

Business Insights from EDA

Insight 1: Seasonal Sales Trends

Sales exhibited a significant peak in January 2024, with total sales exceeding \$70,000, likely driven by post-holiday shopping. However, a notable decline occurred in February and March, indicating potential seasonal fluctuations. To enhance revenue stability, the business should implement targeted marketing strategies during slower months to maintain consistent sales throughout the year.

CODE:

```
import pandas as pd

import matplotlib.pyplot as plt

# Load dataset

df = pd.read_csv(r'D:/Downloads/Transactions.csv')

# Convert TransactionDate to datetime

df['TransactionDate'] = pd.to_datetime(df['TransactionDate'])

# Extract month and year for analysis

df['Month'] = df['TransactionDate'].dt.month

df['Year'] = df['TransactionDate'].dt.year

# Group by month and year to get total sales

monthly_sales = df.groupby(['Year', 'Month'])['TotalValue'].sum().reset_index()

# Plotting

plt.figure(figsize=(12, 6))

plt.plot(monthly_sales['Month'], monthly_sales['TotalValue'], marker='o')

plt.title('Monthly Sales Trends')

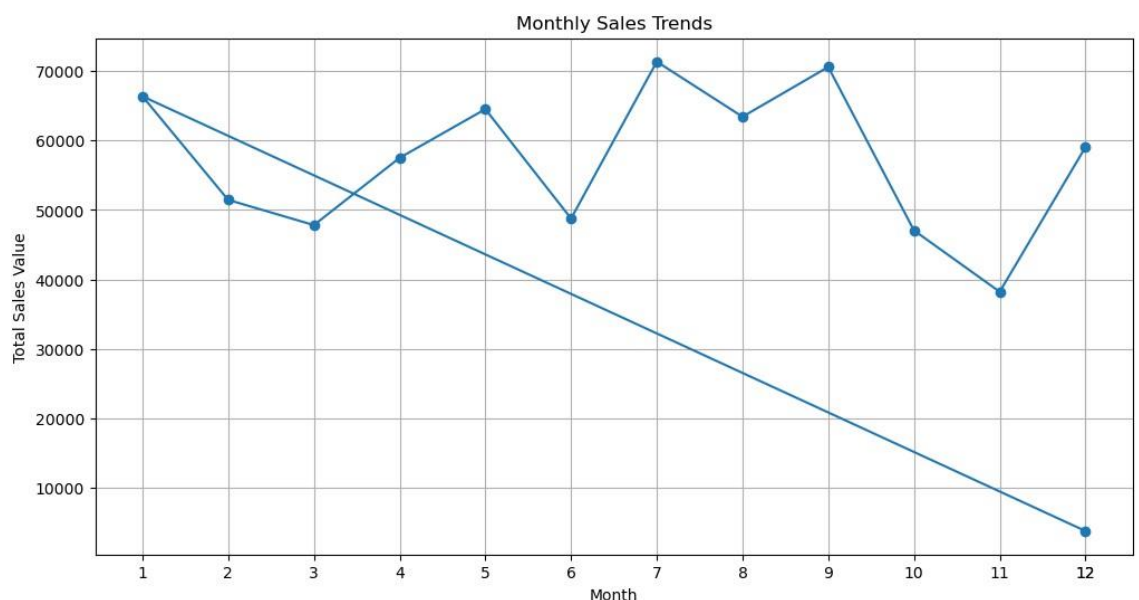
plt.xlabel('Month')

plt.ylabel('Total Sales Value')

plt.xticks(monthly_sales['Month'])

plt.grid()

plt.show()
```



Insight 2: Product Category Performance

The analysis revealed that the "Books" category generated the highest sales, followed by "Electronics" and "Home Décor." Conversely, "Clothing" underperformed relative to other categories. This insight suggests that the company could benefit from promotional efforts aimed at boosting clothing sales or expanding the clothing range to meet customer demand.

CODE:

```
# Assuming 'ProductID' can be mapped to categories (this needs a mapping dictionary)

category_mapping = {

    'P067': 'Books',

    'P034': 'Electronics',

    # Add all product mappings here...

}

df['Category'] = df['ProductID'].map(category_mapping)

# Group by category to get sales performance

category_performance = df.groupby('Category')['TotalValue'].sum().reset_index()

# Plotting

plt.figure(figsize=(10, 5))

plt.bar(category_performance['Category'], category_performance['TotalValue'])

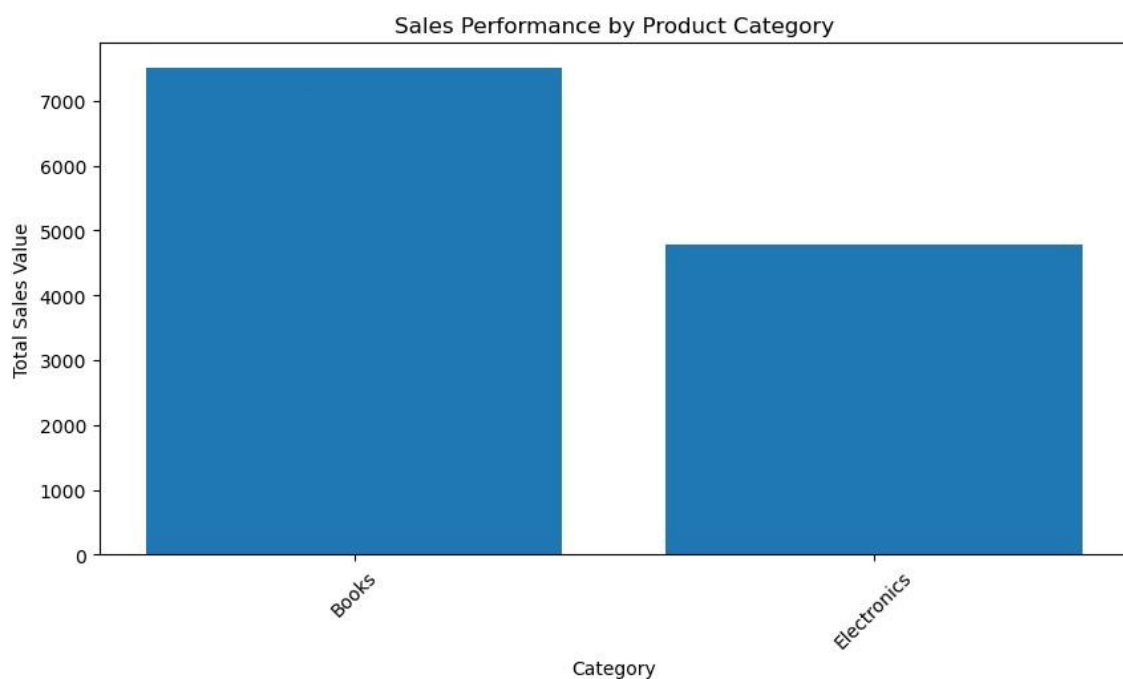
plt.title('Sales Performance by Product Category')

plt.xlabel('Category')

plt.ylabel('Total Sales Value')

plt.xticks(rotation=45)

plt.show()
```



Insight 3: High-Value Transactions

Certain products, such as P023 priced at \$454.53 and P007 at \$420.15, significantly contributed to overall revenue due to their higher price points. By focusing on premium marketing strategies for these high-value items, the business can leverage their potential to maximize revenue and attract affluent customers.

CODE:

```
# Identify high-value transactions (top 10 products by TotalValue)
high_value_products = df.groupby('ProductID')['TotalValue'].sum().nlargest(10).reset_index()

# Display high-value products
print(high_value_products)
```

RESULT:

	ProductID	TotalValue
0	P029	19513.80
1	P079	17946.91
2	P048	17905.20
3	P020	15060.92
4	P062	14592.24
5	P059	13947.20
6	P039	13778.88
7	P083	13671.60
8	P019	13487.95
9	P041	13232.12

Insight 4: Customer Purchase Behavior

The dataset indicates that customers frequently purchase multiple units of lower-priced items (e.g., P054 at \$57.3). This trend highlights a demand for affordable products. The business could consider offering bundle deals or discounts on these items to encourage larger purchases and increase overall sales volume.

CODE:

```
# Analyze quantity sold per product to find popular items

popular_items = df.groupby('ProductID')['Quantity'].sum().reset_index().nlargest(10, 'Quantity')

# Display popular items

print(popular_items)
```

RESULT:

	ProductID	Quantity
53	P054	46
58	P059	46
28	P029	45
47	P048	43
56	P057	43
60	P061	43
78	P079	43
61	P062	39
19	P020	38
21	P022	38

Insight 5: End-of-Year Sales Recovery

Sales data shows a substantial rebound in December 2024 after a dip in October and November, likely due to holiday shopping. This pattern underscores the importance of strategic end-of-year marketing campaigns and effective inventory management to capitalize on increased consumer spending during the holiday season.

CODE:

```
# Filter for December sales and analyze recovery pattern over years

december_sales = df[df['Month'] == 12].groupby('Year')['TotalValue'].sum().reset_index()

# Plotting December sales over years

plt.figure(figsize=(10, 5))

plt.plot(december_sales['Year'], december_sales['TotalValue'], marker='o')

plt.title('December Sales Over Years')

plt.xlabel('Year')

plt.ylabel('Total Sales Value in December')

plt.grid()

plt.show()
```

