Descriptive Statistics

Descriptive statistics are a collection of quantitative measures that summarize and describe the main characteristics of a dataset.

```
df1.describe()
```

	discounted_price	actual_price	discount_percentage	rating	rating_count
count	1465.000000	1465.000000	1465.000000	1465.000000	1463.000000
mean	3125.310874	5444.990635	47.691468	4.096587	18295.541353
std	6944.304394	10874.826864	21.635905	0.291574	42753.864952
min	39.000000	39.000000	0.000000	2.000000	2.000000
25%	325.000000	800.000000	32.000000	4.000000	1186.000000
50%	799.000000	1650.000000	50.000000	4.100000	5179.000000
75%	1999.000000	4295.000000	63.000000	4.300000	17336.500000
max	77990.000000	139900.000000	94.000000	5.000000	426973.000000

Most products are moderately priced with a median discounted price of ₹799.

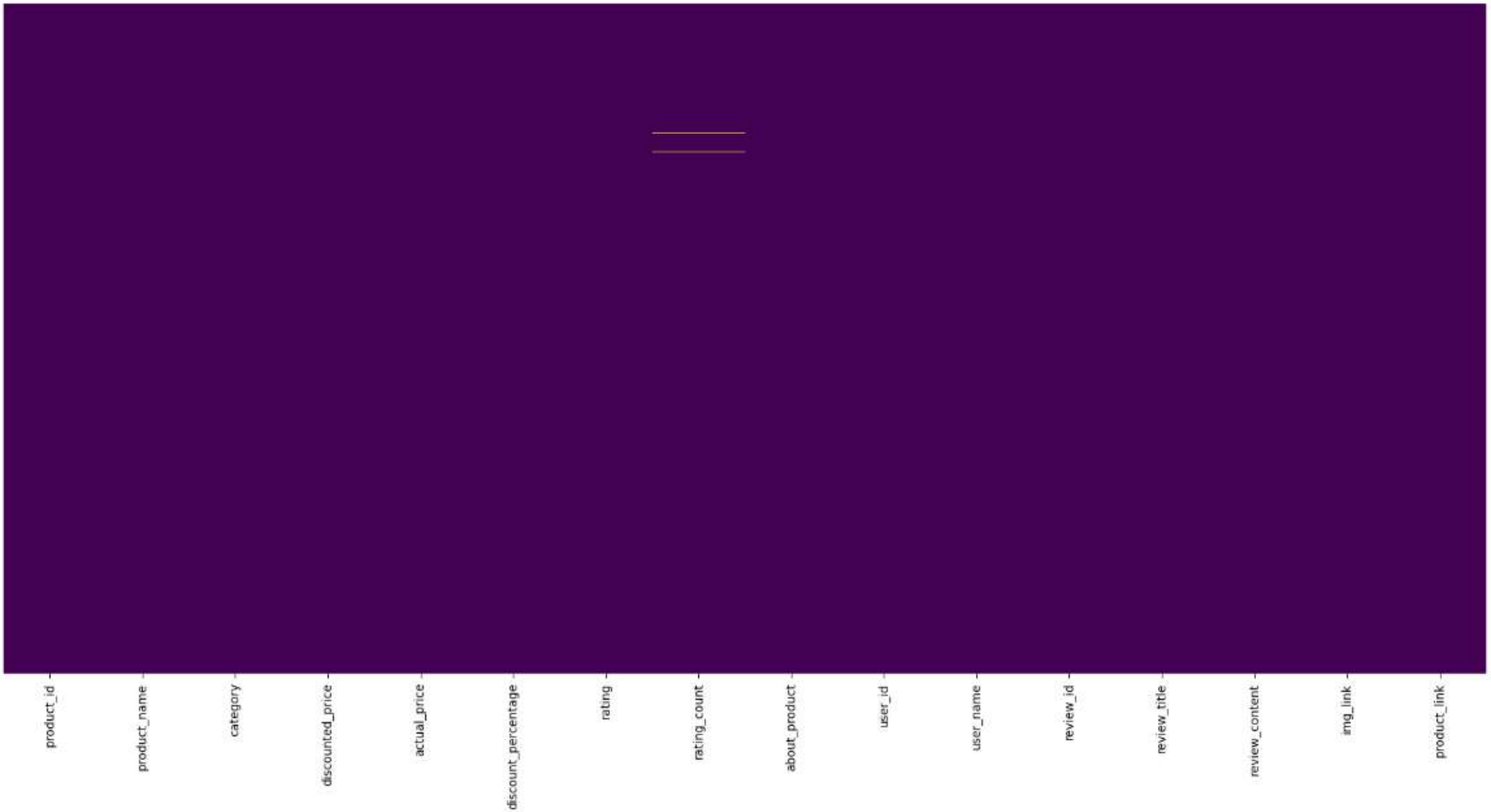
Median discount is 50%, typically ranging from 32% to 63%.

Ratings are generally high, around 4.1 out of 5.

Review counts vary widely, some products are extremely popular.

A few products are outliers in price, discount, and popularity.

```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(22,10))
sns.heatmap(df1.isnull(), cbar=False, yticklabels=False, cmap='viridis')
plt.show()
```

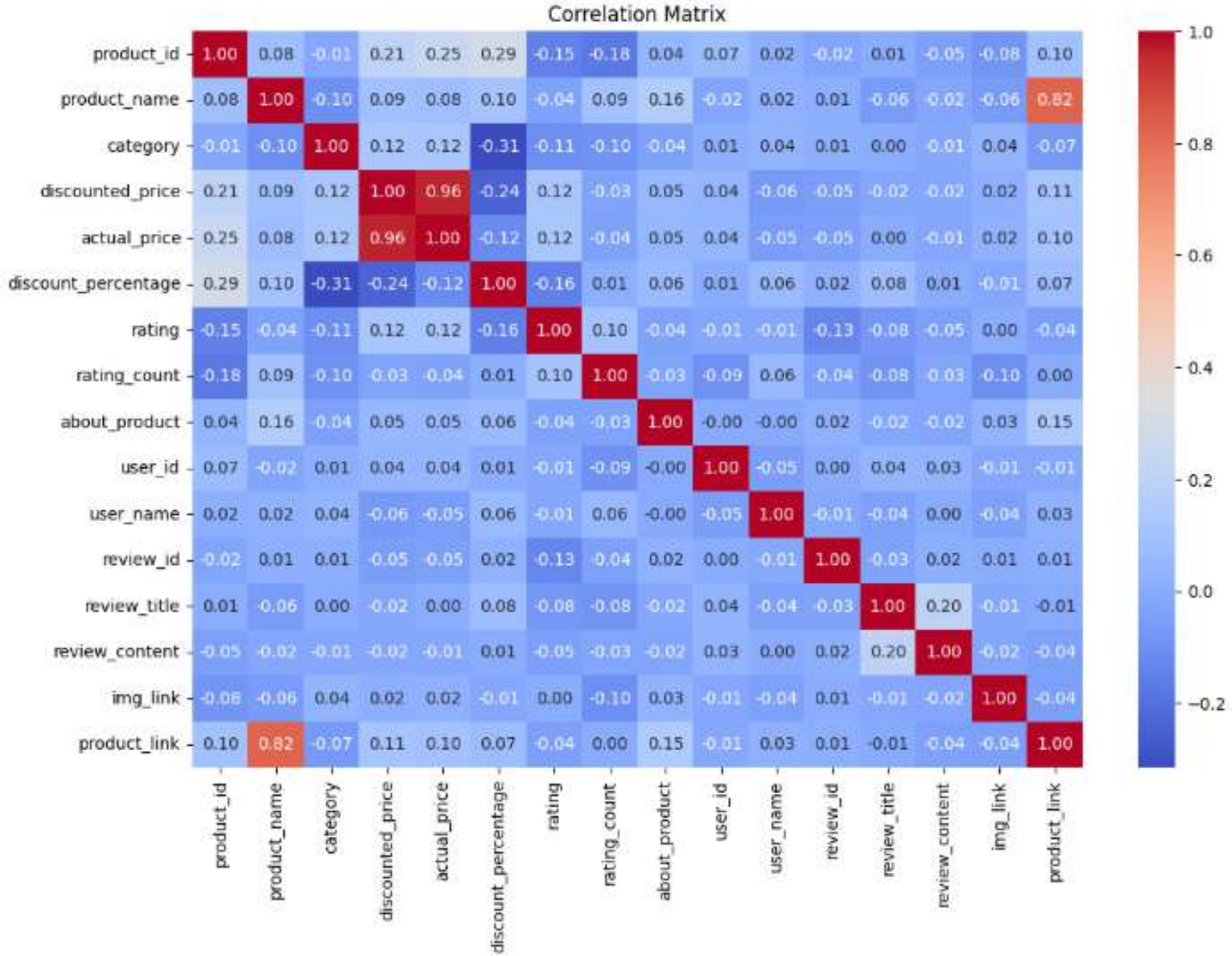


Create a heatmap to visualize the correlations

```
import matplotlib.pyplot as plt

plt.figure(figsize=(12,8))
sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap="coolwarm")
plt.title("Correlation Matrix")
plt.show()
```

...

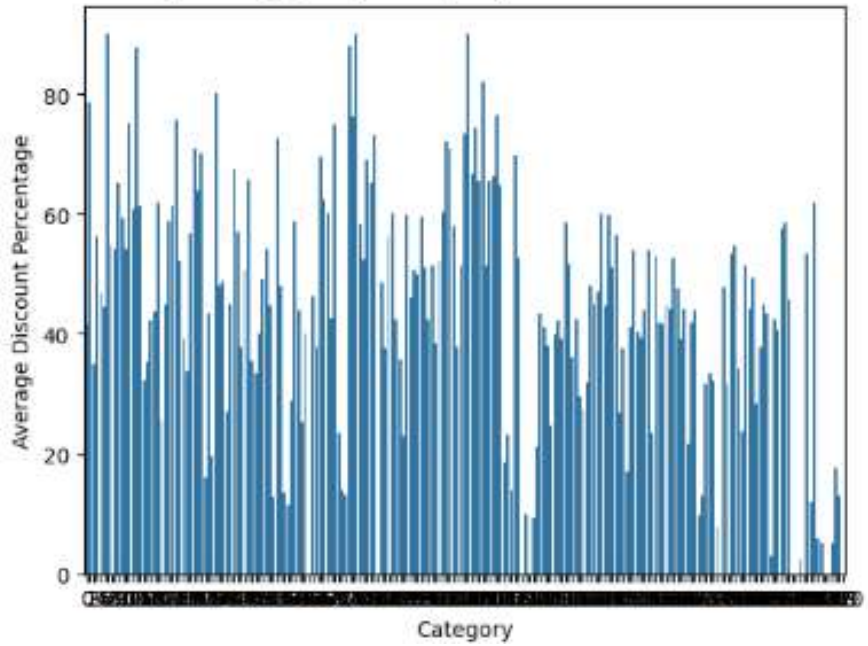


Q4: How does the average discount percentage vary across categories?

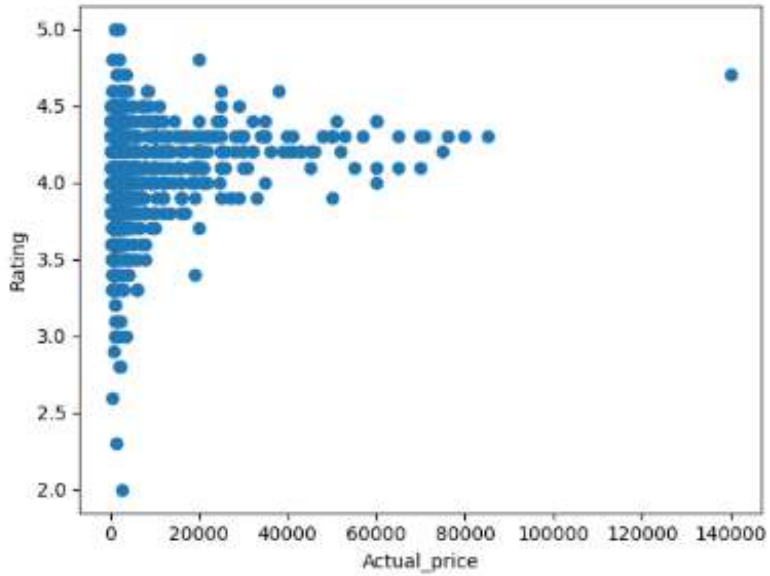
Average discount percentages vary widely across categories, ranging from 0% to 78.39%. Categories 1 and 3 stand out with notably higher average discounts (78.39% and 56.34%), suggesting potential factors like clearance efforts, high competition, or lower-profit margins. Categories 0, 206, 207, 210 have average discounts of 0%, indicating consistent pricing or strong demand for products within those categories. Other categories exhibit varying discount percentages, likely reflecting diverse pricing strategies and market dynamics.

```
avg_discount_per_category = df1.groupby('category')['discount_percentage'].mean()
print(avg_discount_per_category)
sns.barplot(x=avg_discount_per_category.index, y=avg_discount_per_category.values)
plt.xlabel("Category")
plt.ylabel("Average Discount Percentage")
plt.show()
```

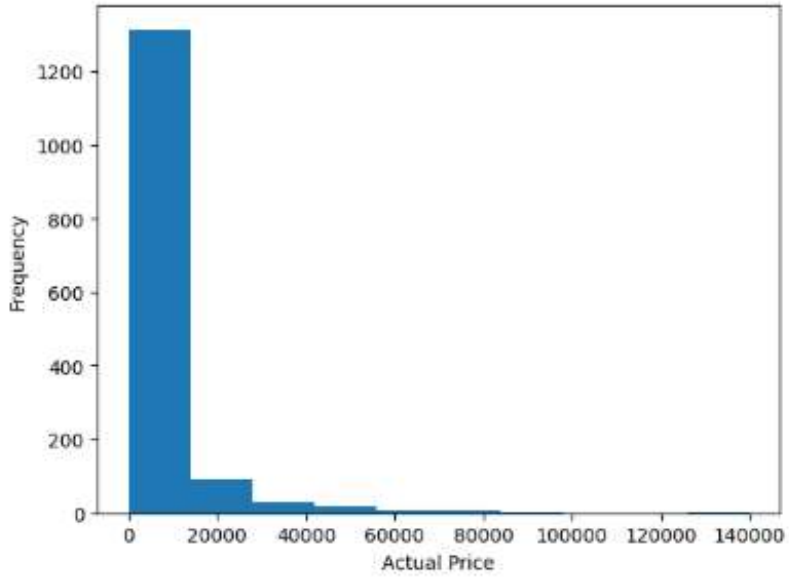
```
category
0      41.525000
1      78.387733
2      35.035035
3      56.335120
4      46.719582
         ...
206     0.000000
207     5.000000
208    17.619048
209    13.074074
210     0.000000
Name: discount_percentage, Length: 211, dtype: float64
```



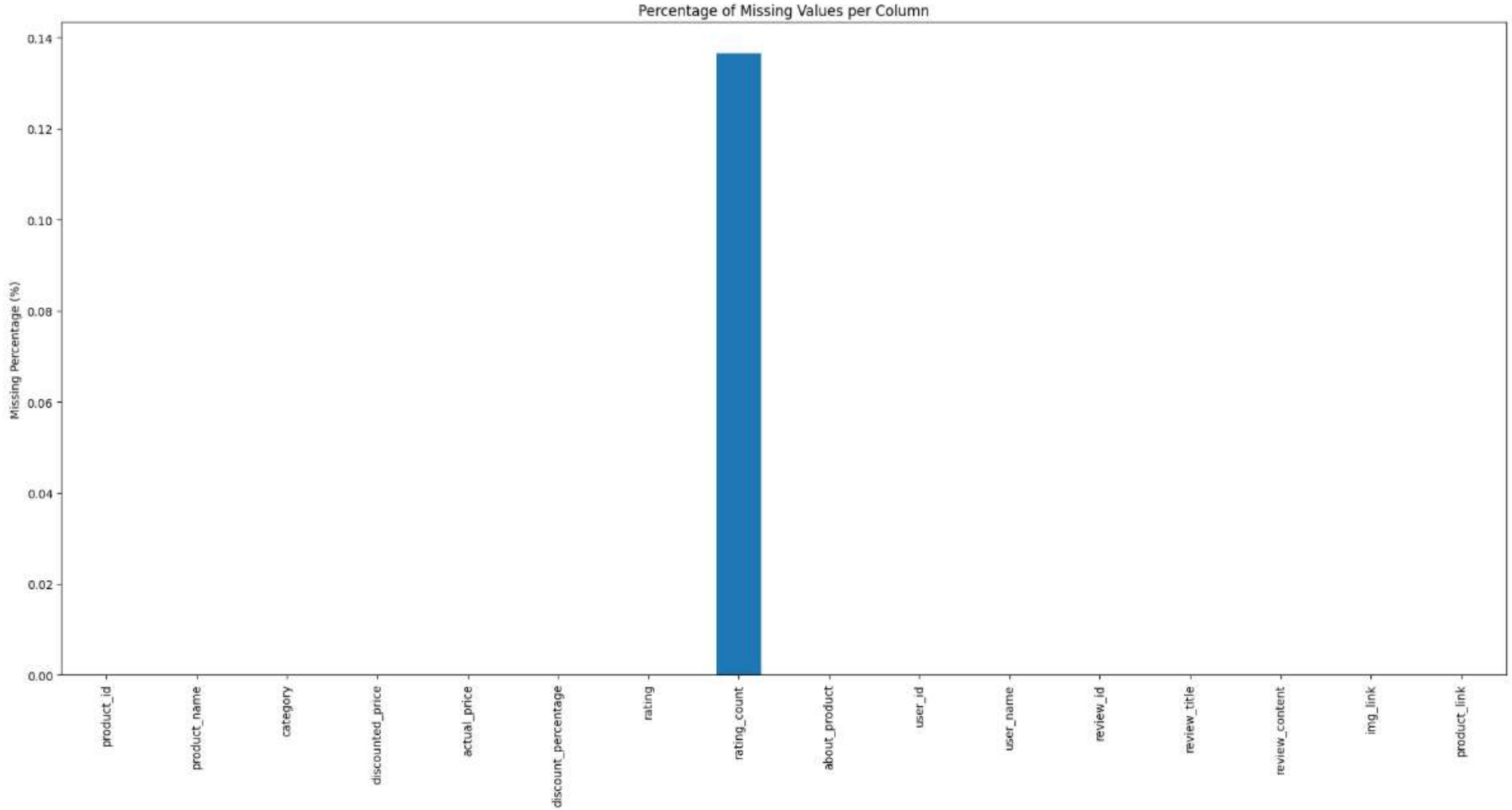
```
# Plot actual_price vs. rating
plt.scatter(df1['actual_price'], df1['rating'])
plt.xlabel('Actual_price')
plt.ylabel('Rating')
plt.show()
```




```
# Plot distribution of actual_price
plt.hist(df1['actual_price'])
plt.xlabel('Actual Price')
plt.ylabel('Frequency')
plt.show()
```



```
missing_percentage = (df.isnull().sum() / len(df1)) * 100
plt.figure(figsize=(22, 10))
missing_percentage.plot(kind='bar')
plt.title("Percentage of Missing Values per Column")
plt.xlabel("Columns")
plt.ylabel("Missing Percentage (%)")
plt.show()
```



```python
# Create histograms
df1["discounted_price"].hist(label="Discounted Price")
df1["actual_price"].hist(label="Actual Price")

# Calculate and analyze discount percentages
df1["discount_percentage"] = (df1["actual_price"] - df1["discounted_price"]) / df1["actual_price"] * 100
df1["discount_percentage"].describe()
df1["discount_percentage"].hist(label="Discount Percentage")
plt.show()
```