

FlipKart Iphone Sales Analysis

Apple iPhones are among the top-selling smartphones worldwide. There is huge competition among smartphone brands in India, where you can get the latest technology in a smartphone at half the price of an iPhone. Still, there are high sales of iPhones in India. Lets Analyse iphone sales in India with the flipkart data.

Import libraries

```
In [11]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
```

Loading Dataset

```
In [2]: # Load the dataset
df = pd.read_csv(r'C:\Users\HP\Desktop\Python\Project\apple_products.csv')
```

Understanding Data

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

Out[4]:

	Product Name	Product URL	Brand	Sale Price	Mrp	Discount Percentage	Number Of Ratings	Number Of Reviews	Upc	Star Rating	Ram
0	APPLE iPhone 8 Plus (Gold, 64 GB)	https://www.flipkart.com/apple-iphone-8-plus-g...	Apple	49900	49900	0	3431	356	MOBEXRGV7EHHTGUH	4.6	2 GB
1	APPLE iPhone 8 Plus (Space Grey, 256 GB)	https://www.flipkart.com/apple-iphone-8-plus-s...	Apple	84900	84900	0	3431	356	MOBEXRGVAC6TJT4F	4.6	2 GB
2	APPLE iPhone 8 Plus (Silver, 256 GB)	https://www.flipkart.com/apple-iphone-8-plus-s...	Apple	84900	84900	0	3431	356	MOBEXRGVGETABXWZ	4.6	2 GB
3	APPLE iPhone 8 (Silver, 256 GB)	https://www.flipkart.com/apple-iphone-8-silver...	Apple	77000	77000	0	11202	794	MOBEXRGVMZWUHCBA	4.5	2 GB
4	APPLE iPhone 8 (Gold, 256 GB)	https://www.flipkart.com/apple-iphone-8-gold-2...	Apple	77000	77000	0	11202	794	MOBEXRGVPK7PFEJZ	4.5	2 GB

Each row seems to represent a different product, and the columns provide various details about each product. Here's a breakdown of what each column might represent:

Product Name: The name or title of the product.

Product URL: The URL where the product can be found, possibly on an online store or website.

Brand: The brand or manufacturer of the product.

Sale Price: The price at which the product is being sold.

Mrp: Maximum Retail Price, which is the highest price at which the product is usually sold.

Discount Percentage: The percentage of discount offered on the product.

Number Of Ratings: The total number of ratings or reviews the product has received.

Number Of Reviews: The total number of written reviews or feedback the product has received.

Upc: Universal Product Code, a unique identifier for the product.

Star Rating: The average star rating of the product based on user reviews.

Ram: This could represent the RAM (Random Access Memory) specification of the product, especially if the products are electronic devices like computers or smartphones.

Before moving forward, let's have a quick look at whether this dataset contains any null values or not:

```
In [6]: print(df.isnull().sum())
```

```
Product Name      0
Product URL       0
Brand             0
Sale Price        0
Mrp               0
Discount Percentage 0
Number Of Ratings 0
Number Of Reviews 0
Upc               0
Star Rating       0
Ram              0
dtype: int64
```

The dataset doesn't have any null values.

Descriptive statistics of the data:

```
In [7]: print(df.describe())
```

	Sale Price	Mrp	Discount Percentage	Number Of Ratings \
count	62.000000	62.000000	62.000000	62.000000
mean	80073.887097	88058.064516	9.951613	22420.403226
std	34310.446132	34728.825597	7.608079	33768.589550
min	29999.000000	39900.000000	0.000000	542.000000
25%	49900.000000	54900.000000	6.000000	740.000000
50%	75900.000000	79900.000000	10.000000	2101.000000
75%	117100.000000	120950.000000	14.000000	43470.000000
max	140900.000000	149900.000000	29.000000	95909.000000

	Number Of Reviews	Star Rating
count	62.000000	62.000000
mean	1861.677419	4.575806
std	2855.883830	0.059190
min	42.000000	4.500000
25%	64.000000	4.500000
50%	180.000000	4.600000
75%	3331.000000	4.600000
max	8161.000000	4.700000

iPhone Sales in India

Now I will create a new dataframe by storing all the data about the top 10 highest-rated iPhones in India on Flipkart. It will help in understanding what kind of iPhones are liked the most in India:

```
In [8]: highest_rated = df.sort_values(by=["Star Rating"],ascending=False)
highest_rated = highest_rated.head(10)
print(highest_rated['Product Name'])
```

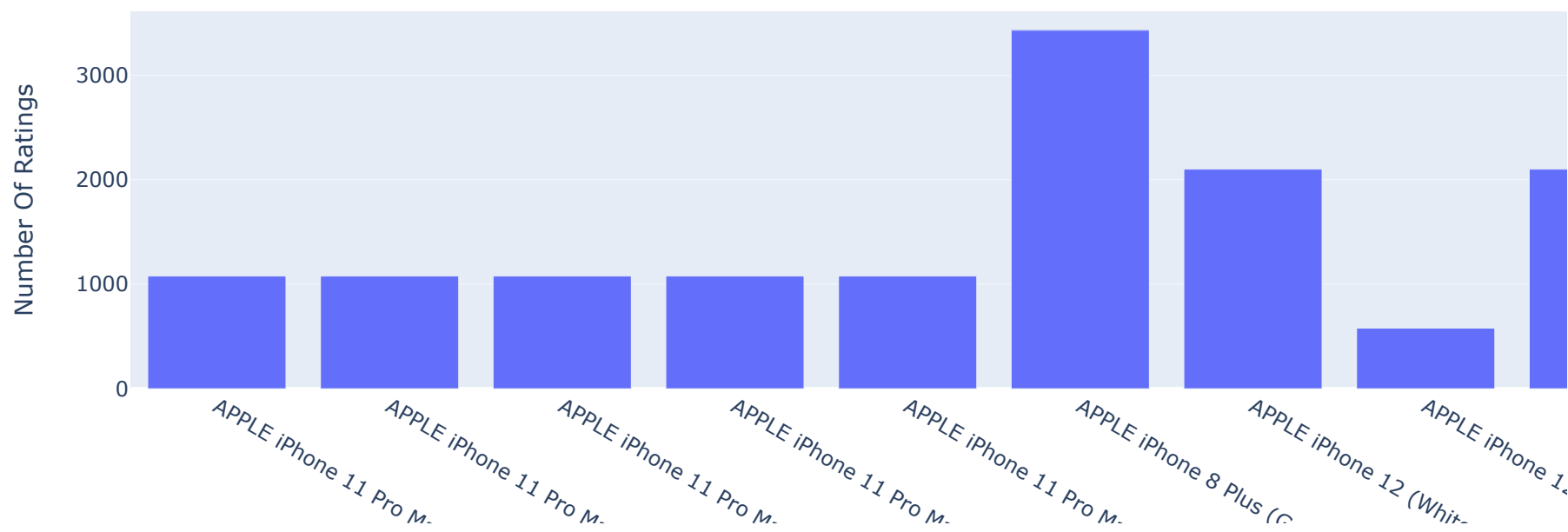
```
20    APPLE iPhone 11 Pro Max (Midnight Green, 64 GB)
17    APPLE iPhone 11 Pro Max (Space Grey, 64 GB)
16    APPLE iPhone 11 Pro Max (Midnight Green, 256 GB)
15    APPLE iPhone 11 Pro Max (Gold, 64 GB)
14    APPLE iPhone 11 Pro Max (Gold, 256 GB)
0     APPLE iPhone 8 Plus (Gold, 64 GB)
29    APPLE iPhone 12 (White, 128 GB)
32    APPLE iPhone 12 Pro Max (Graphite, 128 GB)
35    APPLE iPhone 12 (Black, 128 GB)
36    APPLE iPhone 12 (Blue, 128 GB)
Name: Product Name, dtype: object
```

According to the above data, below are the top 5 most liked iPhones in India: APPLE iPhone 11 Pro Max (Midnight Green, 64 GB) APPLE iPhone 11 Pro Max (Space Grey, 64 GB) APPLE iPhone 11 Pro Max (Midnight Green, 256 GB) APPLE iPhone 11 Pro Max (Gold, 64 GB) APPLE iPhone 11 Pro Max (Gold, 256 GB)

Number of ratings of the highest-rated iPhones on Flipkart:

```
In [12]: iphones = highest_rated["Product Name"].value_counts()
label = iphones.index
counts = highest_rated["Number Of Ratings"]
figure = px.bar(highest_rated, x=label,
                 y = counts,
                 title="Number of Ratings of Highest Rated iPhones")
figure.show()
```

Number of Ratings of Highest Rated iPhones

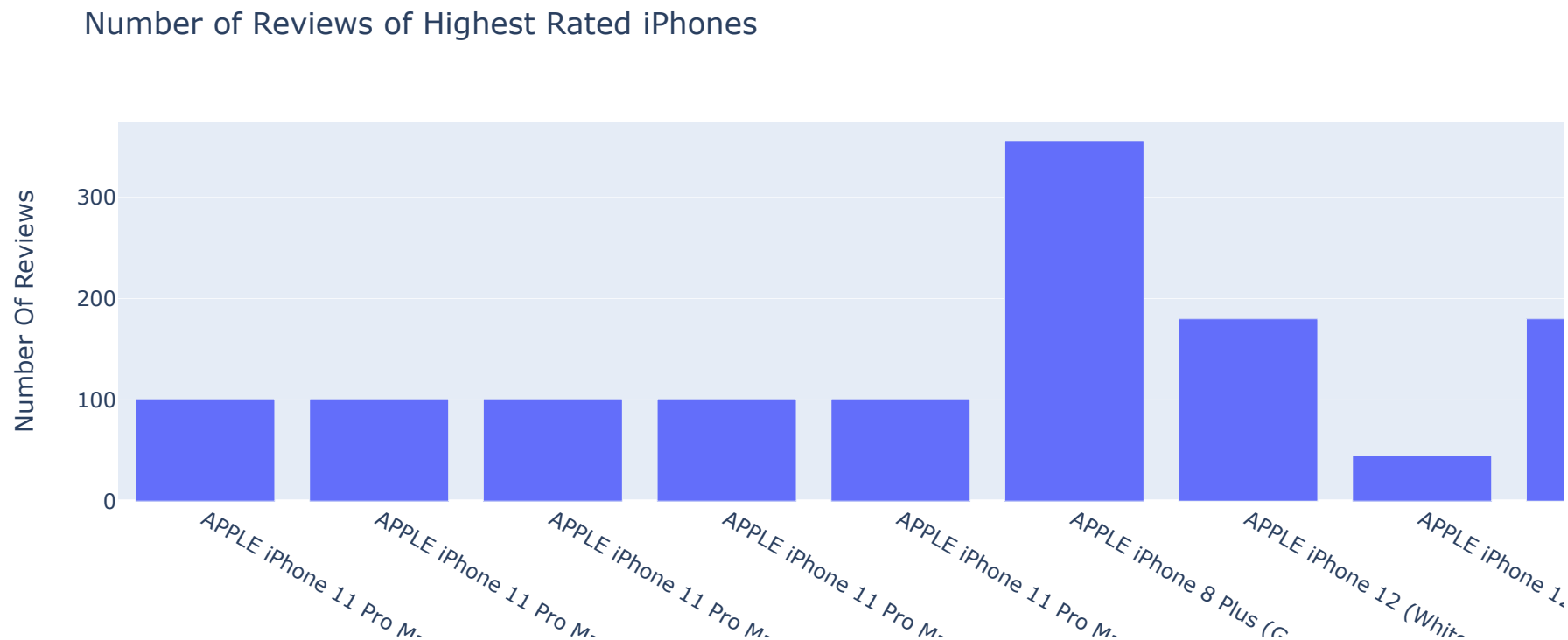


According to the above bar graph, APPLE iPhone 8 Plus (Gold, 64 GB) has the most ratings on Flipkart. Now let's have a look at the

Number of reviews of the highest-rated iPhones on Flipkart:

```
In [13]: iphones = highest_rated["Product Name"].value_counts()
label = iphones.index
counts = highest_rated["Number Of Reviews"]
figure = px.bar(highest_rated, x=label,
                y = counts,
```

```
title="Number of Reviews of Highest Rated iPhones")  
figure.show()
```

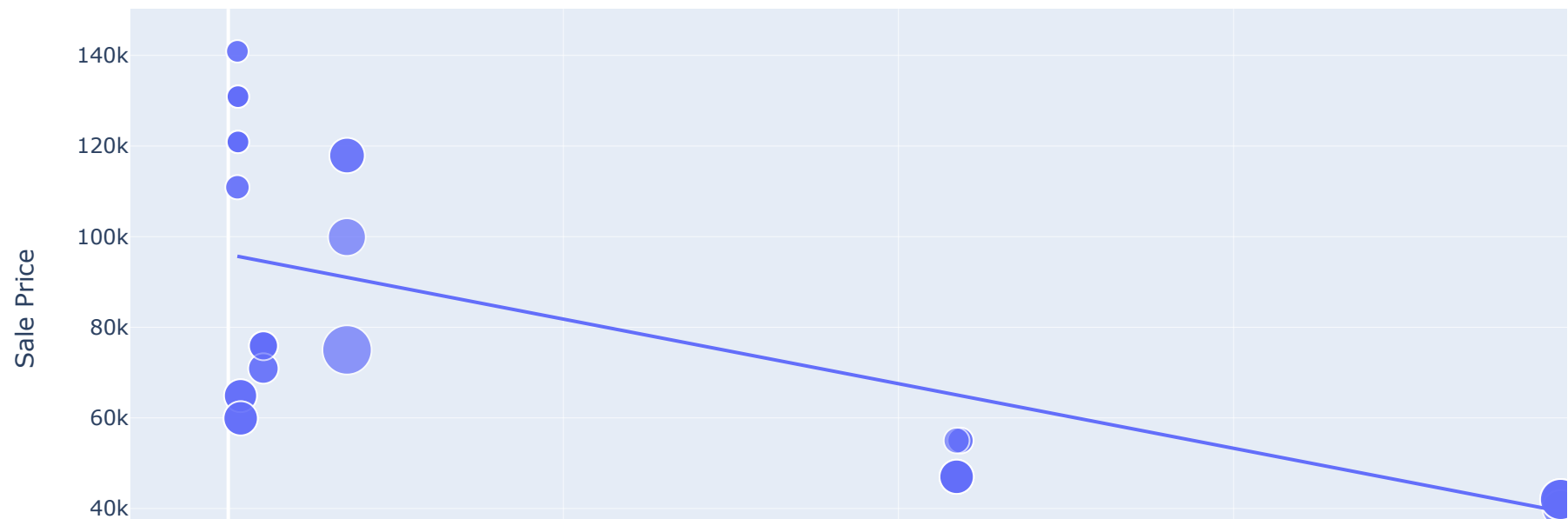


APPLE iPhone 8 Plus (Gold, 64 GB) is also leading in the highest number of reviews on Flipkart among the highest-rated iPhones in India. Now let's have a look at the

Relationship between the sale price of iPhones and their ratings on Flipkart:

```
In [15]: figure = px.scatter(data_frame = df, x="Number Of Ratings",  
                             y="Sale Price", size="Discount Percentage",  
                             trendline="ols",  
                             title="Relationship between Sale Price and Number of Ratings of iPhones")  
figure.show()
```

Relationship between Sale Price and Number of Ratings of iPhones

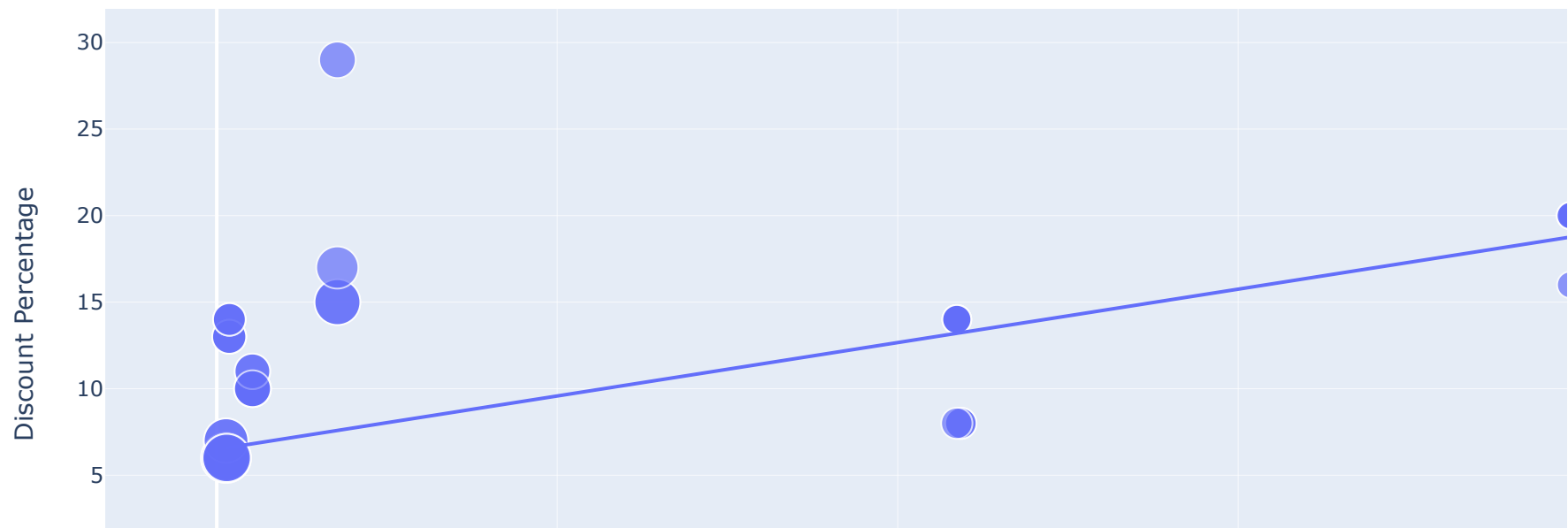


There is a negative linear relationship between the sale price of iPhones and the number of ratings. It means iPhones with lower sale prices are sold more in India. Now let's have a look at the

Relationship between the discount percentage on iPhones on Flipkart and the number of ratings:

```
In [16]: figure = px.scatter(data_frame = df, x="Number Of Ratings",  
                             y="Discount Percentage", size="Sale Price",  
                             trendline="ols",  
                             title="Relationship between Discount Percentage and Number of Ratings of iPhones")  
figure.show()
```

Relationship between Discount Percentage and Number of Ratings of iPhones



There is a linear relationship between the discount percentage on iPhones on Flipkart and the number of ratings. It means iPhones with high discounts are sold more in India.

There is a linear relationship between the discount percentage on iPhones on Flipkart and the number of ratings. It means iPhones with high discounts are sold more in India.

1. APPLE iPhone 8 Plus (Gold, 64 GB) was the most appreciated iPhone in India

2. iPhones with lower sale prices are sold more in India

3.iPhones with high discounts are sold more in India

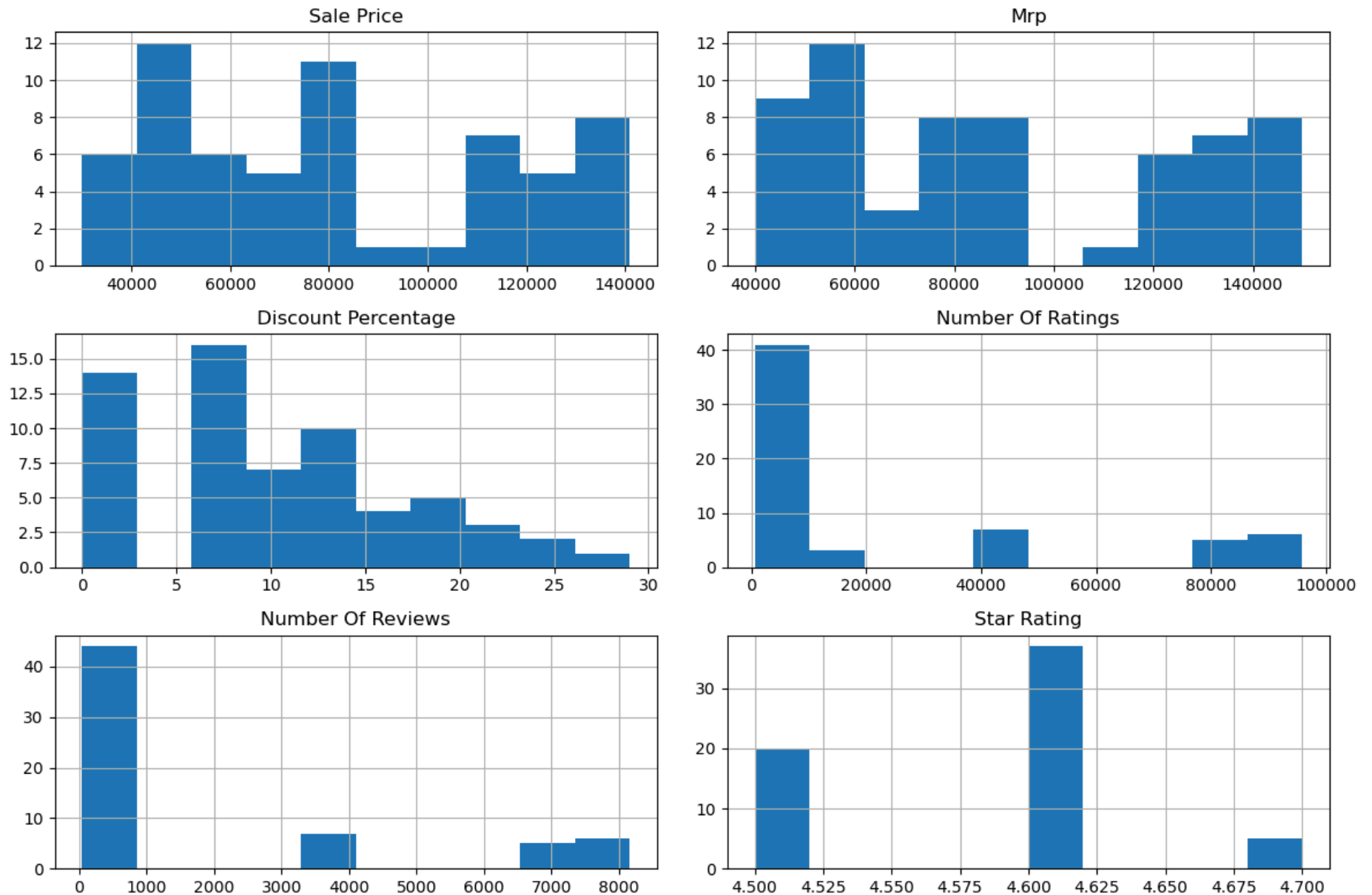
Summary Statistics:

```
In [17]: # Summary statistics
print(df.describe())
```

	Sale Price	Mrp	Discount Percentage	Number Of Ratings \
count	62.000000	62.000000	62.000000	62.000000
mean	80073.887097	88058.064516	9.951613	22420.403226
std	34310.446132	34728.825597	7.608079	33768.589550
min	29999.000000	39900.000000	0.000000	542.000000
25%	49900.000000	54900.000000	6.000000	740.000000
50%	75900.000000	79900.000000	10.000000	2101.000000
75%	117100.000000	120950.000000	14.000000	43470.000000
max	140900.000000	149900.000000	29.000000	95909.000000

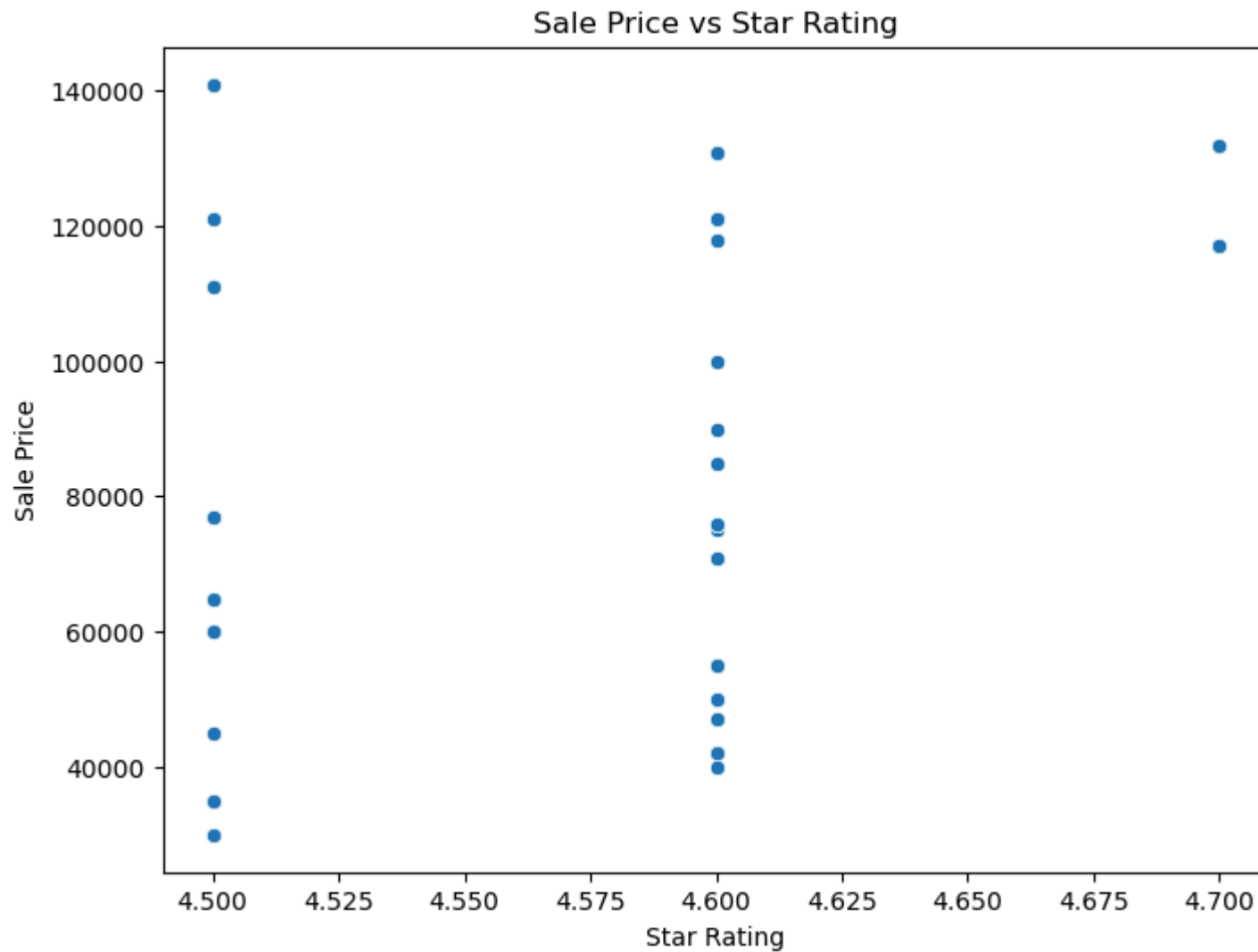
	Number Of Reviews	Star Rating
count	62.000000	62.000000
mean	1861.677419	4.575806
std	2855.883830	0.059190
min	42.000000	4.500000
25%	64.000000	4.500000
50%	180.000000	4.600000
75%	3331.000000	4.600000
max	8161.000000	4.700000

```
In [18]: # Histograms for numeric variables
numeric_cols = ['Sale Price', 'Mrp', 'Discount Percentage', 'Number Of Ratings', 'Number Of Reviews', 'Star Rating']
df[numeric_cols].hist(figsize=(12, 8))
plt.tight_layout()
plt.show()
```



```
In [21]: # Scatter plot for Sale Price vs Star Rating
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Star Rating', y='Sale Price')
plt.title('Sale Price vs Star Rating')
plt.xlabel('Star Rating')
```

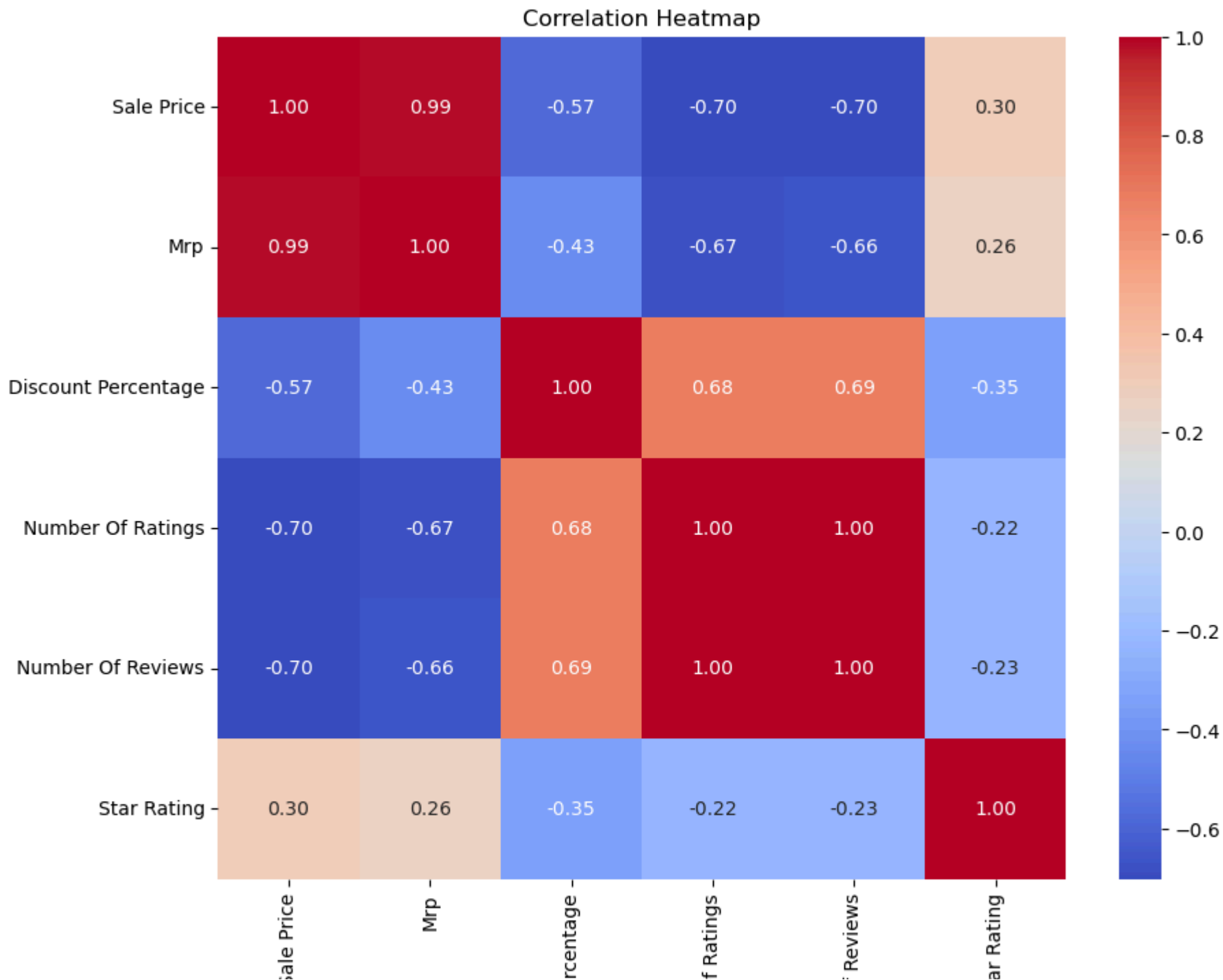
```
plt.ylabel('Sale Price')  
plt.show()
```



```
In [20]: # Correlation heatmap  
plt.figure(figsize=(10, 8))  
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")  
plt.title('Correlation Heatmap')  
plt.show()
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_11744\4137570652.py:3: FutureWarning:
```

```
The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
```



5

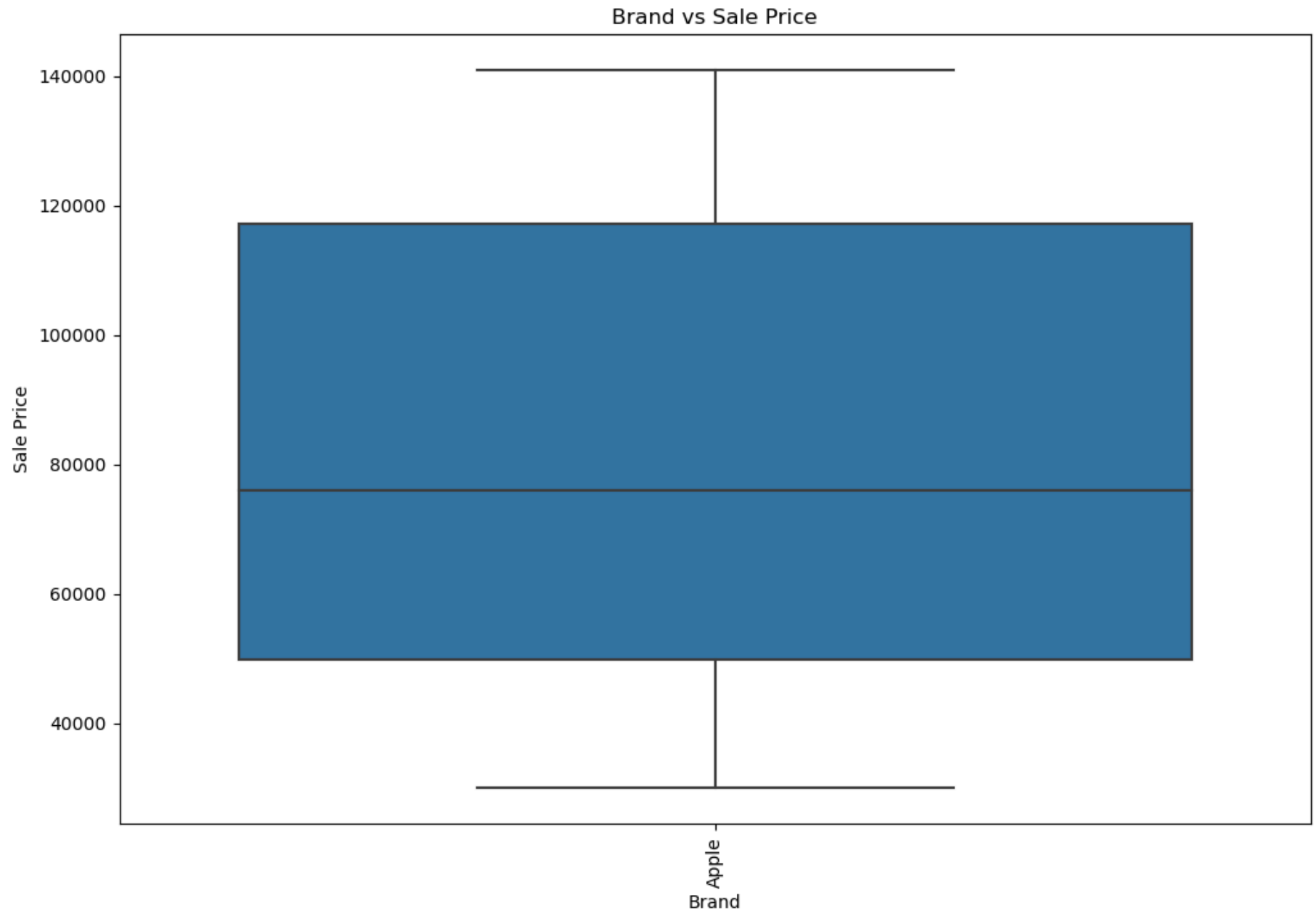
Discount Pe

Number O

Number Of

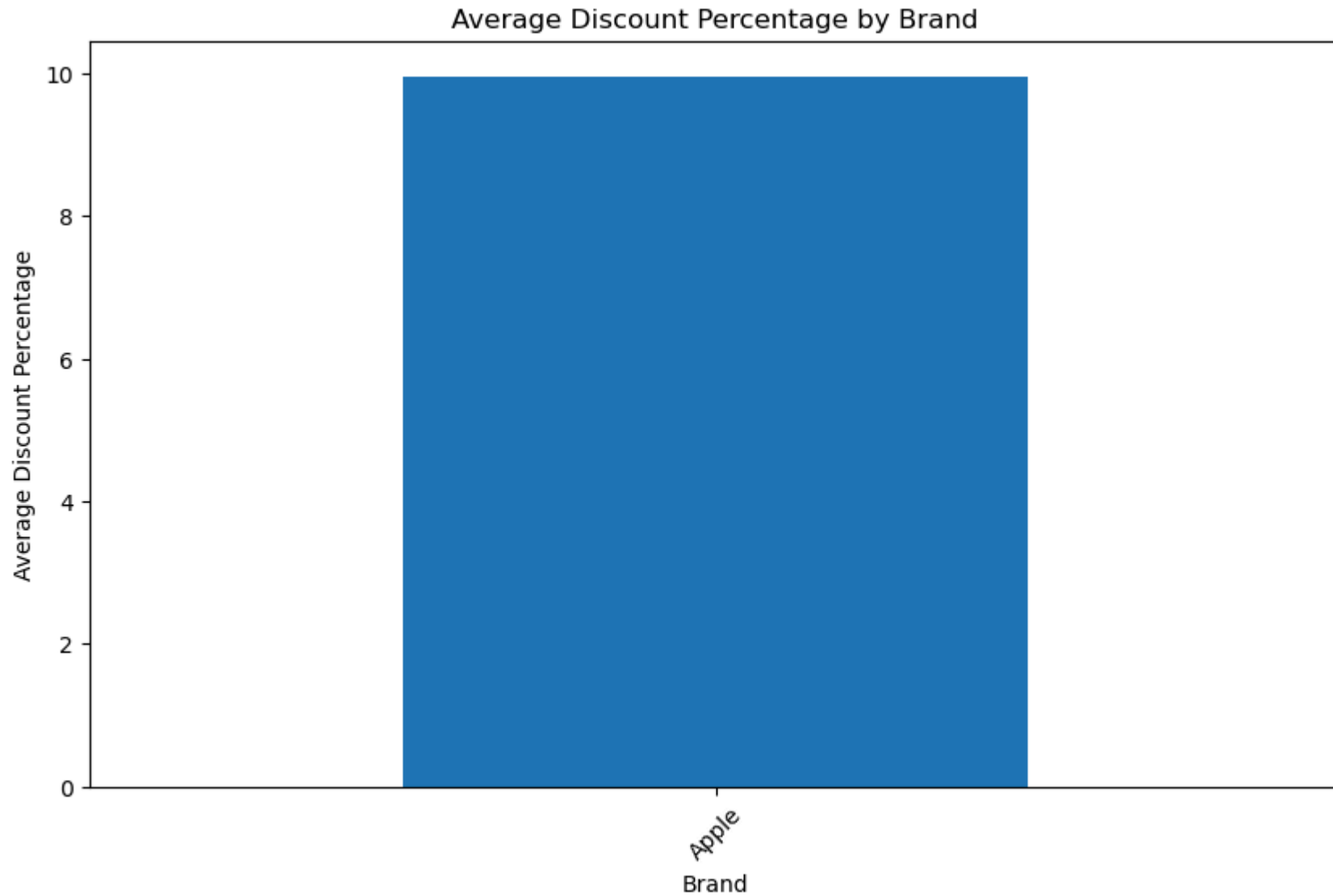
Sta

```
In [22]: # Boxplot for Brand vs Sale Price
plt.figure(figsize=(12, 8))
sns.boxplot(data=df, x='Brand', y='Sale Price')
plt.title('Brand vs Sale Price')
plt.xlabel('Brand')
plt.ylabel('Sale Price')
plt.xticks(rotation=90)
plt.show()
```

```
In [23]: # Which brands have the highest average discount percentages on their products?
avg_discount_per_brand = df.groupby('Brand')['Discount Percentage'].mean().sort_values(ascending=False)
```

```
plt.figure(figsize=(10, 6))
avg_discount_per_brand.plot(kind='bar')
plt.title('Average Discount Percentage by Brand')
plt.xlabel('Brand')
plt.ylabel('Average Discount Percentage')
plt.xticks(rotation=45)
plt.show()
```

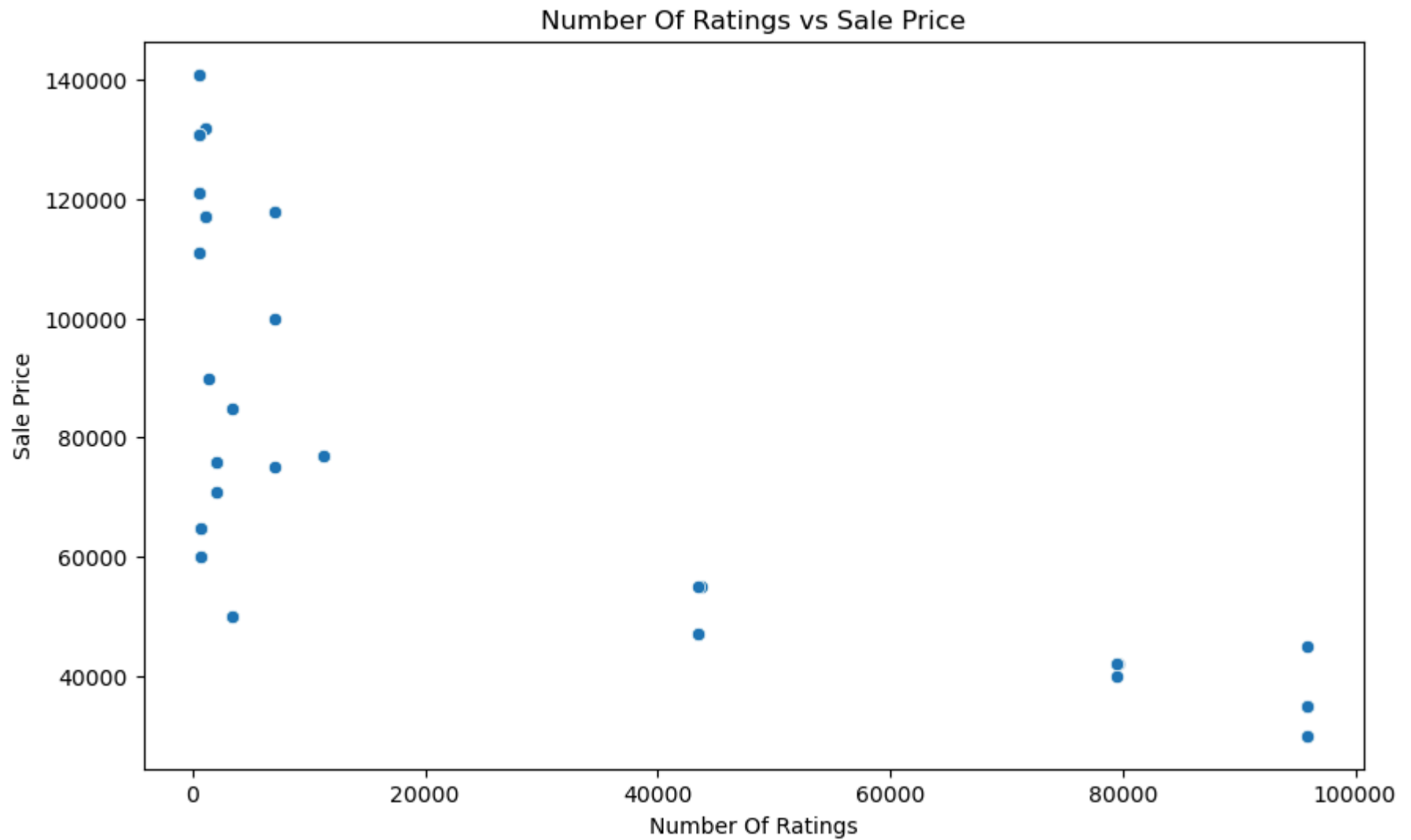


```
In [24]: # Is there a correlation between the number of ratings/reviews and the sales price of products?
correlation_ratings_price = df[['Number Of Ratings', 'Number Of Reviews', 'Sale Price']].corr()
print("Correlation between Number Of Ratings and Sale Price:", correlation_ratings_price.loc['Number Of Ratings', 'Sale Price'])
print("Correlation between Number Of Reviews and Sale Price:", correlation_ratings_price.loc['Number Of Reviews', 'Sale Price'])
```

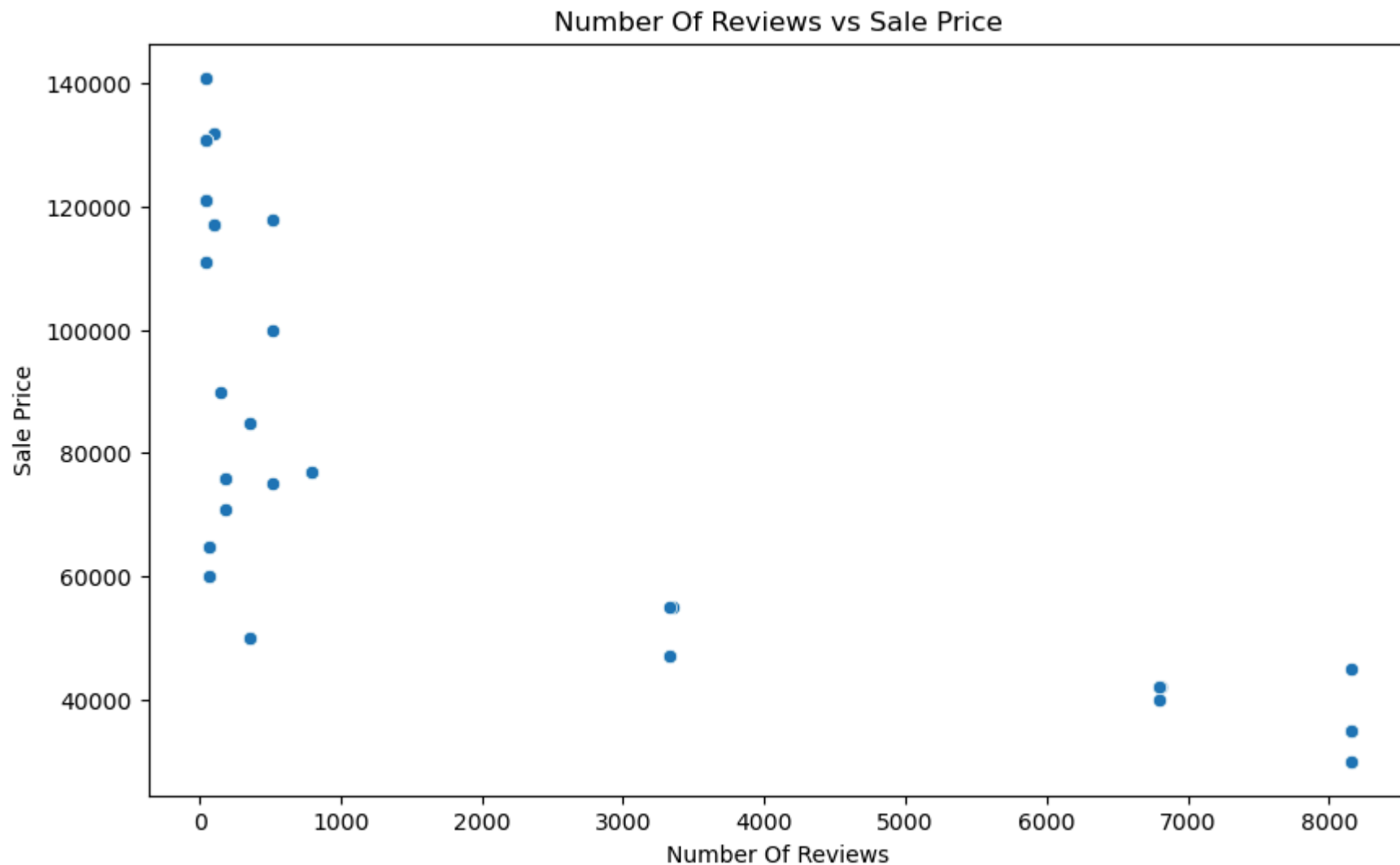
Correlation between Number Of Ratings and Sale Price: -0.7015259181182031

Correlation between Number Of Reviews and Sale Price: -0.6960291835220084

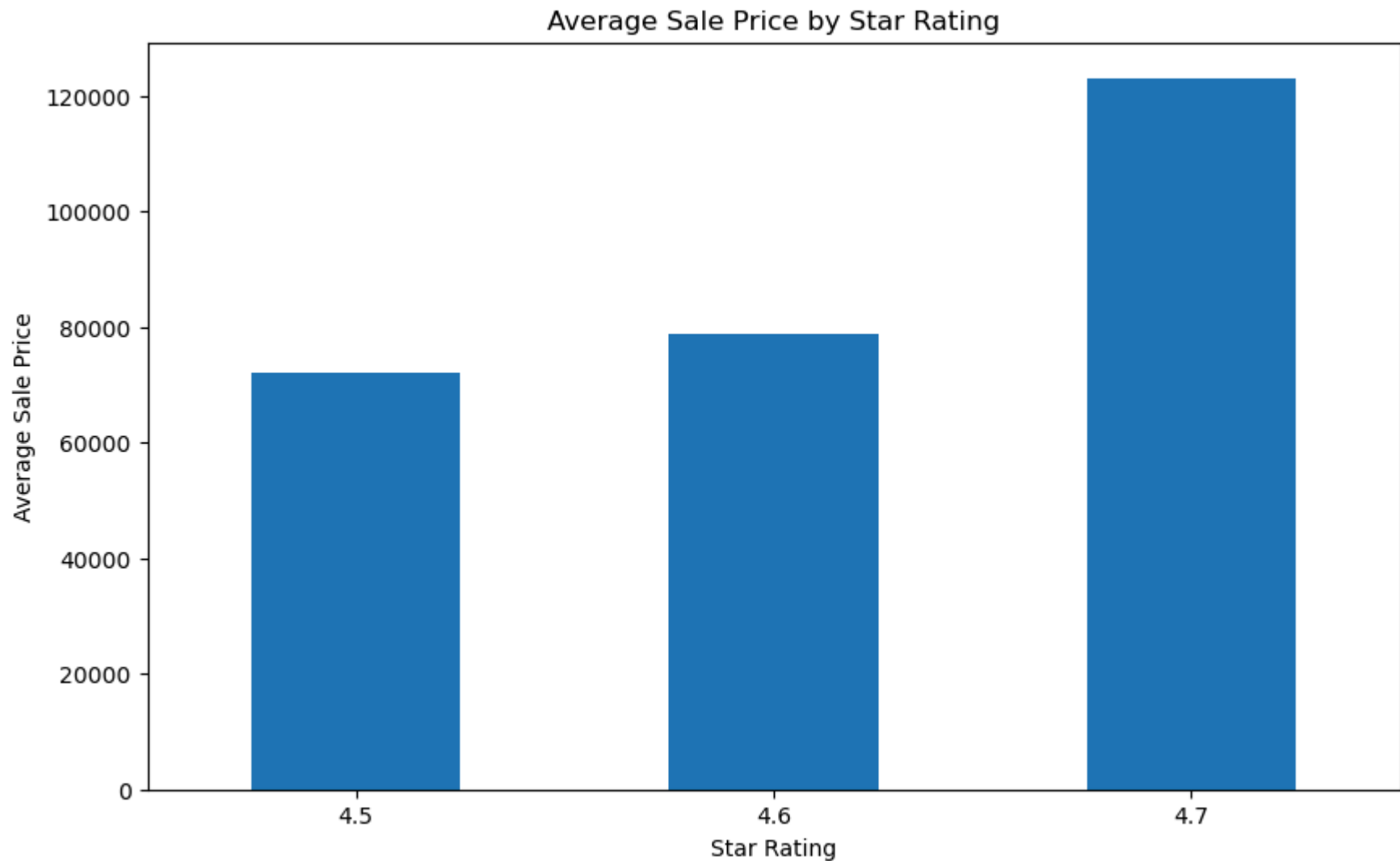
```
In [25]: # Visualize the relationship between ratings/reviews and sales price
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='Number Of Ratings', y='Sale Price')
plt.title('Number Of Ratings vs Sale Price')
plt.xlabel('Number Of Ratings')
plt.ylabel('Sale Price')
plt.show()
```



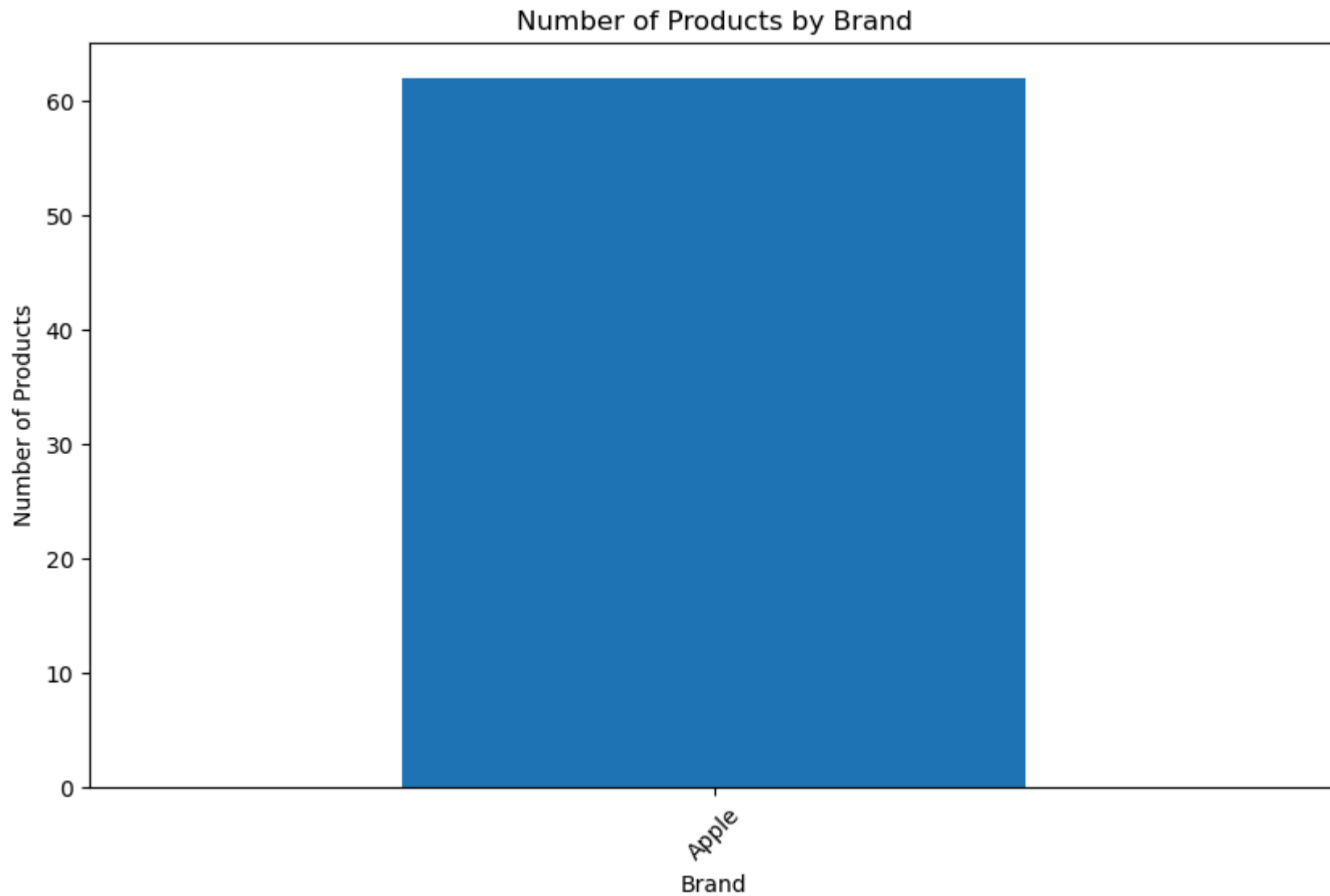
```
In [26]: plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='Number Of Reviews', y='Sale Price')
plt.title('Number Of Reviews vs Sale Price')
plt.xlabel('Number Of Reviews')
plt.ylabel('Sale Price')
plt.show()
```



```
In [27]: # Do products with higher star ratings tend to have higher sales prices?
avg_price_per_rating = df.groupby('Star Rating')['Sale Price'].mean()
plt.figure(figsize=(10, 6))
avg_price_per_rating.plot(kind='bar')
plt.title('Average Sale Price by Star Rating')
plt.xlabel('Star Rating')
plt.ylabel('Average Sale Price')
plt.xticks(rotation=0)
plt.show()
```



```
In [28]: # 4. Are there any brands that dominate the market in terms of the number of products offered?
products_per_brand = df['Brand'].value_counts()
plt.figure(figsize=(10, 6))
products_per_brand.plot(kind='bar')
plt.title('Number of Products by Brand')
plt.xlabel('Brand')
plt.ylabel('Number of Products')
plt.xticks(rotation=45)
plt.show()
```



```
In [29]: # 5. Is there a relationship between the sale price and the discount percentage of products?
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='Sale Price', y='Discount Percentage')
plt.title('Sale Price vs Discount Percentage')
plt.xlabel('Sale Price')
plt.ylabel('Discount Percentage')
plt.show()
```

```
correlation_price_discount = df[['Sale Price', 'Discount Percentage']].corr().loc['Sale Price', 'Discount Percentage']  
print("Correlation between Sale Price and Discount Percentage:", correlation_price_discount)
```



Correlation between Sale Price and Discount Percentage: -0.5696033014974266



