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   index             27555 non-null  int64
 1   product           27554 non-null  object
 2   category          27555 non-null  float64
 3   sub_category      27555 non-null  object
 4   brand             27554 non-null  object
 5   sale_price        27555 non-null  int64
 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
 10  discount_percent  27555 non-null  float64
dtypes: float64(4), int64(2), object(5)
memory usage: 2.3+ MB
```

STEP 7 :FIND OUT TOP AND LEAST SOLD PRODUCTS

```
top_selling = df.sort_values(by='sale_price', ascending=False)
print("Top selling products:")
print(top_selling.head())  # Or use head(1) for the single top product
```

```
Top selling products:
       index                                          product  category
1249    1250                                        Beard Kit  1.000000
248      249  4mm Aluminium Induction Base Chapati Roti Tawa...  1.253747
436      437                       Balloon - Polka Dot, 12 Inch  2.000000
288      289          Arrabbiata Tomato Pasta Sauce With Chilli  1.253747
25301  25302                                   Bravura Clipper  1.253747

                 sub_category       brand  sale_price  market_price  \
1249           Men's Grooming  Uncle Tony      112475        3300.0
248      Cookware & Non Stick       HAZEL      111649        1289.0
436     Party & Festive Needs    B Vishal       88899         129.0
288    Sauces, Spreads & Dips   Montanini       22325         325.0
25301  Pet Food & Accessories        Wahl       12500       12500.0

                           type  rating  \
1249         Combos & Gift Sets     NaN
248            Tawa & Sauce Pan     NaN
436    Caps, Balloons & Candles     3.9
288     Mustard & Cheese Sauces     5.0
25301   Pet Cleaning & Grooming     NaN

                                            description  discount_percent
1249   The combination of a beard oil, a beard wash, ...      -3308.333333
248    Hazel Aluminium Tawa has an ergonomic design f...      -8561.675718
436    Whether it is a party in the office, a Christm...     -68813.953488
288                                                  NaN      -6769.230769
25301  The bravura clipper is a must-have clipper for...          0.000000
```

```
top_selling[["sale_price","product"]].head()
```

	sale_price	product
1249	112475	Beard Kit
248	111649	4mm Aluminium Induction Base Chapati Roti Tawa...
436	88899	Balloon - Polka Dot, 12 Inch
288	22325	Arrabbiata Tomato Pasta Sauce With Chilli
25301	12500	Bravura Clipper

```
top_selling[["sale_price","product"]].tail()
```

	sale_price	product
14184	5	Tomato - Local, Organically Grown
2761	5	Orbit Sugar-Free Chewing Gum - Lemon & Lime
21228	5	Dish Shine Bar
21312	3	Serum
26976	2	Curry Leaves

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

```
column = 'sale_price'  # Change this to any column you want

Q1 = df[column].quantile(0.25)
Q3 = df[column].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
```

```
df# Replace 'your_column' with the column you're analyzing
Q1 = df['market_price'].quantile(0.25)
Q3 = df['sale_price'].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Get outliers
outliers = df[(df['market_price'] < lower_bound) | (df['sale_price'] > upper_bo
```

```
outliers = df[(df[column] < lower_bound) | (df[column] > upper_bound)]
print(outliers)
```

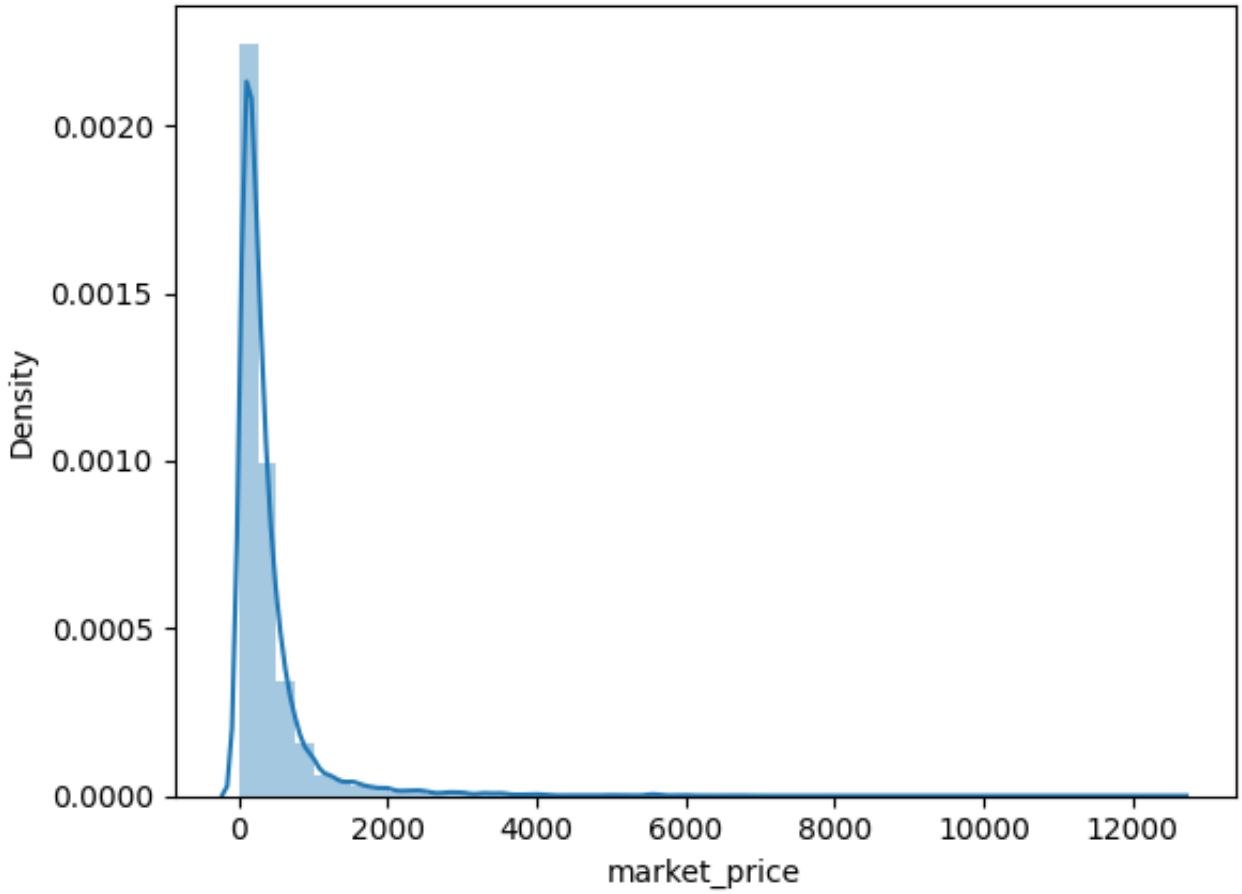
```
       index                                          product  category
8          9  Biotin & Collagen Volumizing Hair Shampoo + Bi...  1.000000
47        48                              Colour Catcher Sheets  2.000000
51        52                                        Peach Syrup  1.253747
69        70             Padded Harness - 3/4 inch, Grey Colour  1.253747
91        92    Hard Anodised Ezee-Pour Saucepan With Lid - L88  1.253747
...      ...                                                ...       ...
27498  27499            Juicer - Fruit & Vegetable, Light Green  1.253747
27505  27506                                Virgin Coconut Oil  1.253747
27514  27515                     Verge & Sheer Perfume For Pair  1.000000
27538  27539  Quista Pro Advanced Whey Protein Formula forti...  1.000000
27542  27543                    Embellish Skin Lightening Serum  1.000000

                 sub_category            brand  sale_price  market_price  \
8                   Hair Care       StBotanica        1098        1098.0
47       All Purpose Cleaners           Dylon         799         799.0
51         Drinks & Beverages           Pekers         850         850.0
69     Pet Food & Accessories          Glenand         840         840.0
91       Cookware & Non Stick   Hawkins Futura         864         910.0
...                       ...              ...         ...           ...
27498     Kitchen Accessories           Ganesh        1071        1071.0
27505      Edible Oils & Ghee          Merkera         875         875.0
27514       Fragrances & Deos   Skinn by Titan        1615        1795.0
27538       Health & Medicine         Himalaya        4500        4500.0
27542               Skin Care  Organic Harvest        1525        1795.0

                                   type  rating  \
8                  Shampoo & Conditioner     3.5
47                     Imported Cleaners     4.0
51              Gourmet Juices & Drinks     4.2
69                Pet Collars & Leashes     NaN
91                     Tawa & Sauce Pan     4.6
...                                 ...     ...
27498  Kitchen Tools & Other Accessories     2.0
27505                  Other Edible Oils     NaN
27514                            Perfume     NaN
27538            Supplements & Proteins     4.0
27542                         Face Care     4.2

                                          description  discount_percent
8      An exclusive blend with Vitamin B7 Biotin, Hyd...          0.000000
47     1. Prevents Colour Run Accidents Colours that ...          0.000000
51     Pekers peach syrup takes you on a historical t...          0.000000
69     These are soft padded harness for your active ...          0.000000
91     Futura Hard Anodised Saucepan comes with a spo...          5.054945
...                                                  ...               ...
27498  This juicer comes in various attractive colour...          0.000000
27505  Merkera Extra Virgin Coconut Oil 100% natural,...          0.000000
27514  VERGE for men paints a picture of a classy out...         10.027855
27538  Quista Pro is a whey protein blend that helps ...          0.000000
27542  Achieve an everlasting illuminated skin by inc...         15.041783

[2205 rows x 11 columns]
```

STEP 9 :CREATING PLOTS AND VISUALISATIONS

PLOTTING

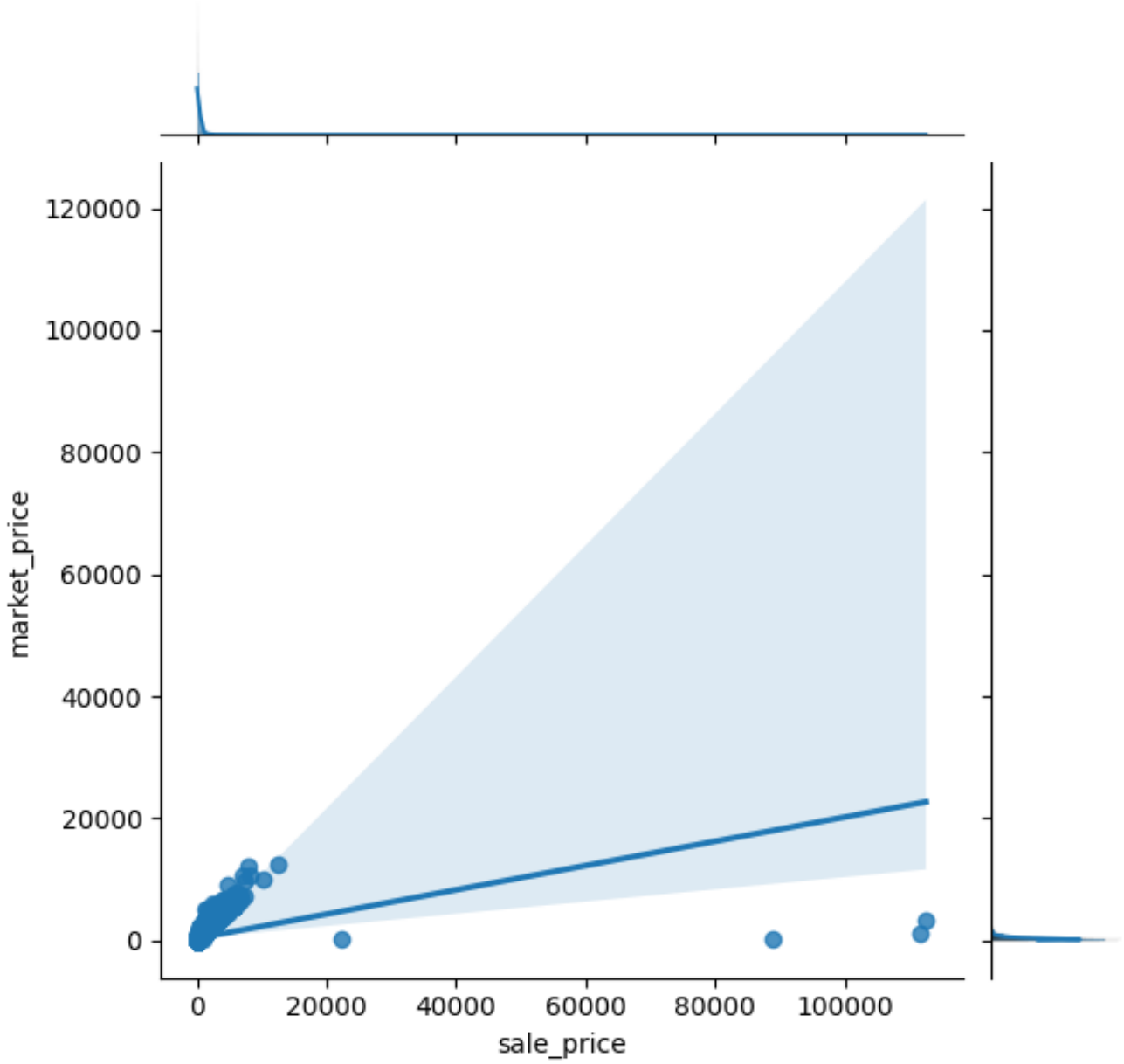
```
sns.distplot(data["market_price"])
```

```
<ipython-input-53-940609e59e7b>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function wit
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data["market_price"])
<Axes: xlabel='market_price', ylabel='Density'>
```



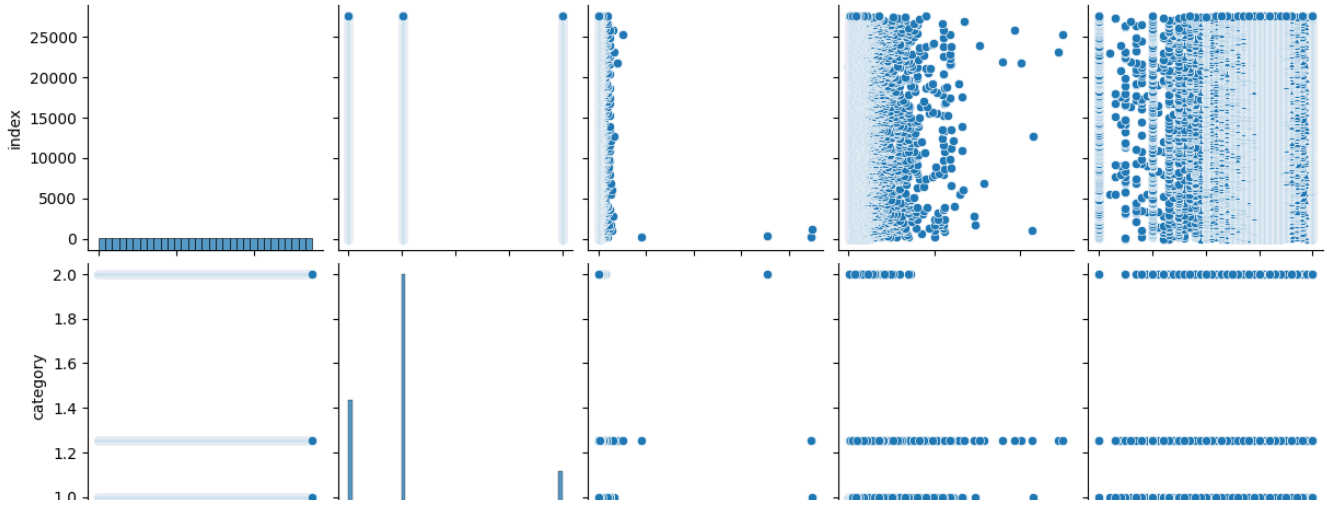
```
sns.jointplot(x="sale_price",y="market_price",data=data,kind="reg")
```

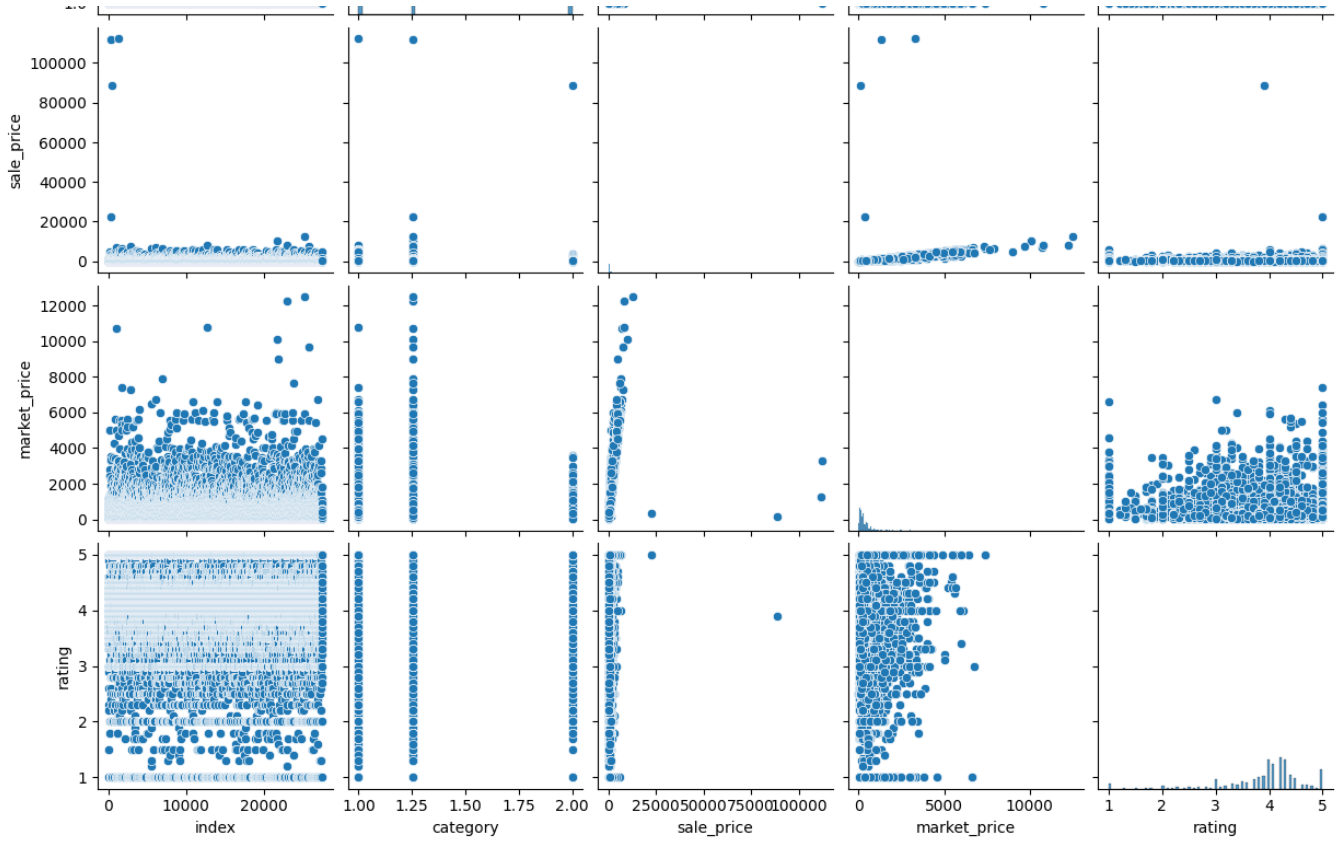
```
<seaborn.axisgrid.JointGrid at 0x78b19dfcd110>
```




```
sns.pairplot(data)
```

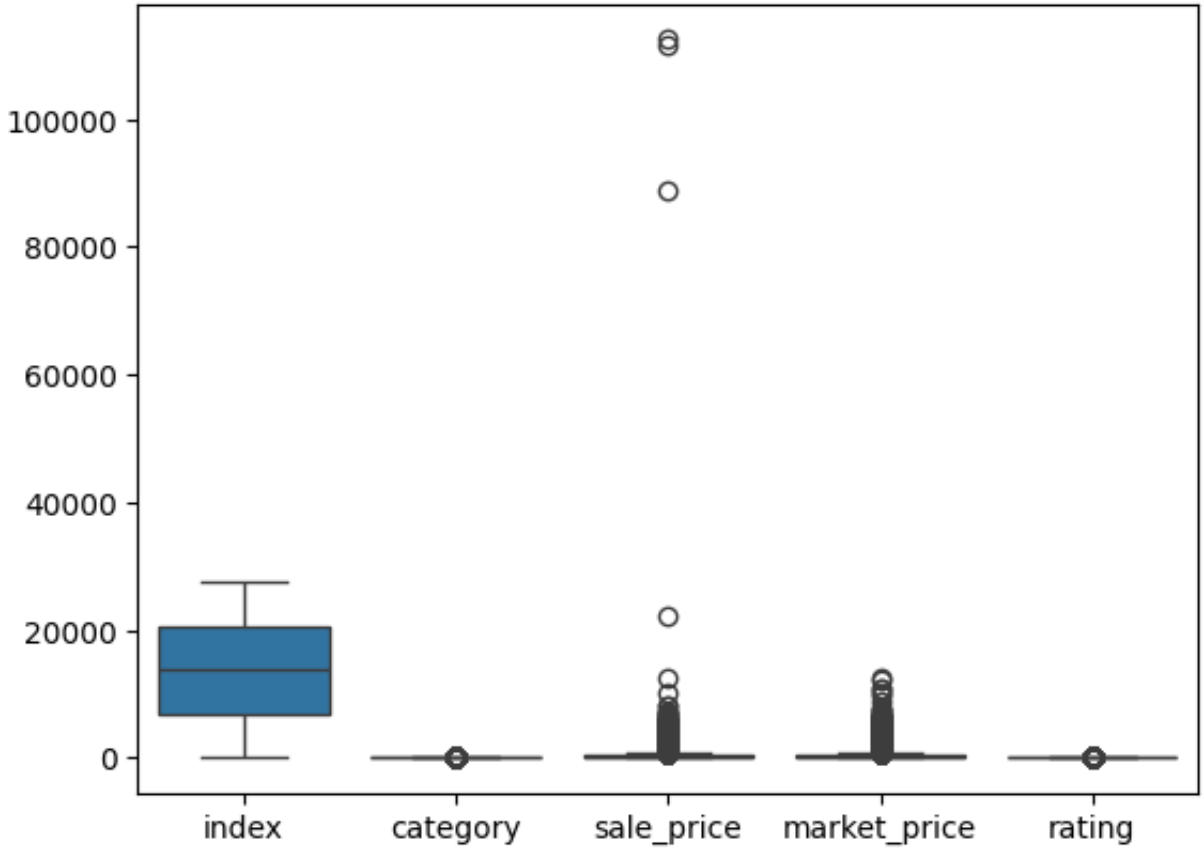
```
<seaborn.axisgrid.PairGrid at 0x78b1960fbf50>
```




sale_price
market_price
rating
index
category
sale_price
market_price
rating

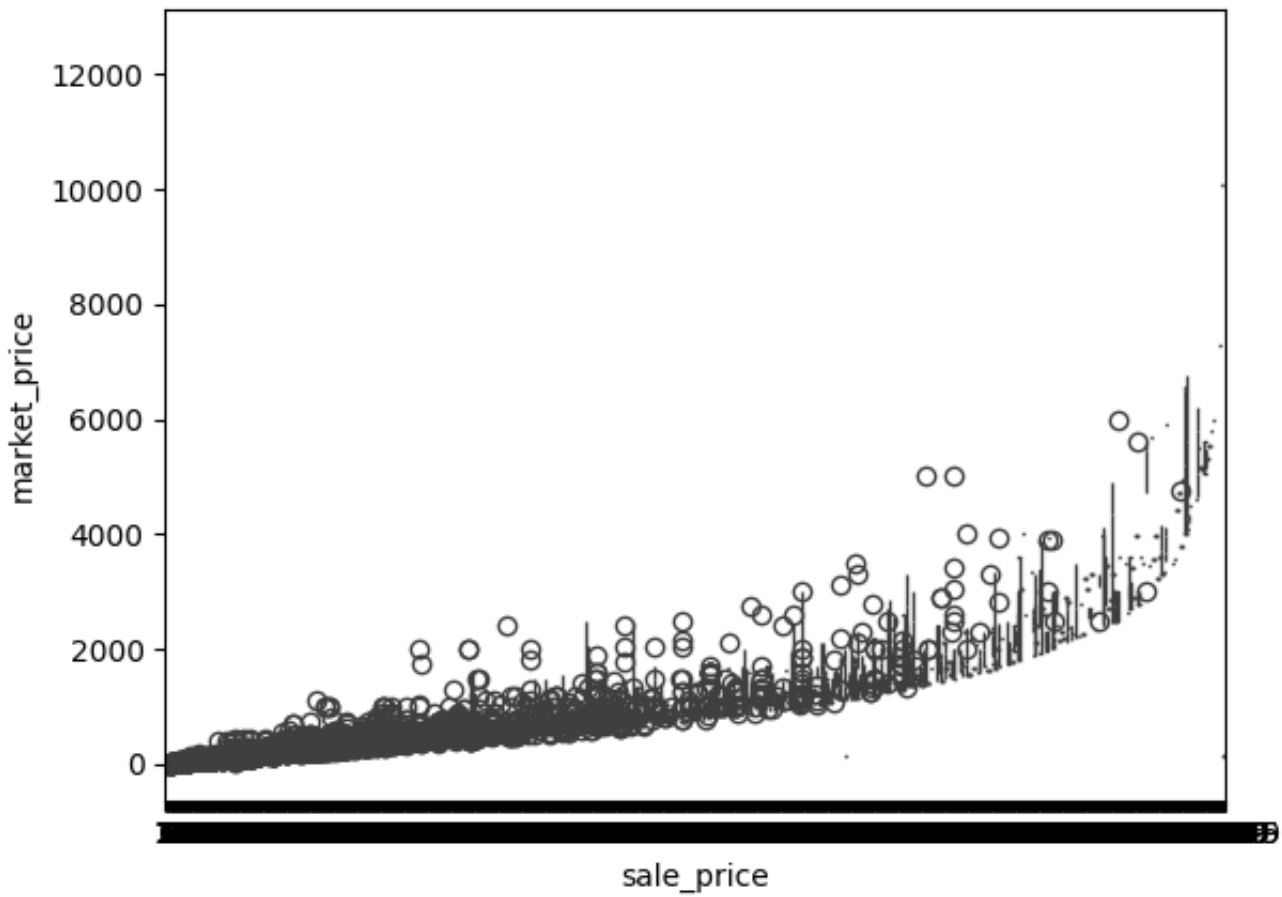
```
sns.boxplot(data=data)
```

```
<Axes: >
```



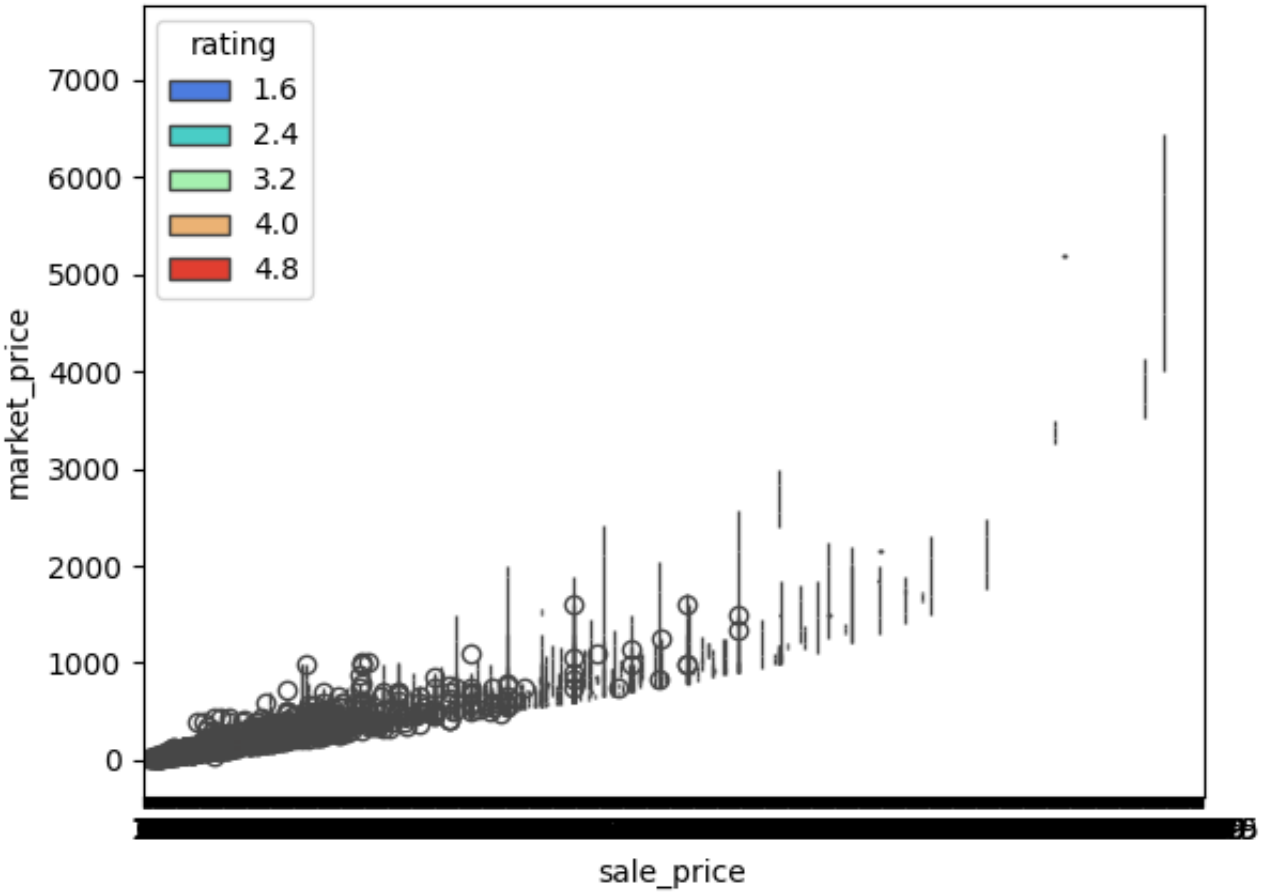
```
sns.boxplot(x="sale_price",y="market_price",data=data)
```

```
<Axes: xlabel='sale_price', ylabel='market_price'>
```



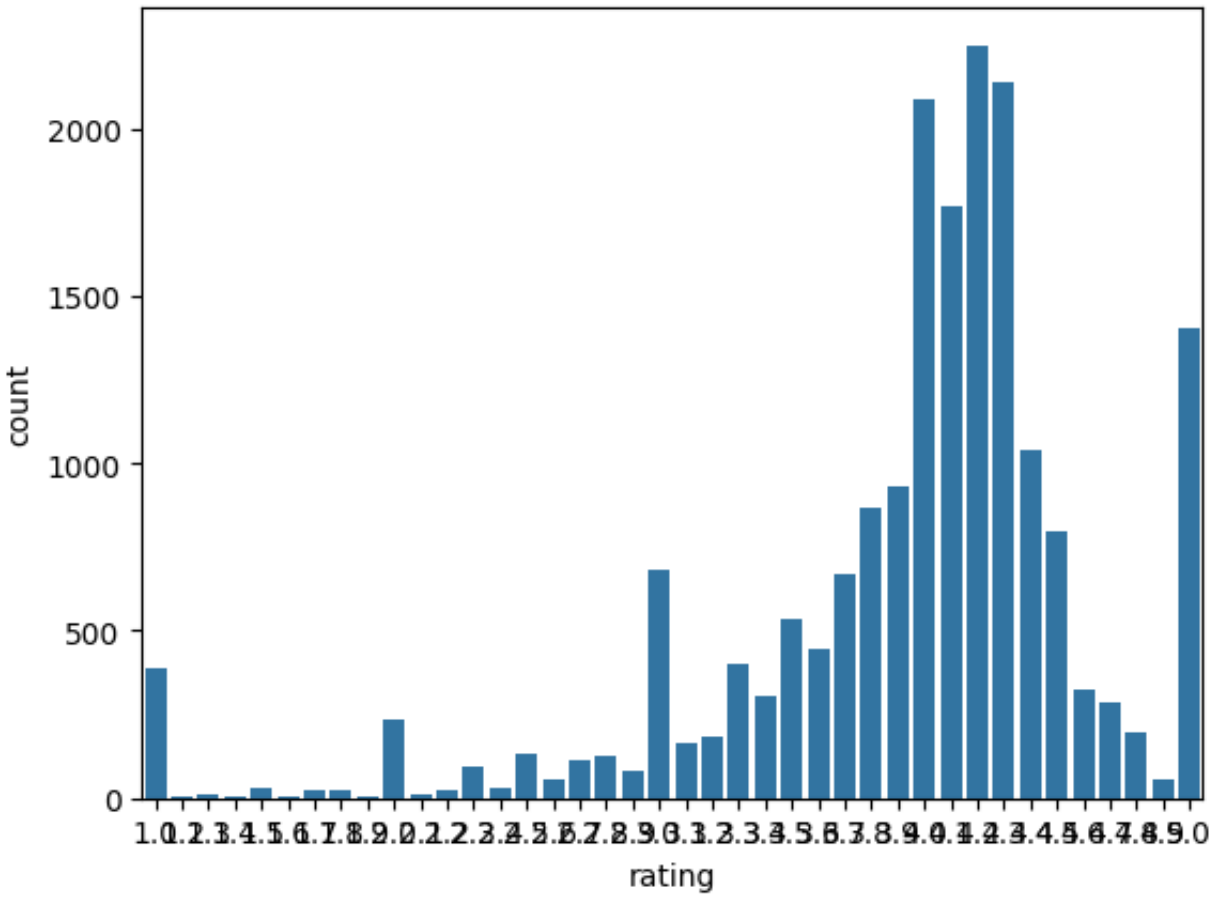
```
sns.boxplot(x="sale_price",y="market_price",data=data,hue="rating",palette="rai
```

```
<Axes: xlabel='sale_price', ylabel='market_price'>
/usr/local/lib/python3.11/dist-packages/IPython/core/events.py:89: UserWarn
  func(*args, **kwargs)
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: Use
  fig.canvas.print_figure(bytes_io, **kw)
```



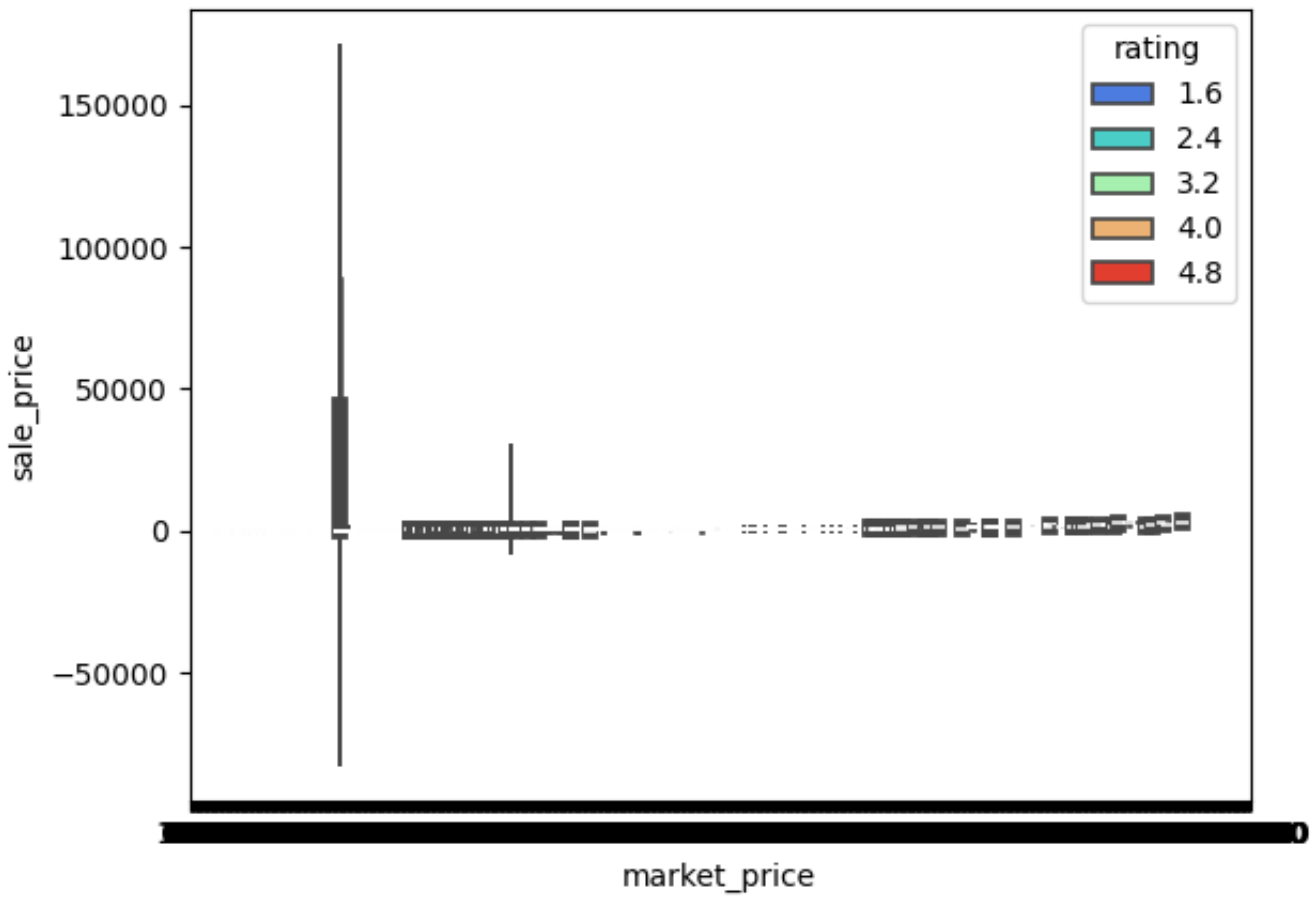
```
sns.countplot(x="rating",data=data)
```

```
<Axes: xlabel='rating', ylabel='count'>
```



```
sns.violinplot(x="market_price",y="sale_price",data=data,hue="rating",palette='
```

```
<Axes: xlabel='market_price', ylabel='sale_price'>
```



```
sns.barplot(x="market_price",y="sale_price",data=data)
```

```
<Axes: xlabel='market_price', ylabel='sale_price'>
```