```
#Yashvi Nagda
# Big Data Workflows in AI-Powered Business Analytics - DAT-1001 - VNA1
# 13th July 2025
# Data visualization File
```

Category Performance Overview

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


# Load the uploaded CSV file
file_path = 'category_Performance.csv'
df = pd.read_csv(file_path)


# Display the first few rows to understand the data structure
df.head()
```

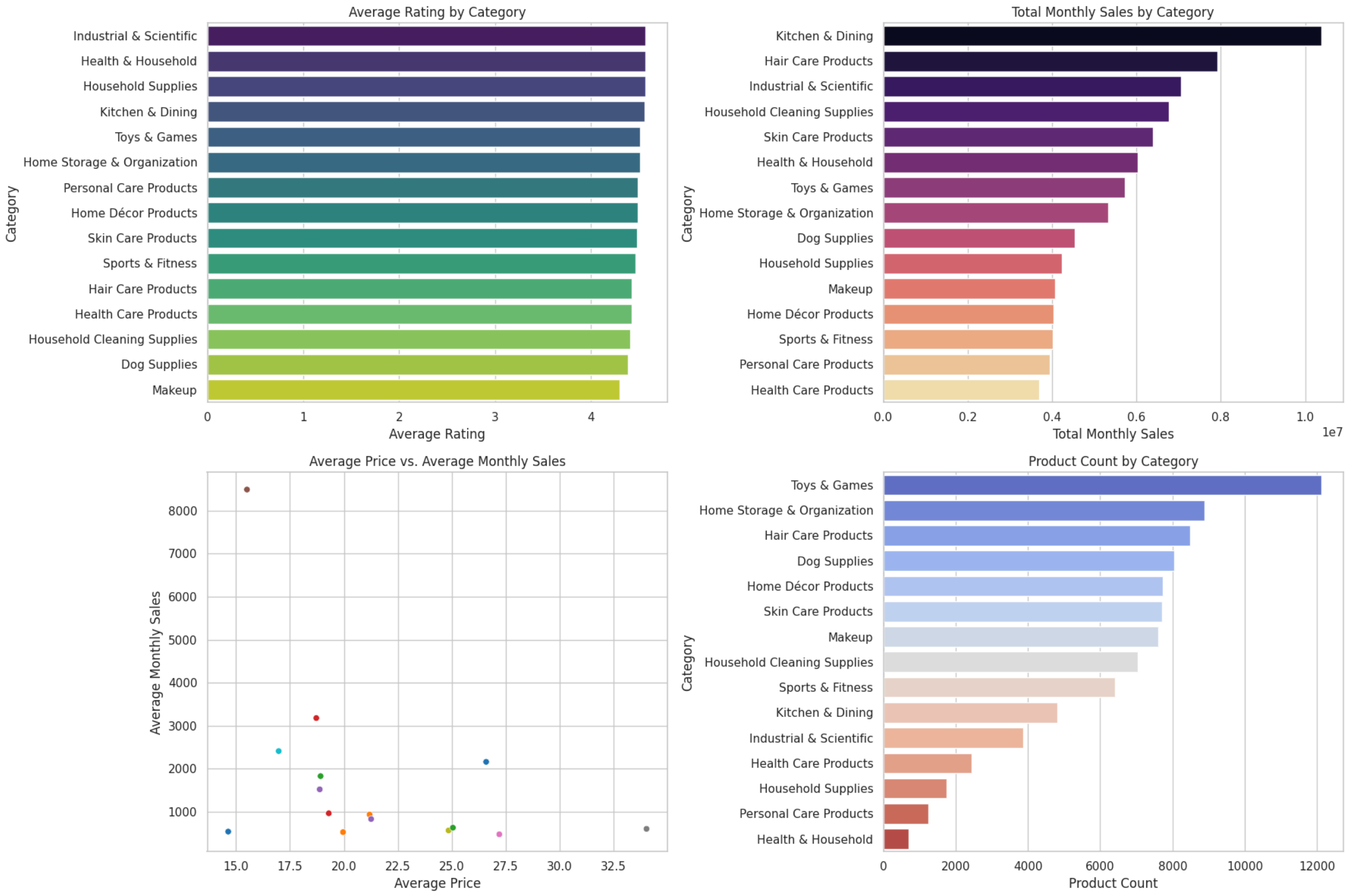
	category_name	product_count	avg_rating	avg_monthly_sales	total_monthly_sales	avg_price
0	Kitchen & Dining	4812	4.56	2158.7	10387600.0	26.59
1	Hair Care Products	8494	4.43	931.9	7915350.0	21.19
2	Industrial & Scientific	3864	4.57	1826.4	7057250.0	18.92
3	Household Cleaning Supplies	7049	4.41	961.6	6778050.0	19.30
4	Skin Care Products	7717	4.48	828.2	6391300.0	21.26

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

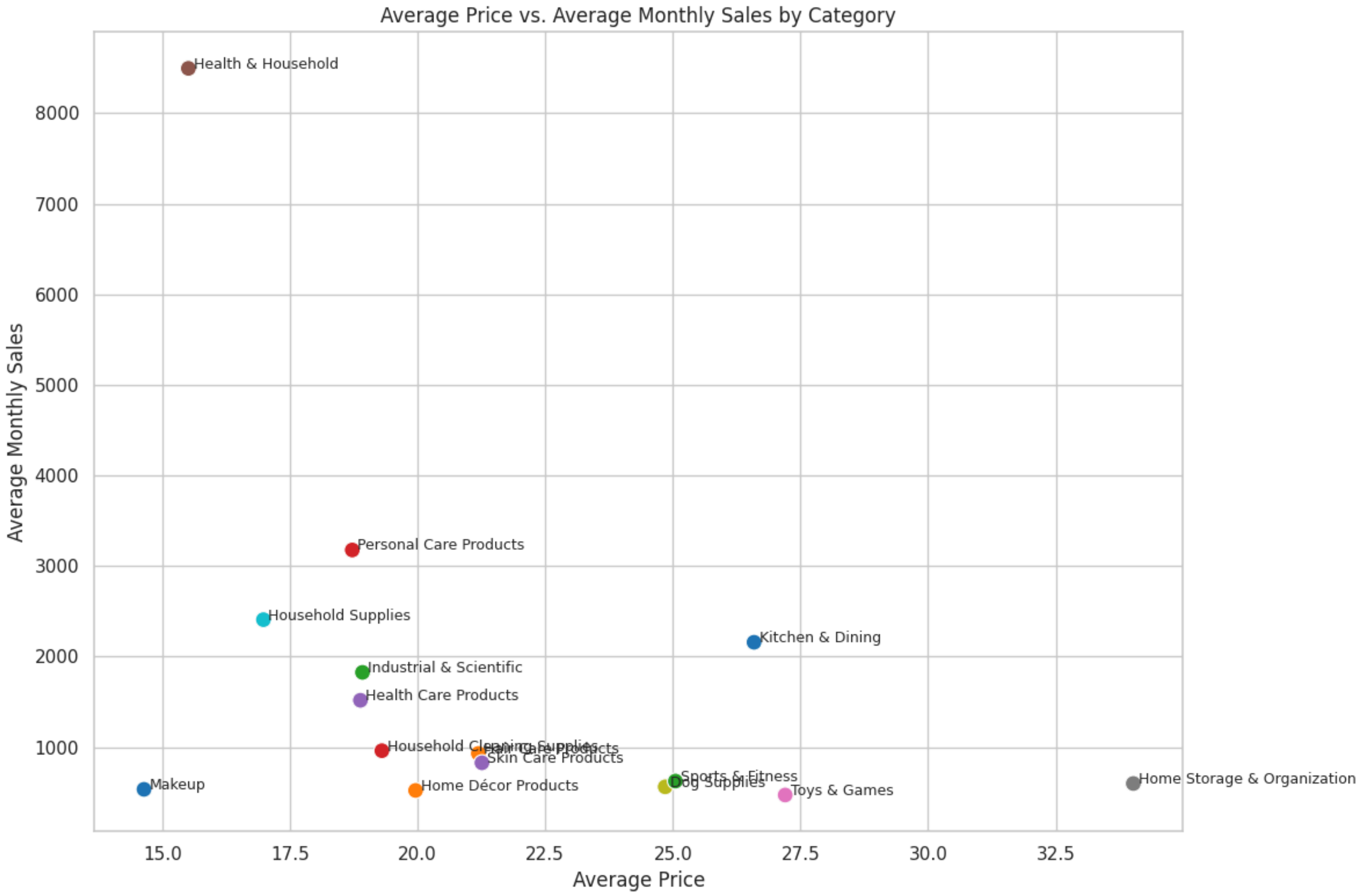
```
# Set style for plots
sns.set(style="whitegrid")

# Create multiple graphs to visualize key aspects of the dataset
fig, axs = plt.subplots(2, 2, figsize=(18, 12))

# 1. Bar chart: Average Rating by Category
sns.barplot(data=df.sort_values("avg_rating", ascending=False),
            x="avg_rating", y="category_name", ax=axs[0, 0], palette="viridis")
axs[0, 0].set_title("Average Rating by Category")
axs[0, 0].set_xlabel("Average Rating")
axs[0, 0].set_ylabel("Category")

# 2. Bar chart: Total Monthly Sales by Category
sns.barplot(data=df.sort_values("total_monthly_sales", ascending=False),
            x="total_monthly_sales", y="category_name", ax=axs[0, 1], palette="magma")
axs[0, 1].set_title("Total Monthly Sales by Category")
axs[0, 1].set_xlabel("Total Monthly Sales")
axs[0, 1].set_ylabel("Category")

# 3. Scatter Plot: Average Price vs. Average Monthly Sales
sns.scatterplot(data=df, x="avg_price", y="avg_monthly_sales", hue="category_name", ax=axs[1, 0], palette="tab10", legend=False)
axs[1, 0].set_title("Average Price vs. Average Monthly Sales")
axs[1, 0].set_xlabel("Average Price")
axs[1, 0].set_ylabel("Average Monthly Sales")

# 4. Bar chart: Product Count by Category
sns.barplot(data=df.sort_values("product_count", ascending=False),
            x="product_count", y="category_name", ax=axs[1, 1], palette="coolwarm")
axs[1, 1].set_title("Product Count by Category")
axs[1, 1].set_xlabel("Product Count")
axs[1, 1].set_ylabel("Category")

plt.tight_layout()
plt.show()
```

What can I help you build?

```
/tmp/ipython-input-9-2522940944.py:8: FutureWarning:

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

  sns.barplot(data=df.sort_values("avg_rating", ascending=False),
/tmp/ipython-input-9-2522940944.py:15: FutureWarning:

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

  sns.barplot(data=df.sort_values("total_monthly_sales", ascending=False),
/tmp/ipython-input-9-2522940944.py:28: FutureWarning:

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

  sns.barplot(data=df.sort_values("product_count", ascending=False),
```

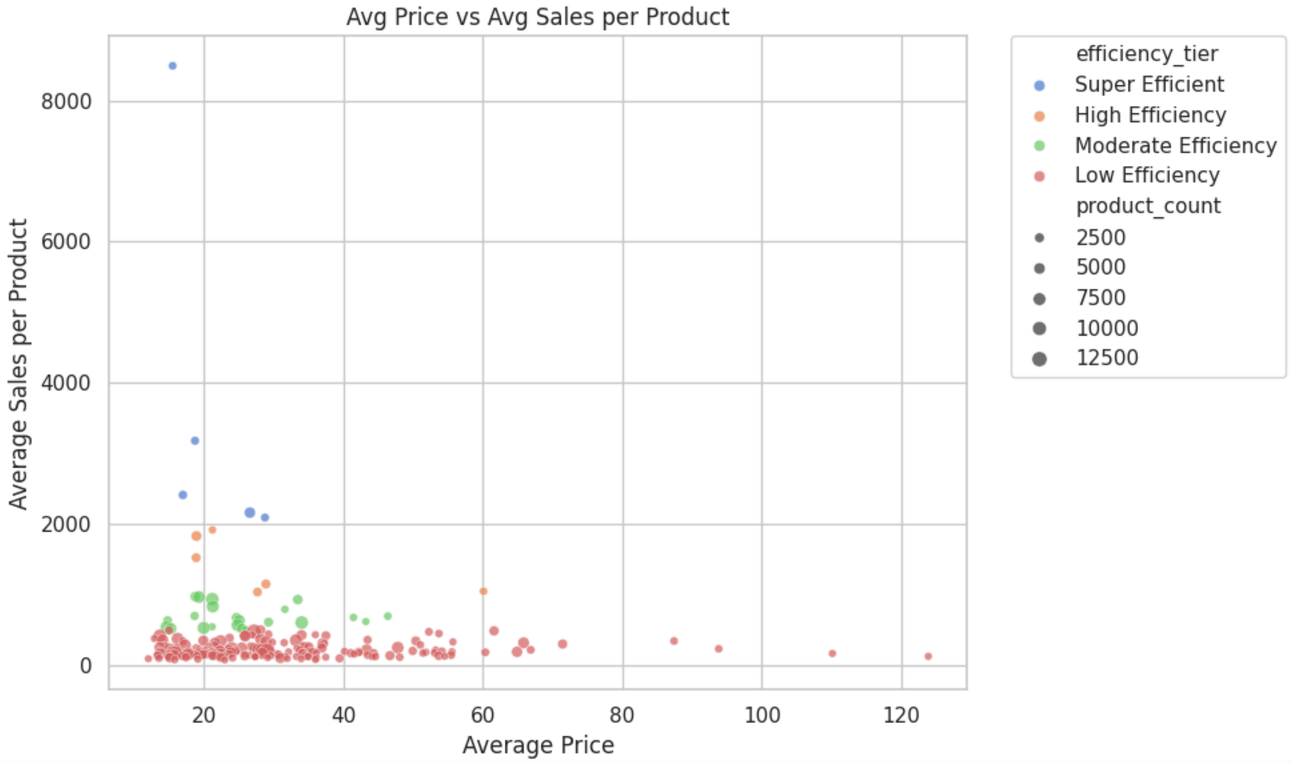


```
# Create a clearer scatter plot with labels for each category point
plt.figure(figsize=(12, 8))
scatter = sns.scatterplot(
    data=df,
    x="avg_price",
    y="avg_monthly_sales",
    hue="category_name",
    palette="tab10",
    s=100,
    legend=False
)

# Add category name labels directly to each point
for i in range(df.shape[0]):
    plt.text(
        x=df["avg_price"][i] + 0.1,  # slight offset to prevent overlap
        y=df["avg_monthly_sales"][i],
        s=df["category_name"][i],
        fontsize=9
    )

plt.title("Average Price vs. Average Monthly Sales by Category")
plt.xlabel("Average Price")
plt.ylabel("Average Monthly Sales")
plt.grid(True)
plt.tight_layout()
plt.show()
```



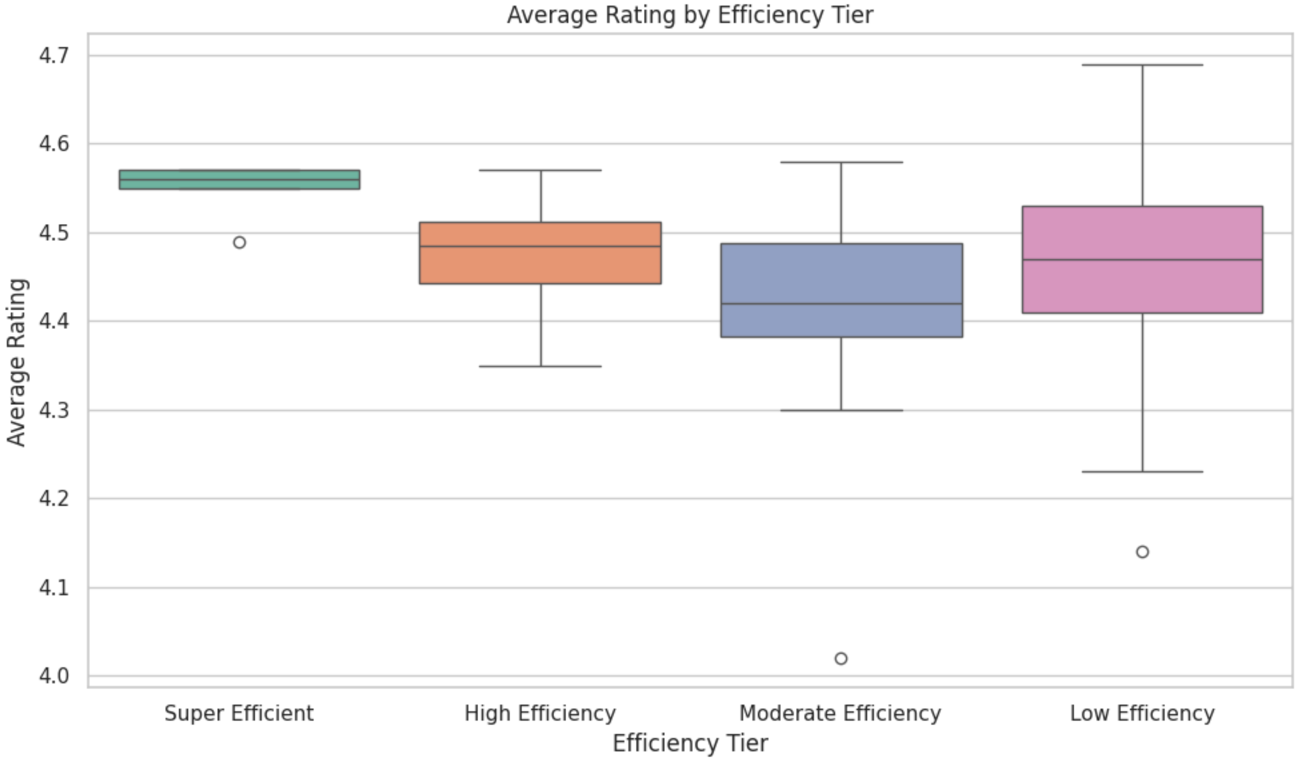
Market Efficiency

```
# Load the uploaded CSV file
file_path = 'Market_Efficiency.csv'
df = pd.read_csv(file_path)


# Display the first few rows to understand the data structure
df.head()


# Set visual style
sns.set(style="whitegrid")

# 1. Scatter plot: Avg Price vs Avg Sales per Product (size = product count)
plt.figure(figsize=(10, 6))
sns.scatterplot(
    data=df,
    x='avg_price',
    y='avg_sales_per_product',
    size='product_count',
    hue='efficiency_tier',
    alpha=0.7,
    palette='muted'
)
plt.title('Avg Price vs Avg Sales per Product')
plt.xlabel('Average Price')
plt.ylabel('Average Sales per Product')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.tight_layout()
plt.show()
```

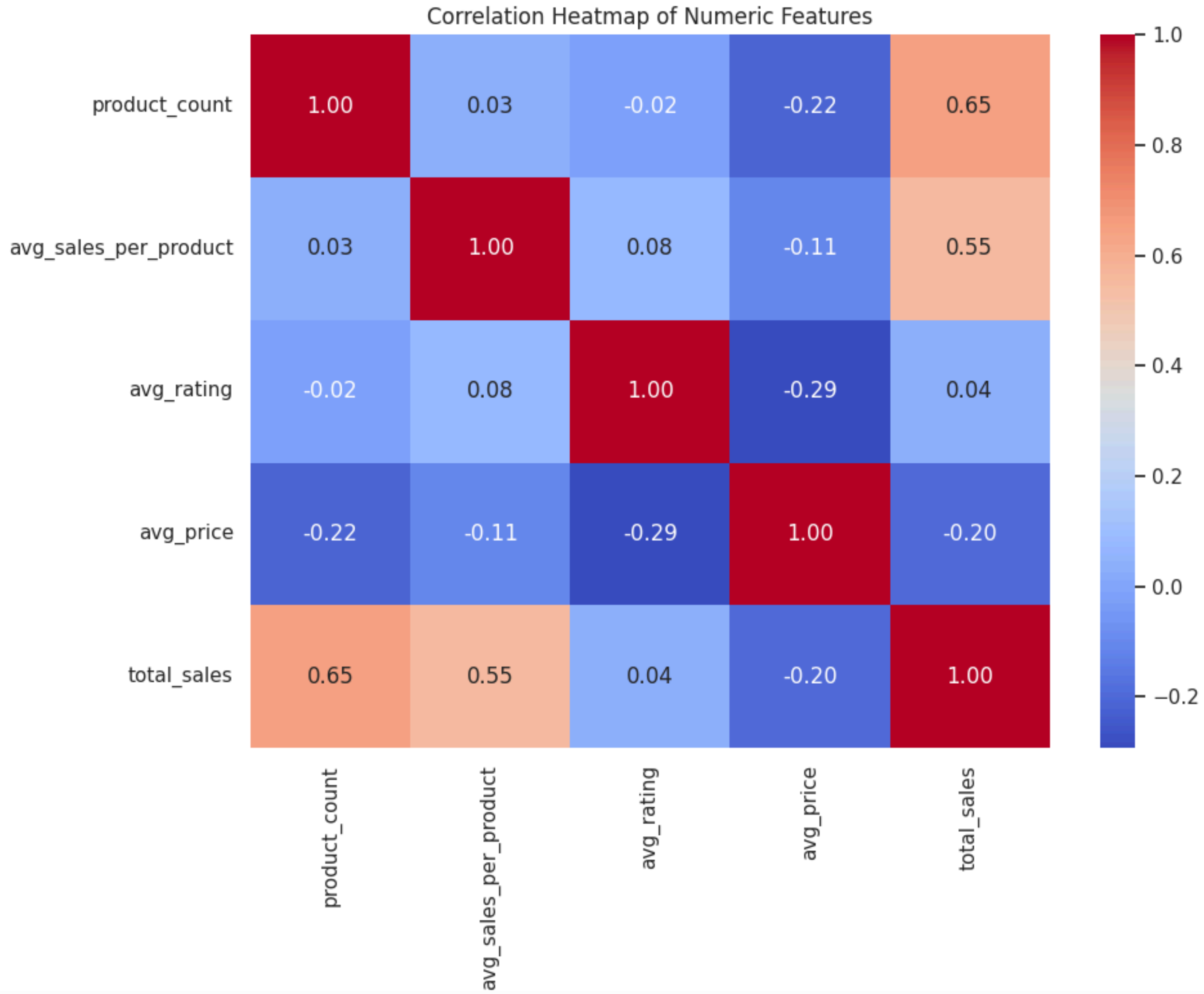


```
# 2. Boxplot: Distribution of Avg Rating by Efficiency Tier
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='efficiency_tier', y='avg_rating', palette='Set2')
plt.title('Average Rating by Efficiency Tier')
plt.xlabel('Efficiency Tier')
plt.ylabel('Average Rating')
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-10-2415250810.py:3: FutureWarning:

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

  sns.boxplot(data=df, x='efficiency_tier', y='avg_rating', palette='Set2')
```



```
# 3. Heatmap: Correlation between numeric features
plt.figure(figsize=(10, 8))
numeric_cols = ['product_count', 'avg_sales_per_product', 'avg_rating', 'avg_price', 'total_sales']
corr = df[numeric_cols].corr()
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap of Numeric Features')
plt.tight_layout()
plt.show()
```



Health and Household

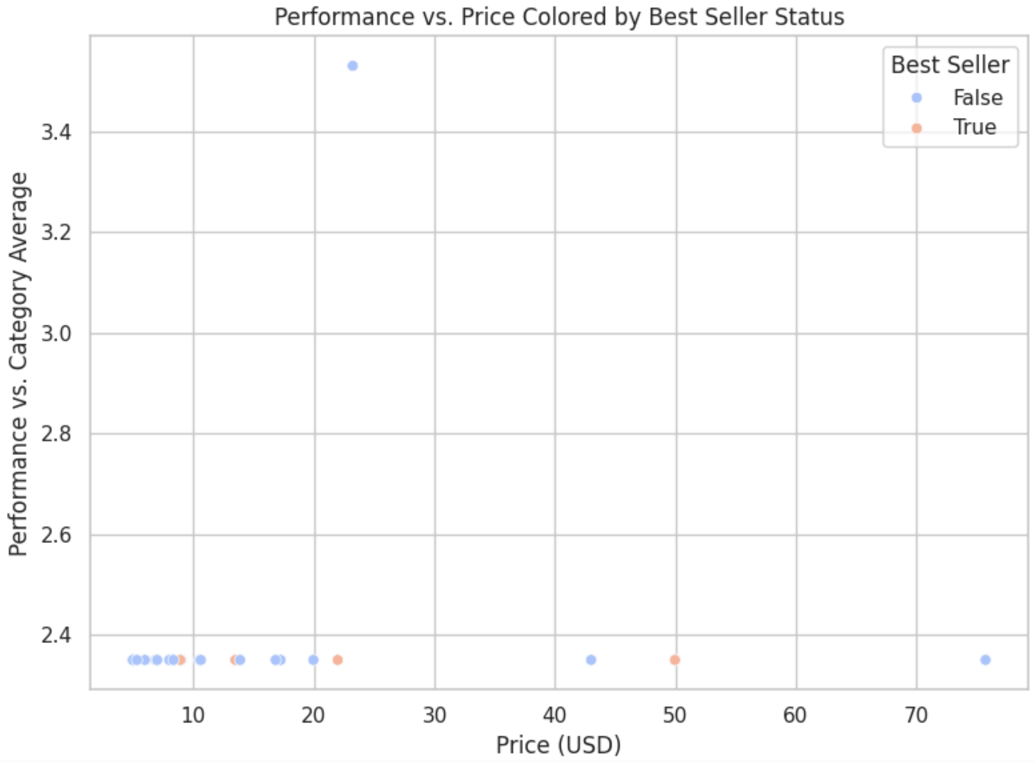
```
# Load the uploaded CSV file
file_path = 'Health & Household products.csv'
df = pd.read_csv(file_path)


# Display the first few rows to understand the data structure
df.head()
```

	product_id	title	stars	price	reviews	boughtInLastMonth	isBestSeller	performance_vs_category_avg
0	B0BVBYQGTW	Dove Body Wash with Pump Sensitive Skin 3 Coun...	4.8	23.22	0	30000	False	3.53
1	B01HTJTPZA	Dove Advanced Care Antiperspirant Cool Essenti...	4.8	13.48	0	20000	True	2.35
2	B089WRB791	Amazon Basics Original Fresh Liquid Hand Soap,...	4.5	6.85	0	20000	False	2.35
3	B081FFRGZB	Softsoap Antibacterial Liquid Hand Soap Refill...	4.5	5.97	0	20000	False	2.35
4	B002JDUMFO	L'Oreal Paris Collagen Daily Face Moisturizer,...	4.5	8.98	0	20000	False	2.35

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

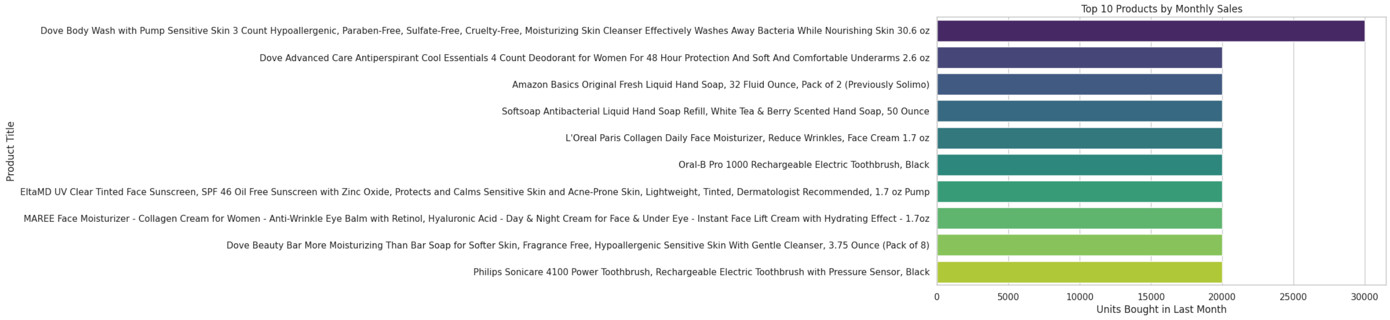
```
# Set seaborn style
sns.set(style="whitegrid")

# 1. Scatter Plot: Performance vs. Price
plt.figure(figsize=(8, 6))
```

```
sns.scatterplot(data=df, x='price', y='performance_vs_category_avg', hue='isBestSeller', palette='coolwarm')
plt.title('Performance vs. Price Colored by Best Seller Status')
plt.xlabel('Price (USD)')
plt.ylabel('Performance vs. Category Average')
plt.legend(title='Best Seller')
plt.tight_layout()
plt.show()
```



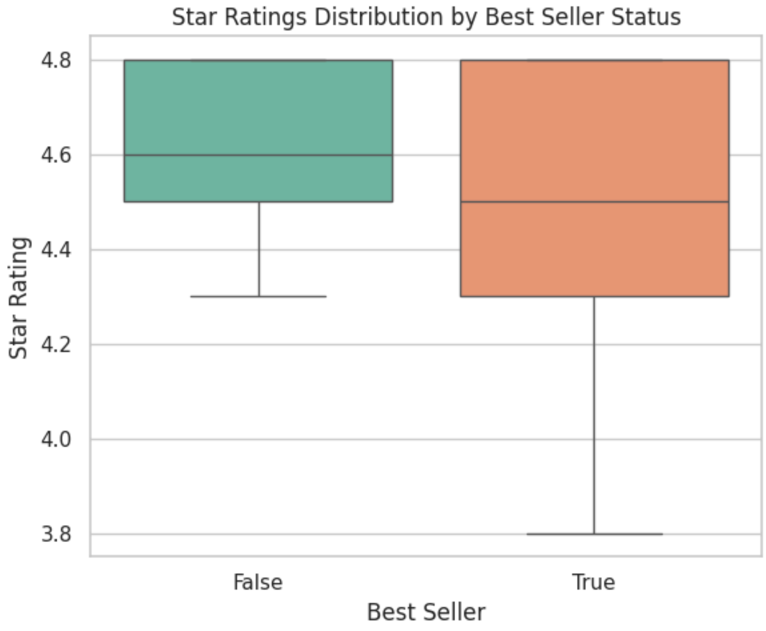
```
# 2. Bar Plot: Top Products by Monthly Sales
top_sales = df.sort_values(by='boughtInLastMonth', ascending=False).head(10)
plt.figure(figsize=(10, 6))
sns.barplot(data=top_sales, y='title', x='boughtInLastMonth', palette='viridis')
plt.title('Top 10 Products by Monthly Sales')
plt.xlabel('Units Bought in Last Month')
plt.ylabel('Product Title')
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-15-1393357869.py:4: FutureWarning:

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

  sns.barplot(data=top_sales, y='title', x='boughtInLastMonth', palette='viridis')
/tmp/ipython-input-15-1393357869.py:8: UserWarning: Tight layout not applied. The left and right margins cannot be made large enough to
  plt.tight_layout()
```



```
# 3. Box Plot: Star Ratings by Best Seller Status
plt.figure(figsize=(6, 5))
sns.boxplot(data=df, x='isBestSeller', y='stars', palette='Set2')
plt.title('Star Ratings Distribution by Best Seller Status')
plt.xlabel('Best Seller')
plt.ylabel('Star Rating')
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-16-422554573.py:3: FutureWarning:

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

  sns.boxplot(data=df, x='isBestSeller', y='stars', palette='Set2')
```



Personal Care Product

```
# Load the uploaded CSV file
file_path = 'Personal Care.csv'
df = pd.read_csv(file_path)
```

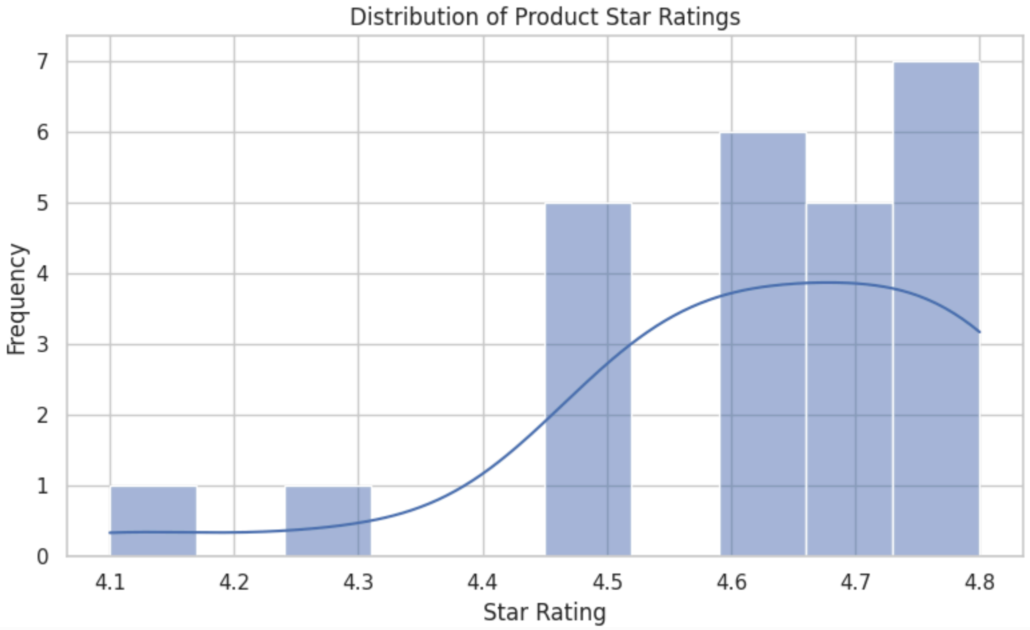
```
df.head()
```

	product_id	title	stars	price	reviews	boughtInLastMonth	isBestSeller	performance_vs_category_avg
0	B00PBX3L7K	COSRX Snail Mucin 96% Power Repairing Essence ...	4.6	15.00	0	100000	True	31.47
1	B00U2VQZDS	Neutrogena Cleansing Fragrance Free Makeup Rem...	4.8	10.27	0	100000	True	31.47
2	B074PVTPBW	Mighty Patch Original from Hero Cosmetics - Hy...	4.5	11.97	0	100000	True	31.47
3	B00TTD9BRC	CeraVe Moisturizing Cream | Body and Face Mois...	4.8	17.78	0	90000	True	28.32
4	B00MEDOY2G	Dove Body Wash with Pump Deep Moisture For Dry...	4.8	9.47	0	70000	True	22.03

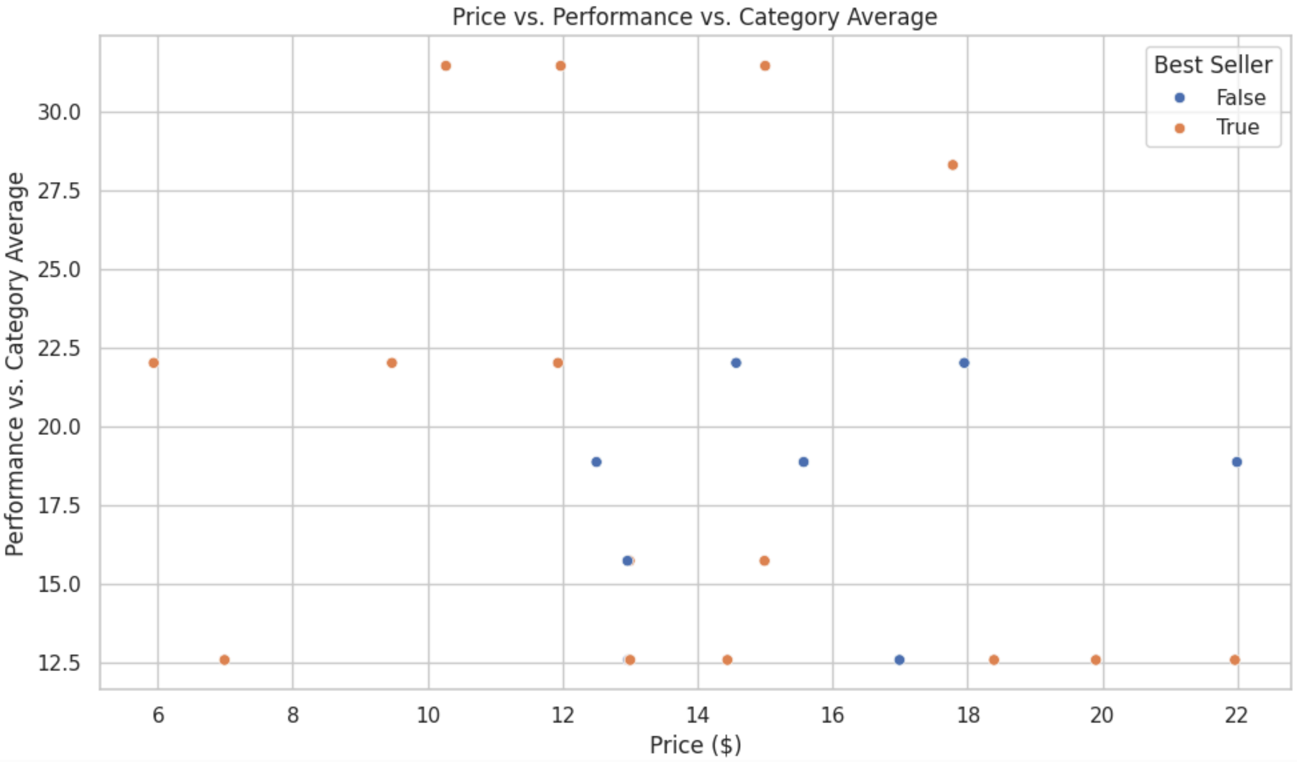
Next steps: Generate code with df View recommended plots New interactive sheet

```
# Convert 'isBestSeller' from string to boolean if needed
df['isBestSeller'] = df['isBestSeller'].astype(bool)

# Set visual style
sns.set(style="whitegrid")
```

```
# 1. Distribution of Star Ratings
plt.figure(figsize=(8, 5))
sns.histplot(df['stars'], bins=10, kde=True)
plt.title('Distribution of Product Star Ratings')
plt.xlabel('Star Rating')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```

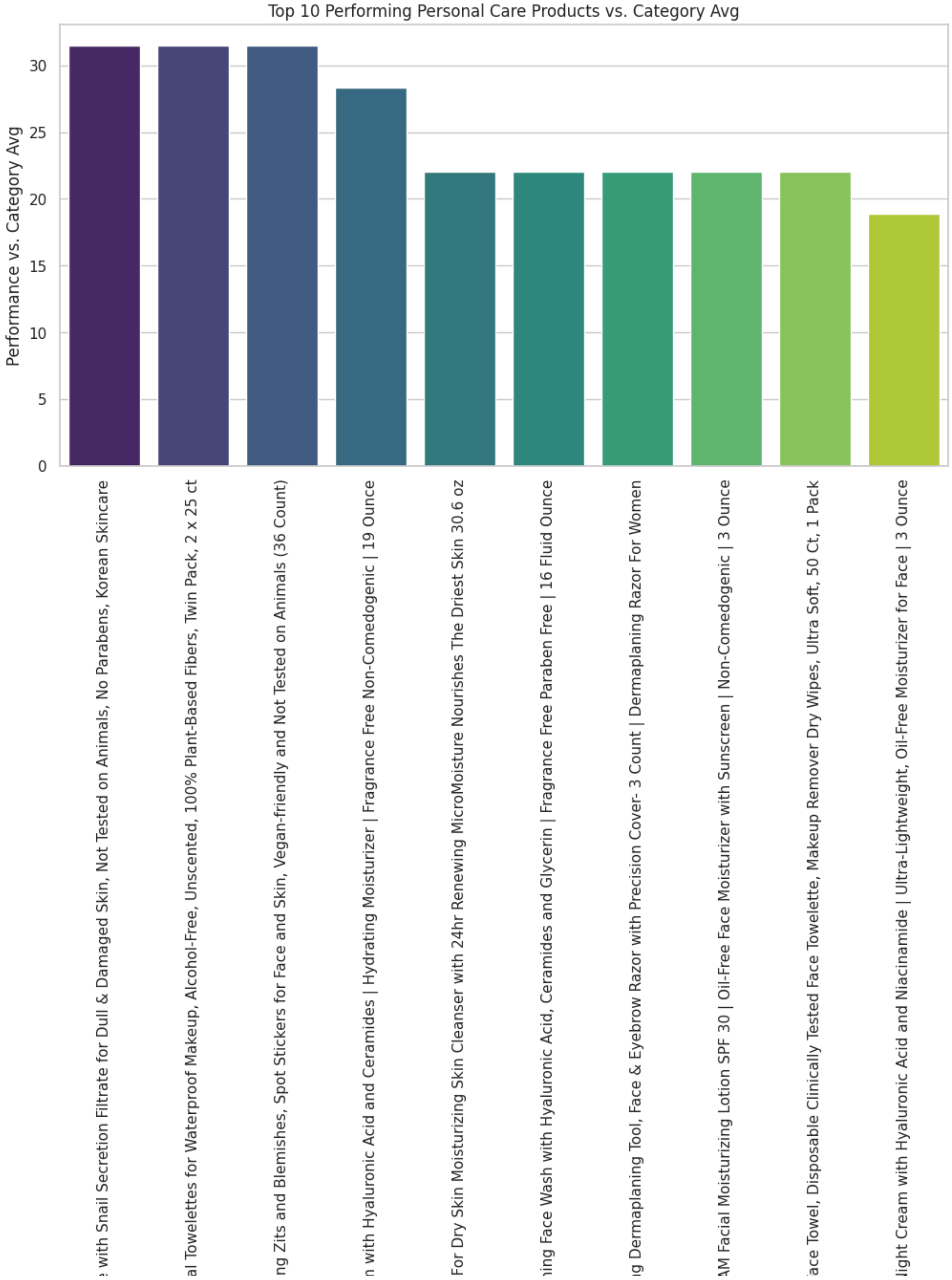


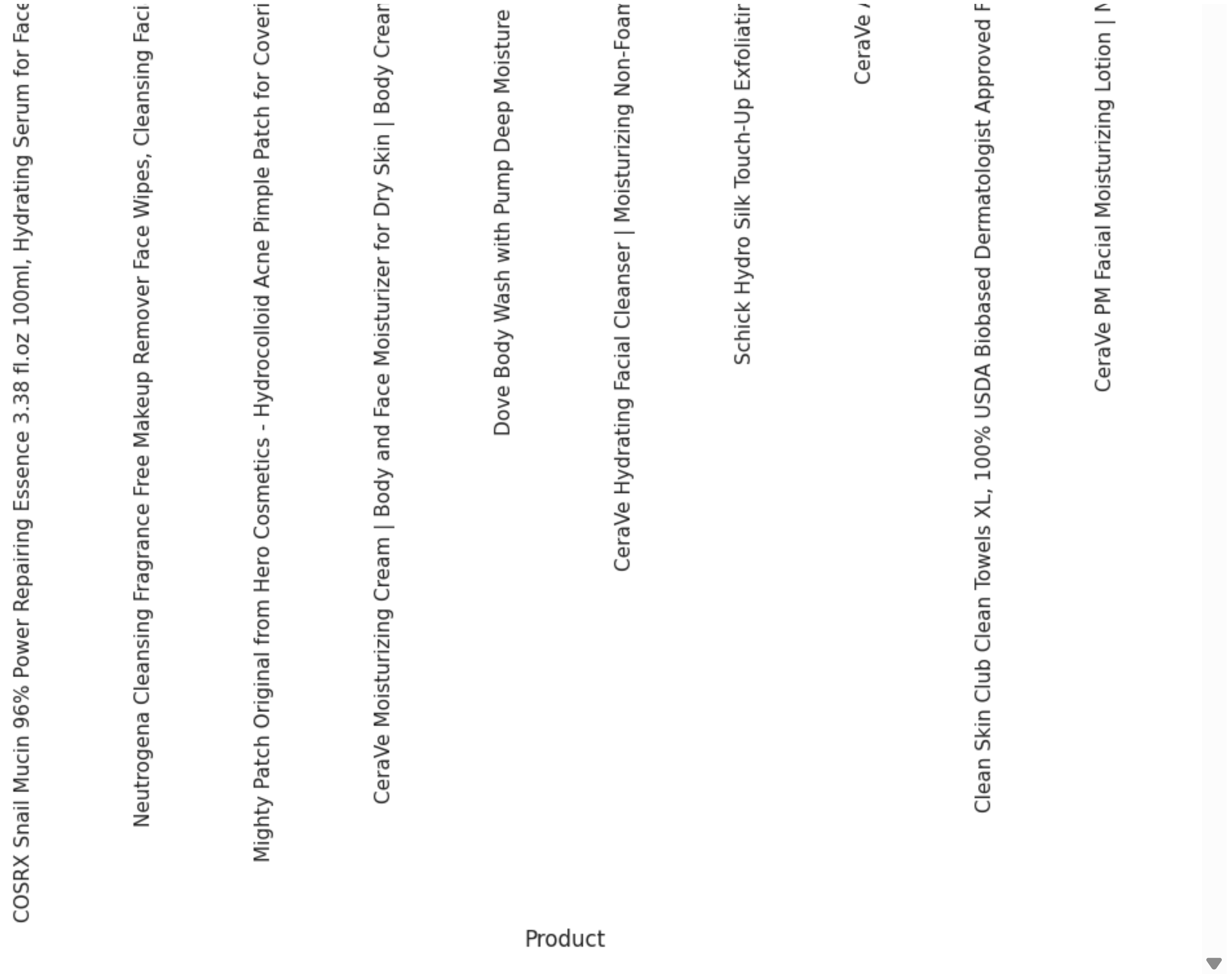
```
# 2. Price vs Performance Compared to Category Average
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='price', y='performance_vs_category_avg', hue='isBestSeller')
plt.title('Price vs. Performance vs. Category Average')
plt.xlabel('Price ($)')
plt.ylabel('Performance vs. Category Average')
plt.legend(title='Best Seller')
plt.tight_layout()
plt.show()
```



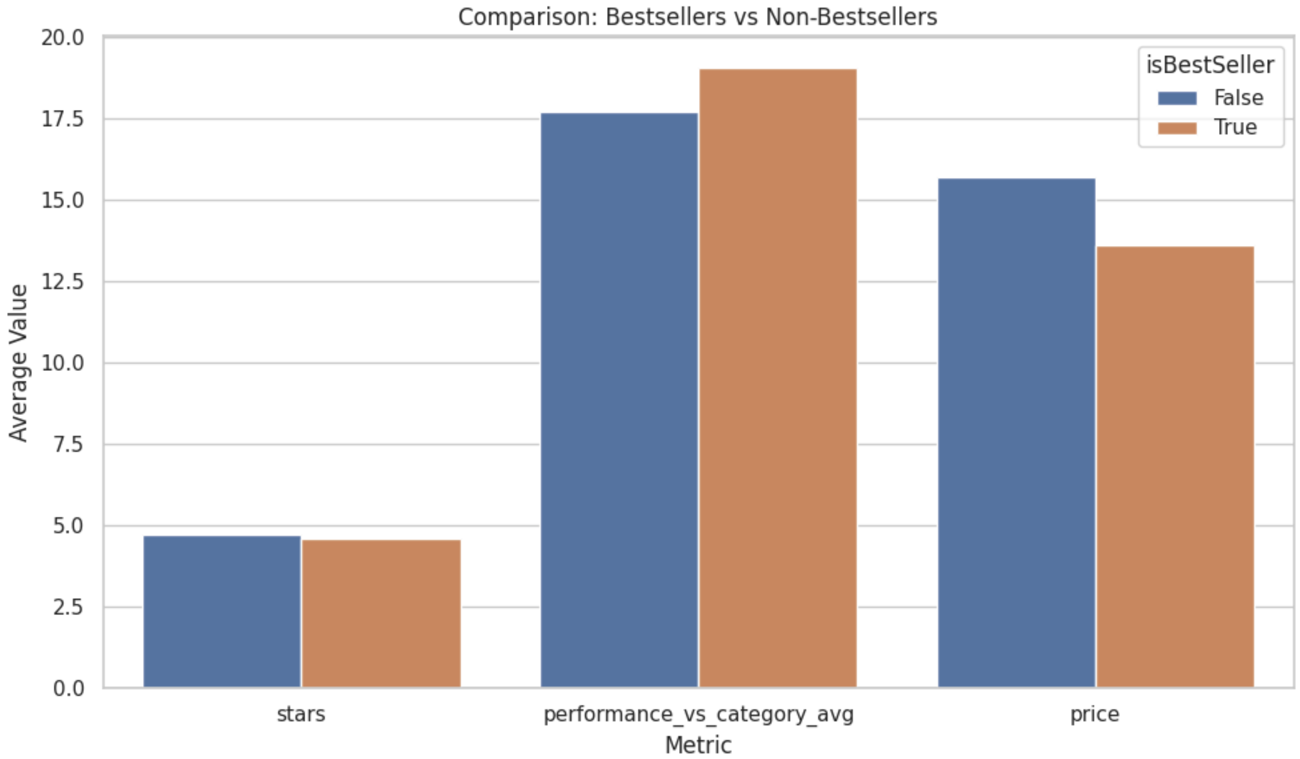
```
# 3. Top Performing Products
top_performers = df.sort_values(by='performance_vs_category_avg', ascending=False).head(10)
plt.figure(figsize=(12, 6))
sns.barplot(data=top_performers, x='title', y='performance_vs_category_avg', palette='viridis')
plt.title('Top 10 Performing Personal Care Products vs. Category Avg')
plt.xticks(rotation=90)
plt.ylabel('Performance vs. Category Avg')
plt.xlabel('Product')
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-21-77137231.py:4: FutureWarning:

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

  sns.barplot(data=top_performers, x='title', y='performance_vs_category_avg', palette='viridis')
/tmp/ipython-input-21-77137231.py:9: UserWarning: Tight layout not applied. The bottom and top margins cannot be made large enough to
  plt.tight_layout()
```




COSRX Snail Mucin 96% Power Repairing Essence 3.38 fl.oz 100ml, Hydrating Serum for Face
Neutrogena Cleansing Fragrance Free Makeup Remover Face Wipes, Cleansing Faci
Mighty Patch Original from Hero Cosmetics - Hydrocolloid Acne Pimple Patch for Coveri
CeraVe Moisturizing Cream | Body and Face Moisturizer for Dry Skin | Body Crear
Dove Body Wash with Pump Deep Moisture
CeraVe Hydrating Facial Cleanser | Moisturizing Non-Foam
Schick Hydro Silk Touch-Up Exfoliatin
CeraVe /
Clean Skin Club Clean Towels XL, 100% USDA Biobased Dermatologist Approved F
CeraVe PM Facial Moisturizing Lotion | N
Product

```
# 4. Bestseller vs Non-Bestseller - Average Rating & Performance
bestseller_stats = df.groupby('isBestSeller').agg({
    'stars': 'mean',
    'performance_vs_category_avg': 'mean',
    'price': 'mean'
}).reset_index()

bestseller_stats_melted = bestseller_stats.melt(id_vars='isBestSeller')

plt.figure(figsize=(10, 6))
sns.barplot(data=bestseller_stats_melted, x='variable', y='value', hue='isBestSeller')
plt.title('Comparison: Bestsellers vs Non-Bestsellers')
plt.ylabel('Average Value')
plt.xlabel('Metric')
plt.tight_layout()
plt.show()
```



Oversaturation Games and Toys

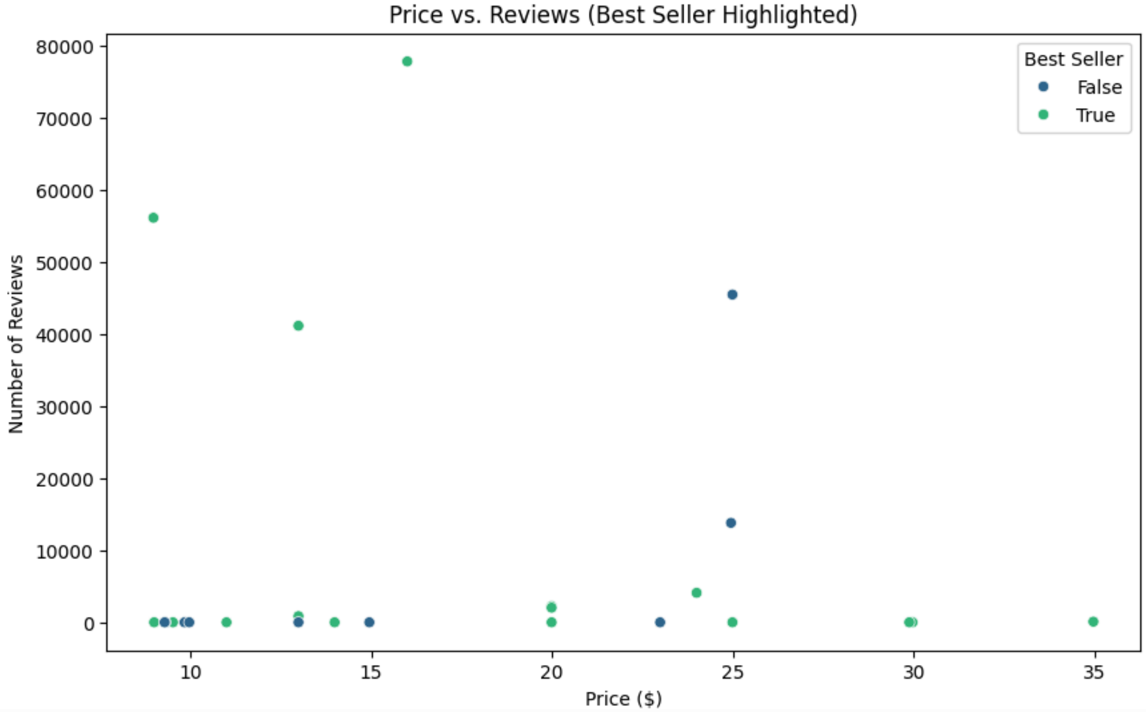
```
# Load the dataset
df = pd.read_csv("Games and Toys.csv")
```

```
df.head()
```

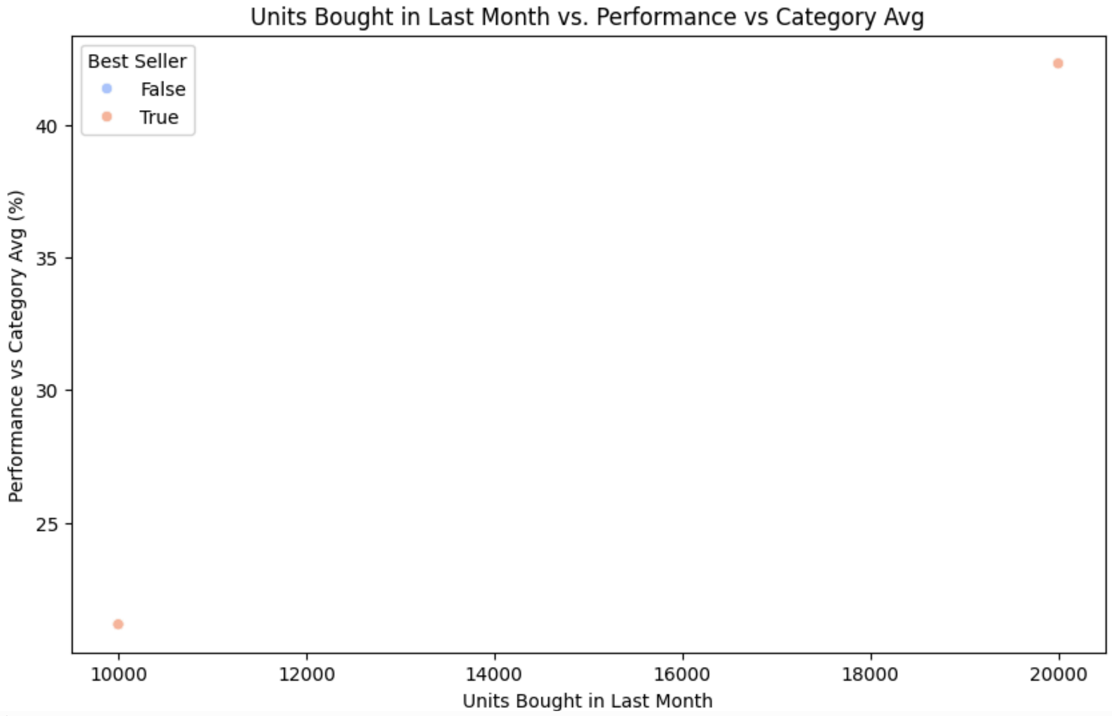
	product_id	title	stars	price	reviews	boughtInLastMonth	isBestSeller	performance_vs_category_avg
0	B07NXDJ52C	Sassy Stacks of Circles Stacking Ring STEM Lea...	4.8	8.98	56107	20000	True	42.33
1	B0BQNFZXTQ	COOKEEZ MAKERY Cinnamon Treatz Oven. Mix & Mak...	4.4	34.97	90	20000	True	42.33
2	B0BRT9C5S2	Air Hogs, Zero Gravity Sprint RC Car Wall Clim...	4.1	19.99	0	10000	True	21.16
3	B07H93M5X8	VTech Musical Rhymes Book, Red 1.74 x 8.76 x 7...	4.8	9.00	0	10000	True	21.16
4	B00D8STBHY	Hasbro Gaming Connect 4 Classic Grid,4 in a Ro...	4.8	9.52	0	10000	True	21.16

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

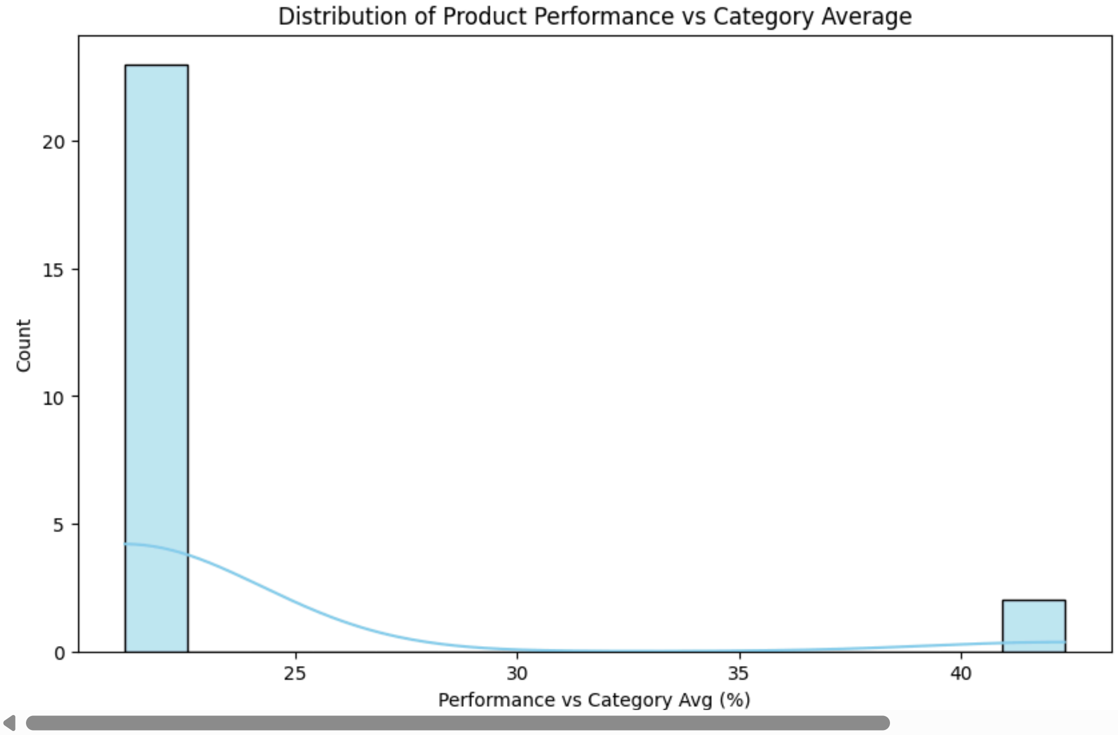
```
# --- 1. Price vs. Reviews ---
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x="price", y="reviews", hue="isBestSeller", palette="viridis")
plt.title("Price vs. Reviews (Best Seller Highlighted)")
plt.xlabel("Price ($)")
plt.ylabel("Number of Reviews")
plt.legend(title="Best Seller")
plt.show()
```



```
# --- 2. Bought in Last Month vs. Performance vs Category Avg ---
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x="boughtInLastMonth", y="performance_vs_category_avg", hue="isBestSeller", palette="coolwarm")
plt.title("Units Bought in Last Month vs. Performance vs Category Avg")
plt.xlabel("Units Bought in Last Month")
plt.ylabel("Performance vs Category Avg (%)")
plt.legend(title="Best Seller")
plt.show()
```

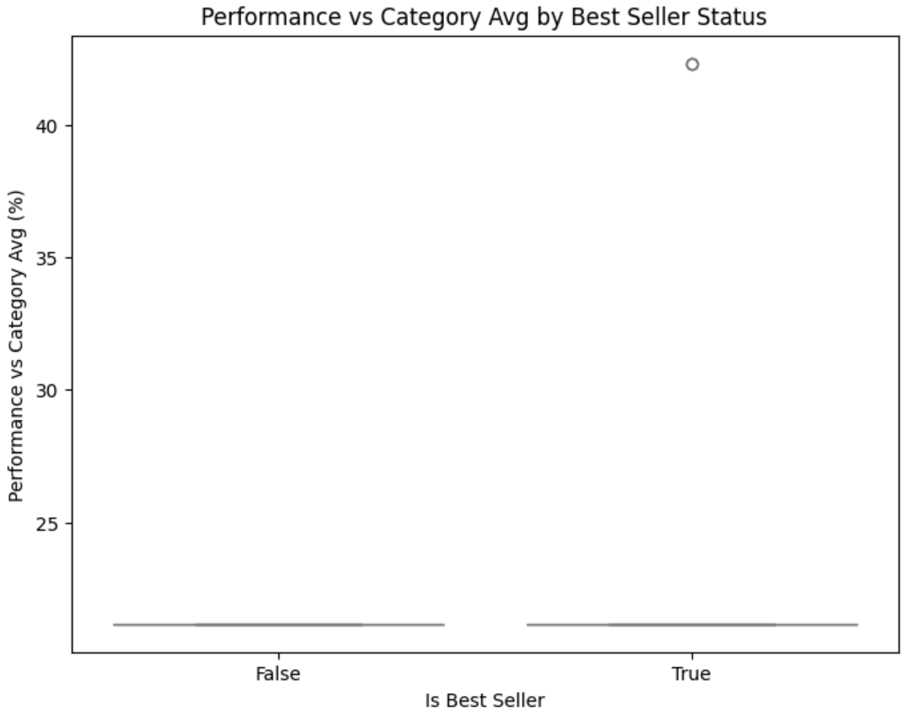


```
# --- 3. Distribution of Performance vs Category Avg ---
plt.figure(figsize=(10, 6))
sns.histplot(df["performance_vs_category_avg"], bins=15, kde=True, color="skyblue")
plt.title("Distribution of Product Performance vs Category Average")
plt.xlabel("Performance vs Category Avg (%)")
plt.ylabel("Count")
plt.show()
```



```
# --- 4. Boxplot: Performance by Best Seller Status ---
plt.figure(figsize=(8, 6))
sns.boxplot(data=df, x="isBestSeller", y="performance_vs_category_avg", palette="pastel")
plt.title("Performance vs Category Avg by Best Seller Status")
plt.xlabel("Is Best Seller")
plt.ylabel("Performance vs Category Avg (%)")
plt.show()
```

```
/tmp/ipython-input-8-3101975696.py:3: FutureWarning:

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

  sns.boxplot(data=df, x="isBestSeller", y="performance_vs_category_avg", palette="pastel")
```



Best Seller

```
# Load the dataset
df = pd.read_csv("Bestseller.csv")# Load the dataset
```

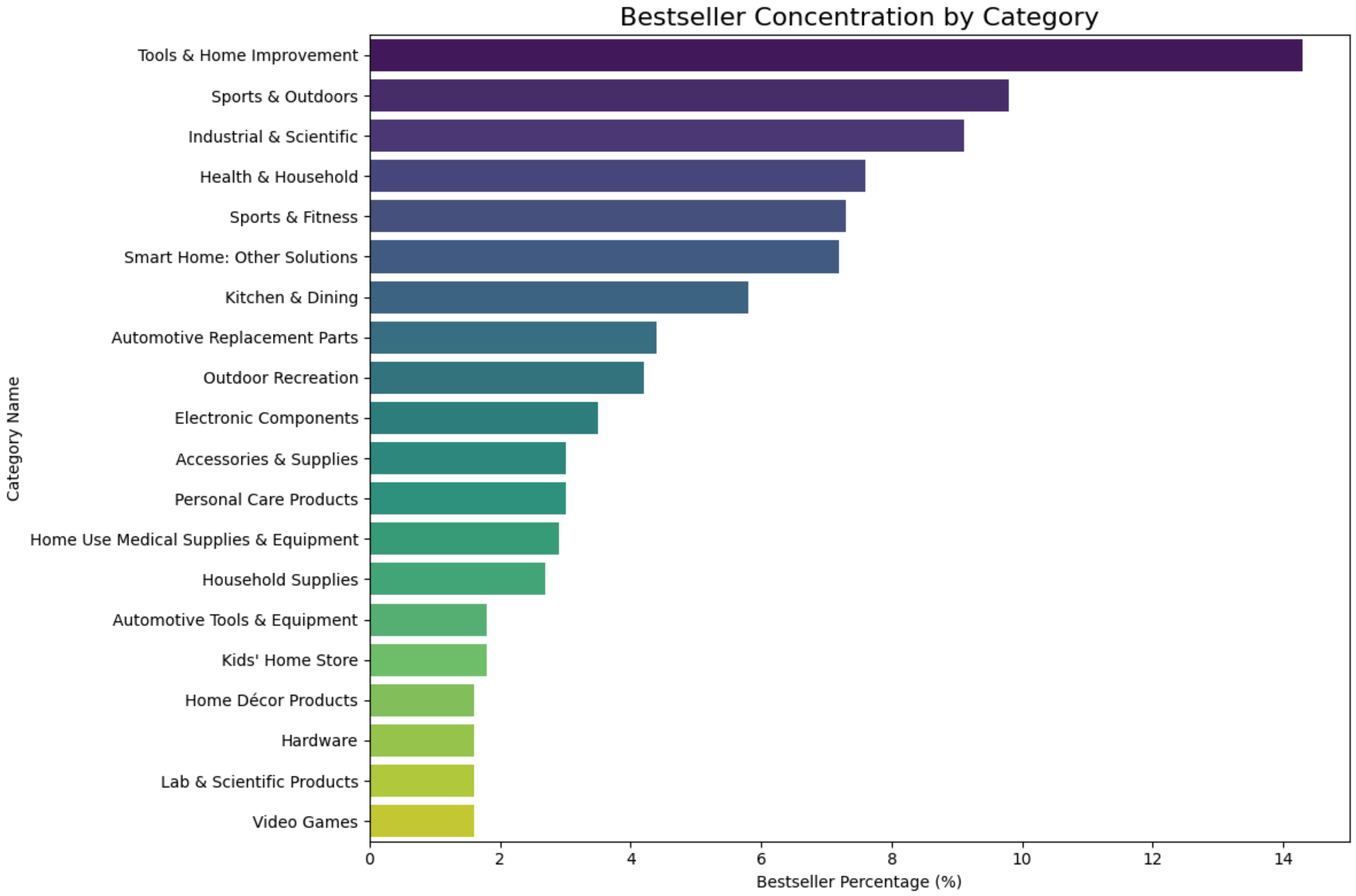
```
df.head()
```

	category_name	total_products	bestseller_count	bestseller_percentage	bestseller_avg_rating	regular_avg_rating	bestseller_avg_pri
0	Tools & Home Improvement	1678	240	14.3	4.53	4.53	28.
1	Sports & Outdoors	2625	256	9.8	4.54	4.52	34.
2	Industrial & Scientific	4403	399	9.1	4.57	4.56	20.
3	Health & Household	714	54	7.6	4.50	4.57	18.
4	Sports & Fitness	6604	483	7.3	4.50	4.46	26.

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

```
df_sorted = df.sort_values(by="bestseller_percentage", ascending=False)

# Plotting
plt.figure(figsize=(12, 8))
sns.barplot(data=df_sorted, x="bestseller_percentage", y="category_name", palette="viridis")
plt.title("Bestseller Concentration by Category", fontsize=16)
plt.xlabel("Bestseller Percentage (%)")
plt.ylabel("Category Name")
plt.tight_layout()
plt.show()
```

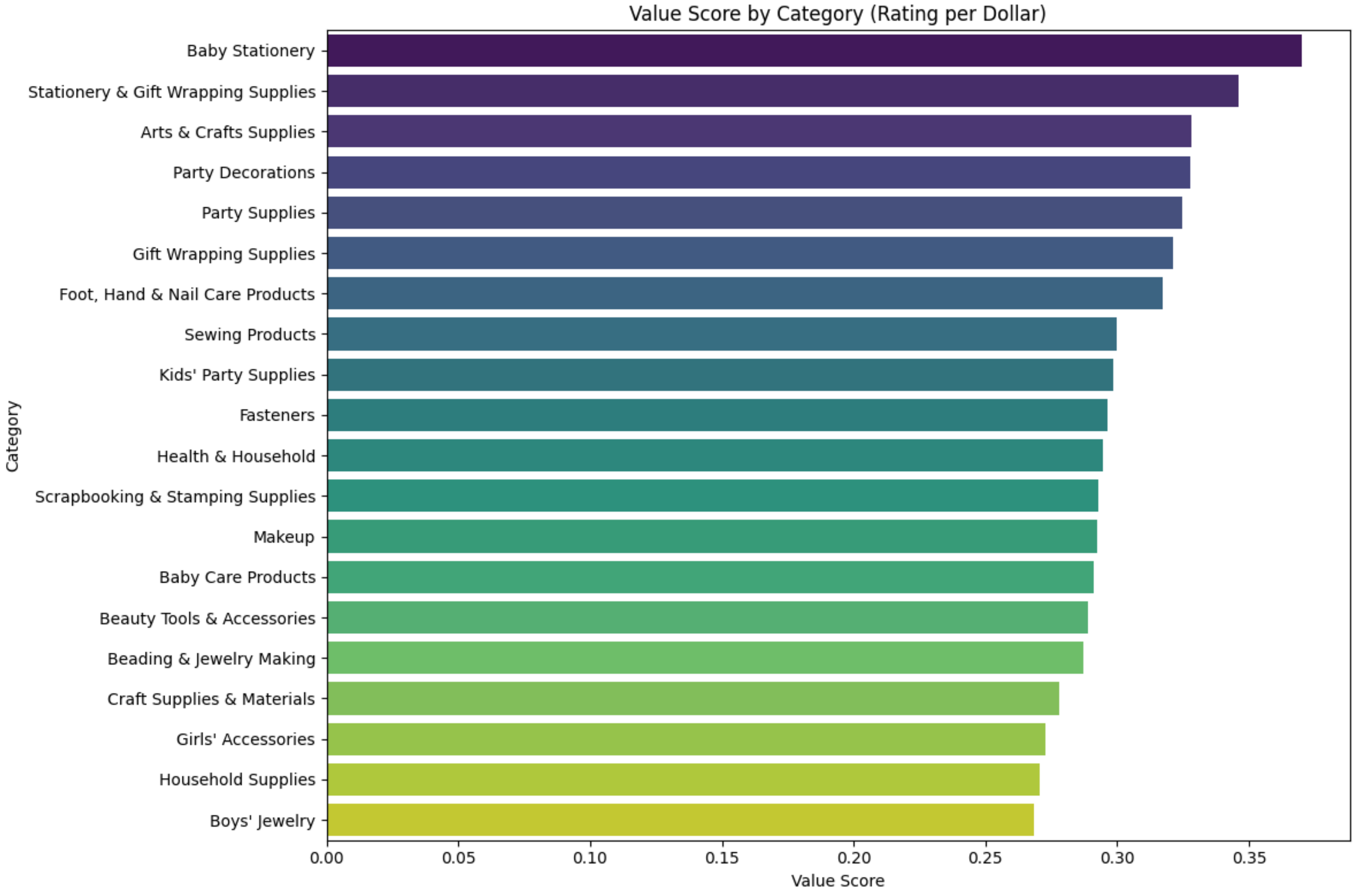
```
/tmp/ipython-input-11-1689300936.py:5: FutureWarning:

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

  sns.barplot(data=df_sorted, x="bestseller_percentage", y="category_name", palette="viridis")
```



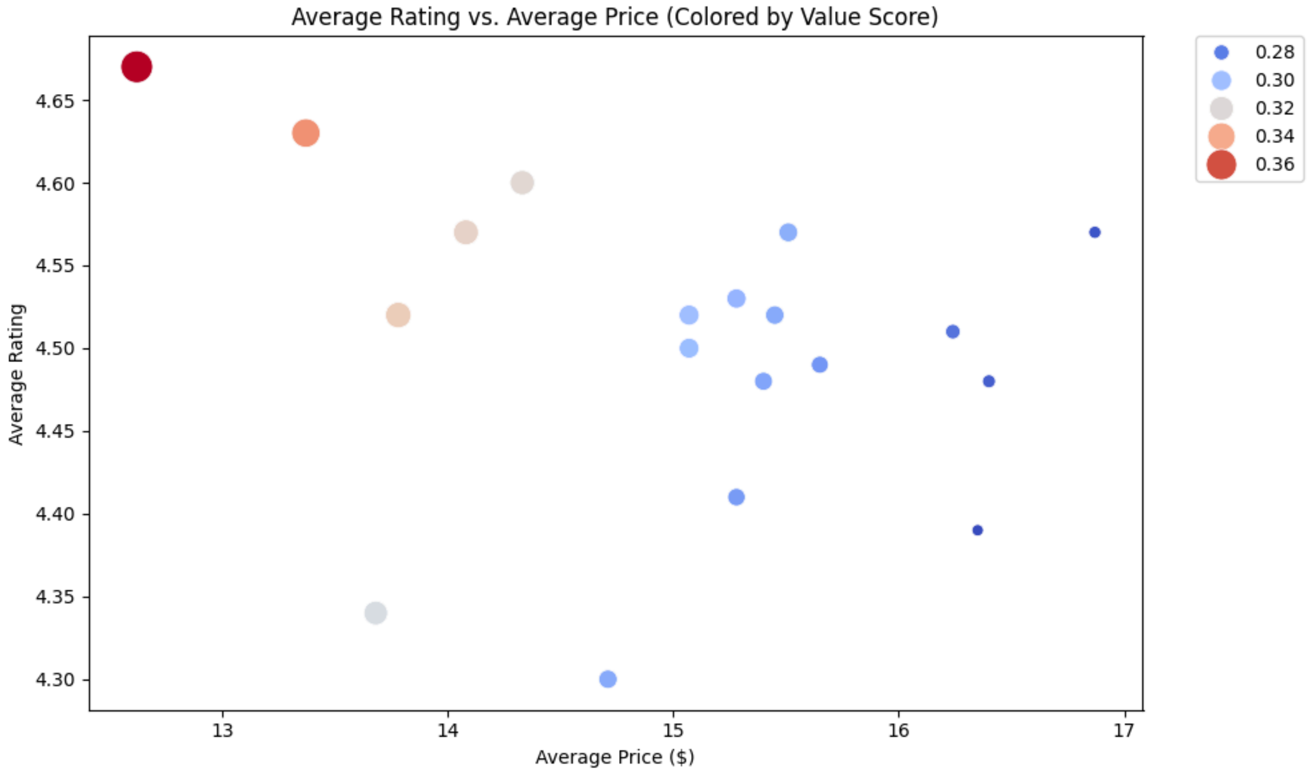
Value Analysis

```
# Load the dataset
df = pd.read_csv("Valuation.csv")# Load the dataset


# --- 1. Bar Chart of Value Score by Category ---
plt.figure(figsize=(12, 8))
sns.barplot(data=df.sort_values("value_score", ascending=False),
            y="category_name", x="value_score", palette="viridis")
plt.title("Value Score by Category (Rating per Dollar)")
plt.xlabel("Value Score")
plt.ylabel("Category")
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-14-1180536068.py:3: FutureWarning:

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

  sns.barplot(data=df.sort_values("value_score", ascending=False),
```



```
# --- 2. Scatter Plot: Avg Price vs Avg Rating, Colored by Value Score ---
plt.figure(figsize=(10, 6))
scatter = sns.scatterplot(data=df, x="avg_price", y="avg_rating", hue="value_score", size="value_score", palette="coolwarm", sizes=(40, 300),
plt.title("Average Rating vs. Average Price (Colored by Value Score)")
plt.xlabel("Average Price ($)")
plt.ylabel("Average Rating")
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.tight_layout()
plt.show()
```



```
# --- 3. Bubble Chart: Value Score vs Avg Monthly Sales, Size by Product Count ---
plt.figure(figsize=(12, 8))
bubble = plt.scatter(df["value_score"], df["avg_monthly_sales"],
                     s=df["product_count"] / 20, alpha=0.6, c=df["value_score"], cmap="plasma", edgecolors="w", linewidth=0.5)
plt.colorbar(label="Value Score")
plt.title("Value Score vs. Average Monthly Sales (Bubble size = Product Count)")
plt.xlabel("Value Score")
plt.ylabel("Average Monthly Sales")
plt.grid(True)
plt.tight_layout()
plt.show()
```