2hfgyr07j

January 26, 2025

```
[1]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
```

```
[4]: customers = pd.read_csv(r"C:\Users\vishn\Downloads\Customers - Customers.csv")
     products = pd.read_csv(r"C:\Users\vishn\Downloads\Products - Products.csv")
     transactions = pd.read_csv(r"C:\Users\vishn\Downloads\Transactions -␣
      ↪Transactions.csv")
```

```
[8]: customers.head()
```

```
[8]:   CustomerID        CustomerName         Region  SignupDate
     0      C0001    Lawrence Carroll  South America  2022-07-10
     1      C0002      Elizabeth Lutz           Asia  2022-02-13
     2      C0003      Michael Rivera  South America  2024-03-07
     3      C0004  Kathleen Rodriguez  South America  2022-10-09
     4      C0005         Laura Weber           Asia  2022-08-15
```

```
[9]: products.head()
```

```
[9]:   ProductID              ProductName     Category   Price
     0      P001     ActiveWear Biography        Books  169.30
     1      P002    ActiveWear Smartwatch  Electronics  346.30
     2      P003  ComfortLiving Biography        Books   44.12
     3      P004            BookWorld Rug   Home Decor   95.69
     4      P005           TechPro T-Shirt     Clothing  429.31
```

```
[10]: transactions.head()
```

```
[10]:   TransactionID CustomerID ProductID      TransactionDate  Quantity  \
     0        T00001      C0199      P067  2024-08-25 12:38:23         1
     1        T00112      C0146      P067  2024-05-27 22:23:54         1
     2        T00166      C0127      P067   2024-04-25 7:38:55         1
     3        T00272      C0087      P067  2024-03-26 22:55:37         2
     4        T00363      C0070      P067  2024-03-21 15:10:10         3
```

```
   TotalValue   Price
0      300.68  300.68
1      300.68  300.68
2      300.68  300.68
3      601.36  300.68
4      902.04  300.68
```

[11]:
```
customers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   CustomerID    200 non-null    object
 1   CustomerName  200 non-null    object
 2   Region        200 non-null    object
 3   SignupDate    200 non-null    object
dtypes: object(4)
memory usage: 6.4+ KB
```

[12]:
```
products.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   ProductID    100 non-null    object 
 1   ProductName  100 non-null    object 
 2   Category     100 non-null    object 
 3   Price        100 non-null    float64
dtypes: float64(1), object(3)
memory usage: 3.3+ KB
```

[13]:
```
transactions.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   TransactionID    1000 non-null   object 
 1   CustomerID       1000 non-null   object 
 2   ProductID        1000 non-null   object 
 3   TransactionDate  1000 non-null   object 
 4   Quantity         1000 non-null   int64  
 5   TotalValue       1000 non-null   float64
 6   Price             1000 non-null   float64
dtypes: float64(2), int64(1), object(4)
memory usage: 54.8+ KB
```

[21]:

```
customers.isnull().sum()
```

[21]:

```
CustomerID      0
CustomerName    0
Region          0
SignupDate      0
dtype: int64
```

[22]:

```
products.isnull().sum()
```

[22]:

```
ProductID      0
ProductName    0
Category       0
Price          0
dtype: int64
```

[23]:

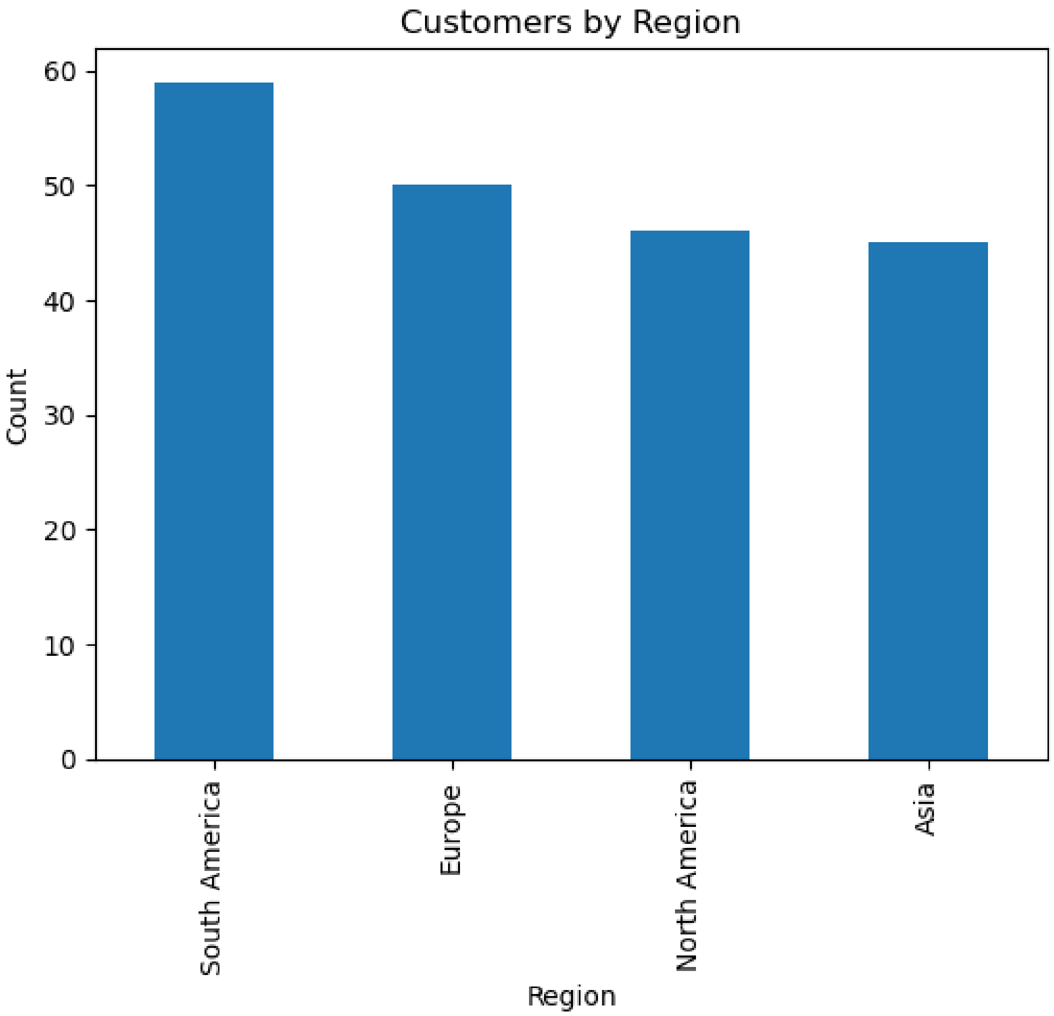
```
transactions.isnull().sum()
```

[23]:

```
TransactionID      0
CustomerID         0
ProductID          0
TransactionDate    0
Quantity           0
TotalValue         0
Price              0
dtype: int64
```

[55]:

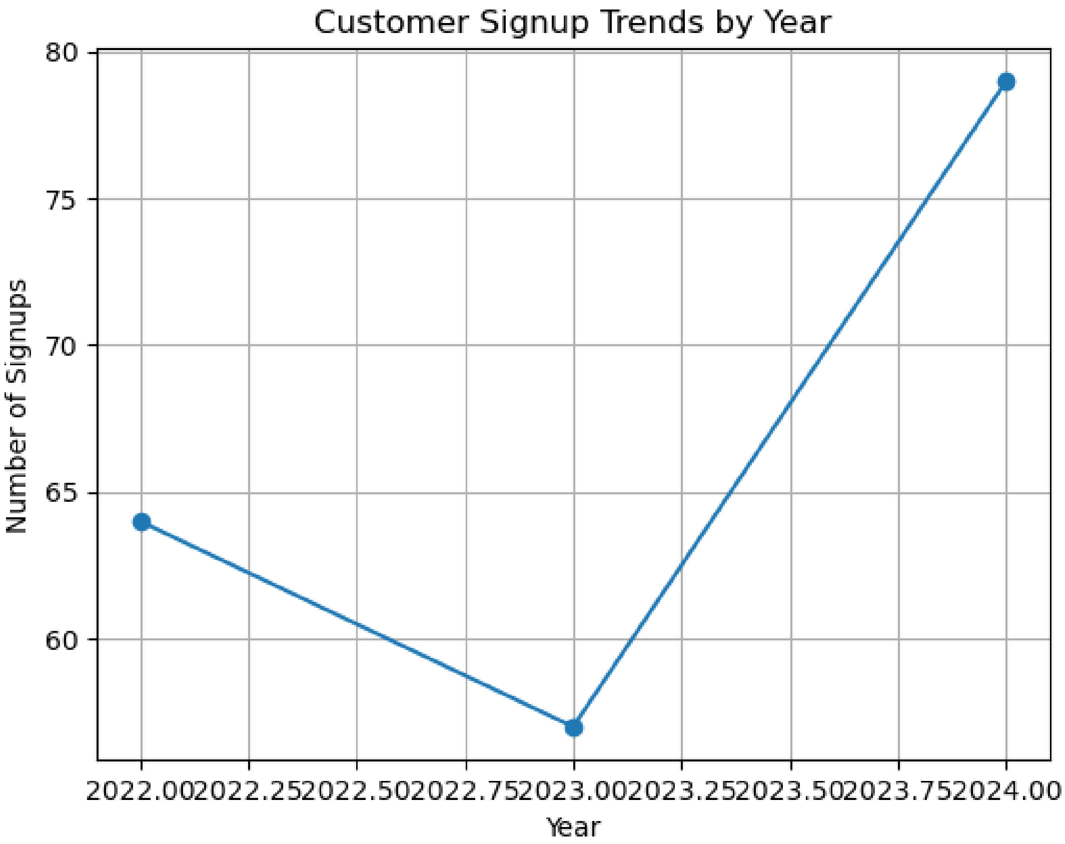
```
region_counts = customers['Region'].value_counts()
region_counts.plot(kind='bar', title='Customers by Region')
plt.xlabel('Region')
plt.ylabel('Count')
plt.show()
```



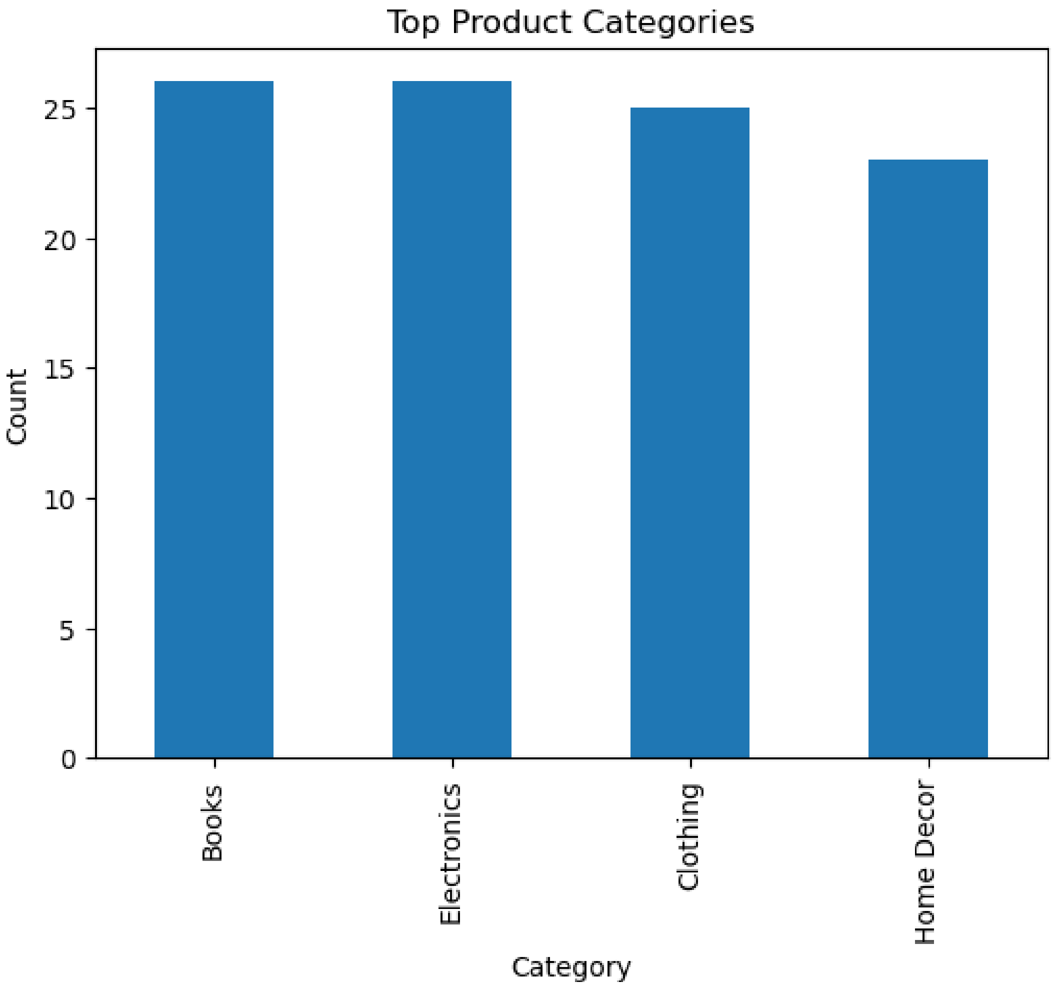
[54]:

```
customers['SignupDate'] = pd.to_datetime(customers['SignupDate'])
signup_trends = customers['SignupDate'].dt.year.value_counts().sort_index()
print(signup_trends)
signup_trends.plot(kind='line', marker='o', title='Customer Signup Trends by Year')
plt.xlabel('Year')
plt.ylabel('Number of Signups')
plt.grid()
plt.show()
```

```
SignupDate
2022    64
2023    57
2024    79
Name: count, dtype: int64
```

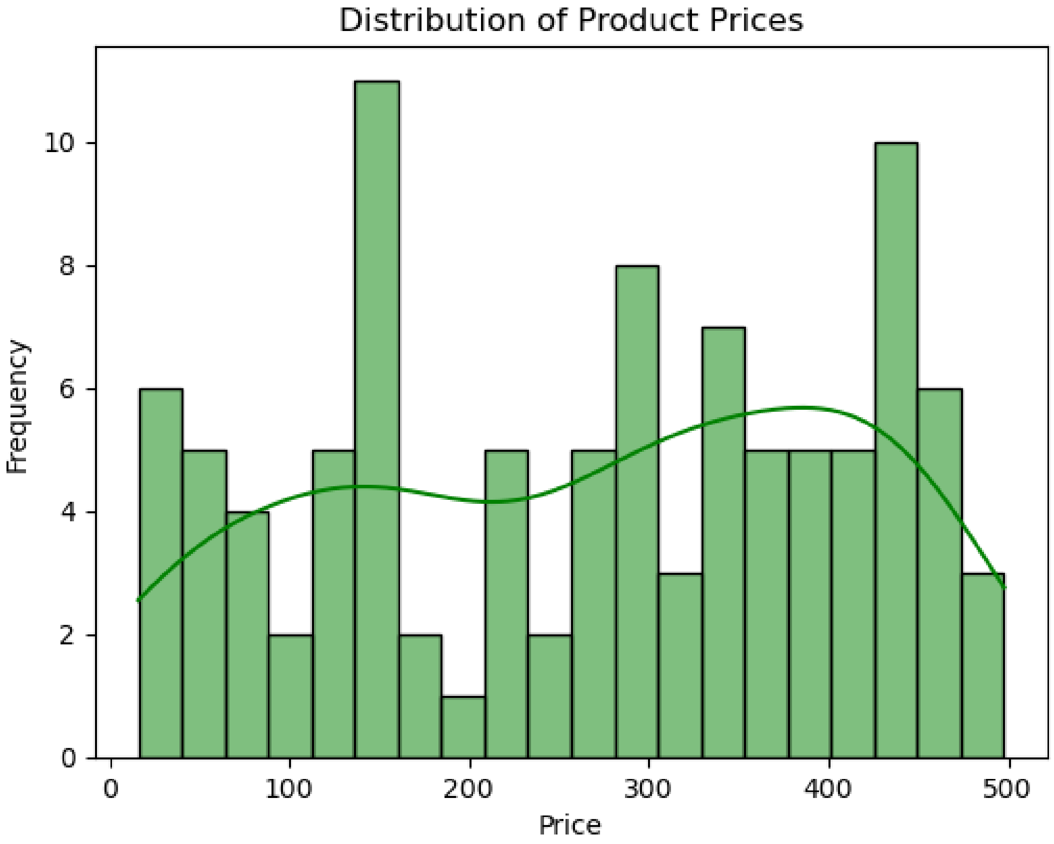


[53]:
```
category_counts = products['Category'].value_counts()
category_counts.plot(kind='bar', title='Top Product Categories')
plt.xlabel('Category')
plt.ylabel('Count')
plt.show()
```



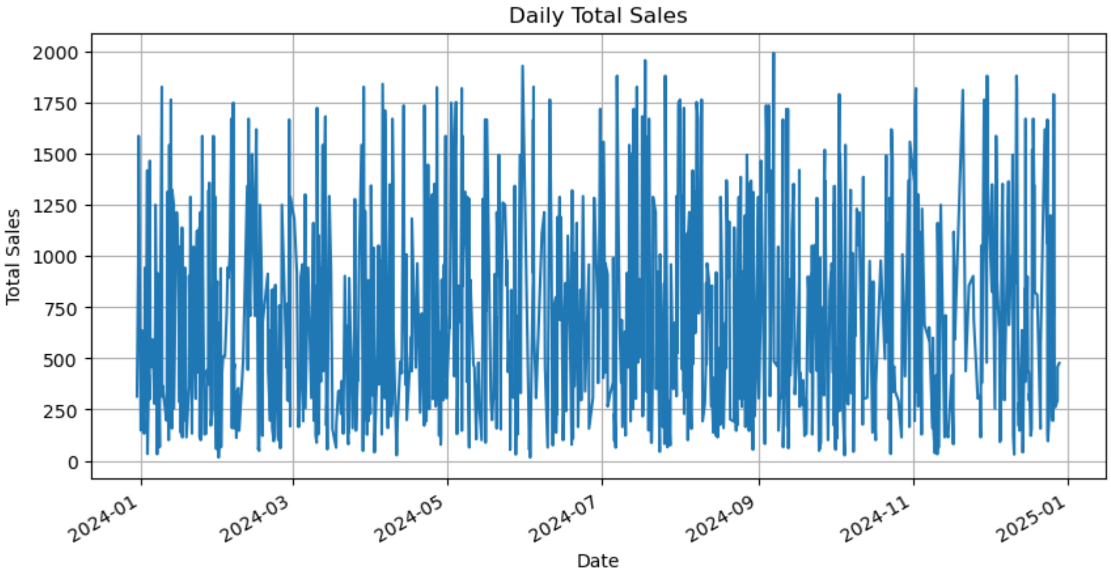
[48]:

```
products['Price'].describe()
sns.histplot(products['Price'], kde=True, bins=20, color='green')
plt.title('Distribution of Product Prices')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.show()
```

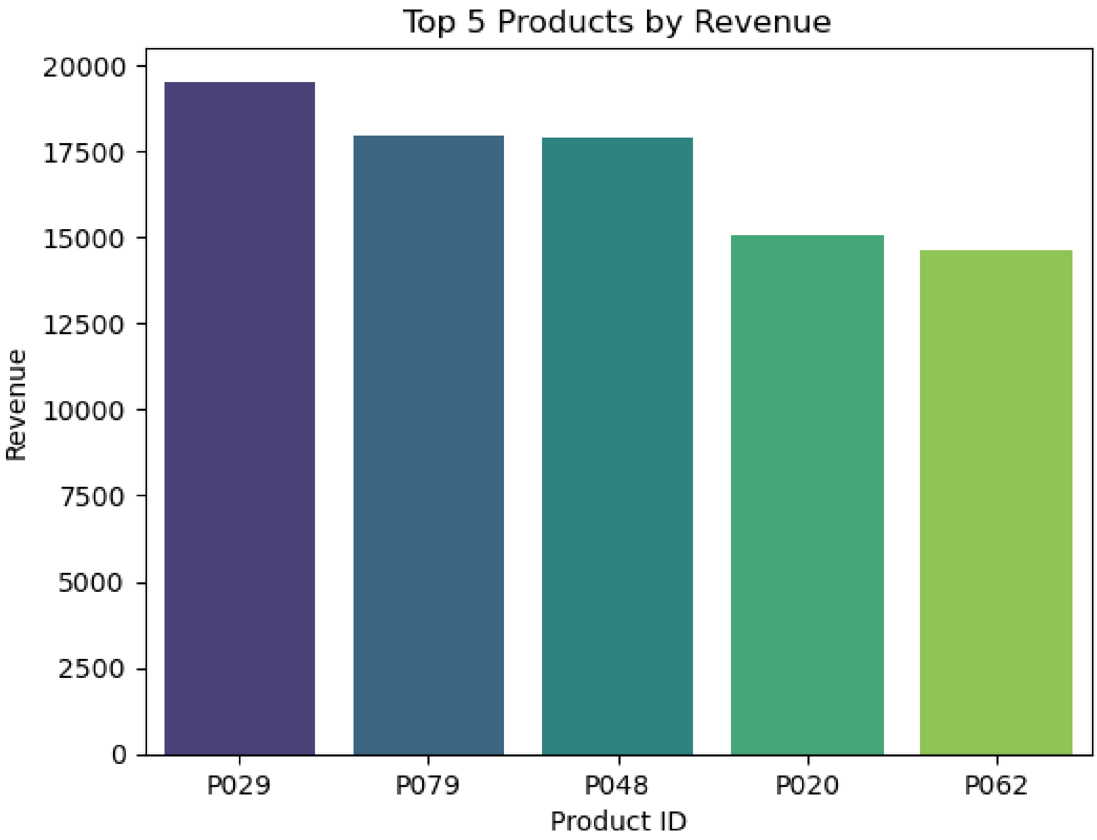


[51]:

```
transactions['TransactionDate'] = pd.
 ↪to_datetime(transactions['TransactionDate'])
sales_by_date = transactions.groupby('TransactionDate')['TotalValue'].sum()
plt.figure(figsize=(10, 5))
sales_by_date.plot(title='Daily Total Sales')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.grid()
plt.show()
```

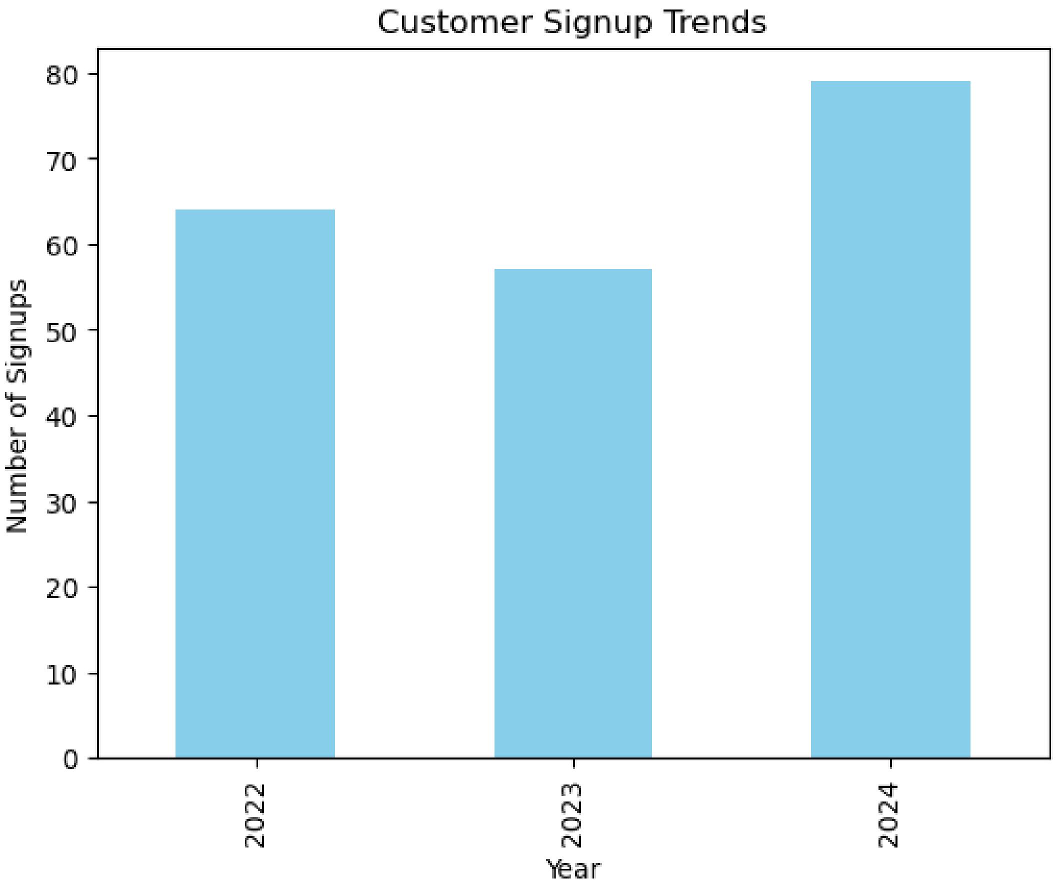


```
[50]: product_revenue = transactions.groupby('ProductID')['TotalValue'].sum().
        ↪sort_values(ascending=False)
      top_products = product_revenue.head(5)
      sns.barplot(x=top_products.index, y=top_products.values, palette='viridis')
      plt.title('Top 5 Products by Revenue')
      plt.xlabel('Product ID')
      plt.ylabel('Revenue')
      plt.show()
```



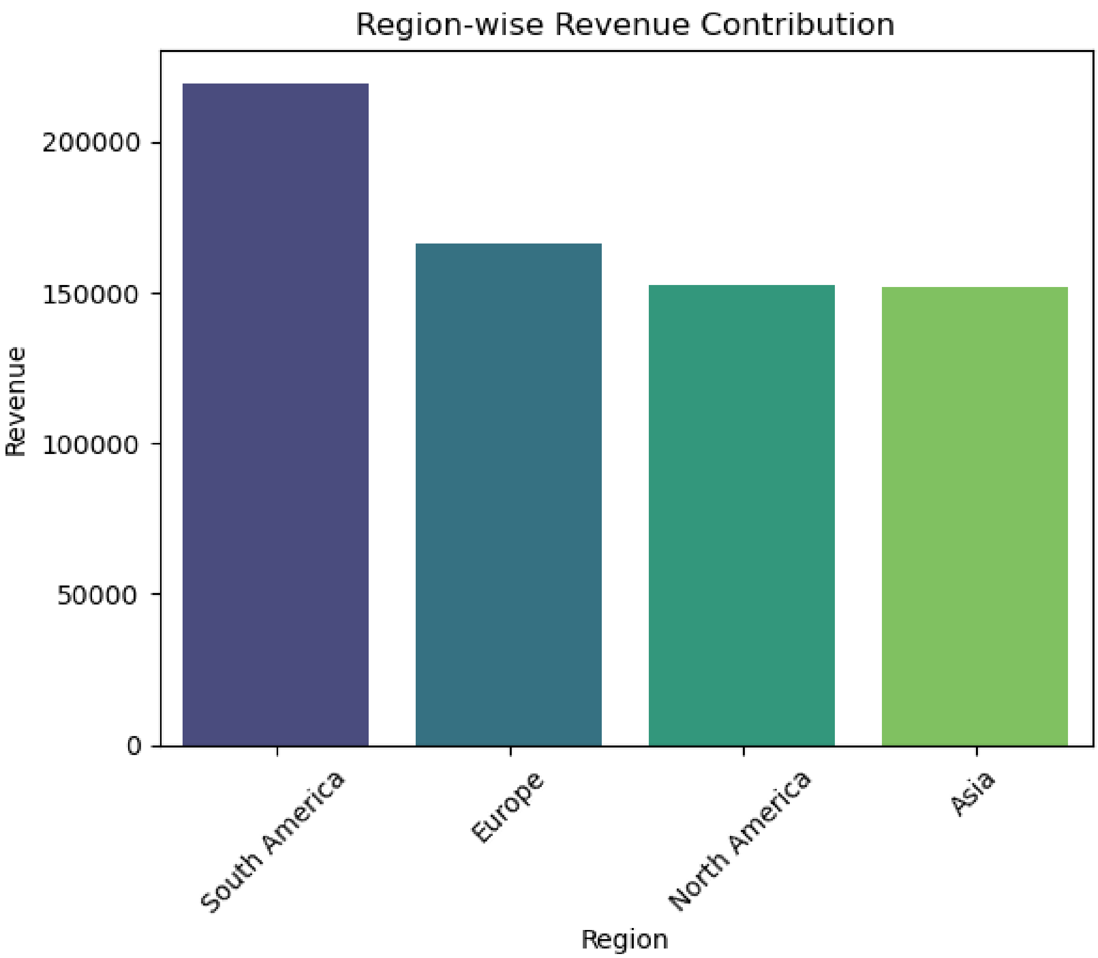
[31]:
```
merged_data = transactions.merge(customers, on='CustomerID').merge(products,␣
    ↪on='ProductID')
```

[32]:
```
signup_trends = customers['SignupDate'].dt.year.value_counts().sort_index()
signup_trends.plot(kind='bar', color='skyblue', title='Customer Signup Trends')
plt.xlabel('Year')
plt.ylabel('Number of Signups')
plt.show()
```

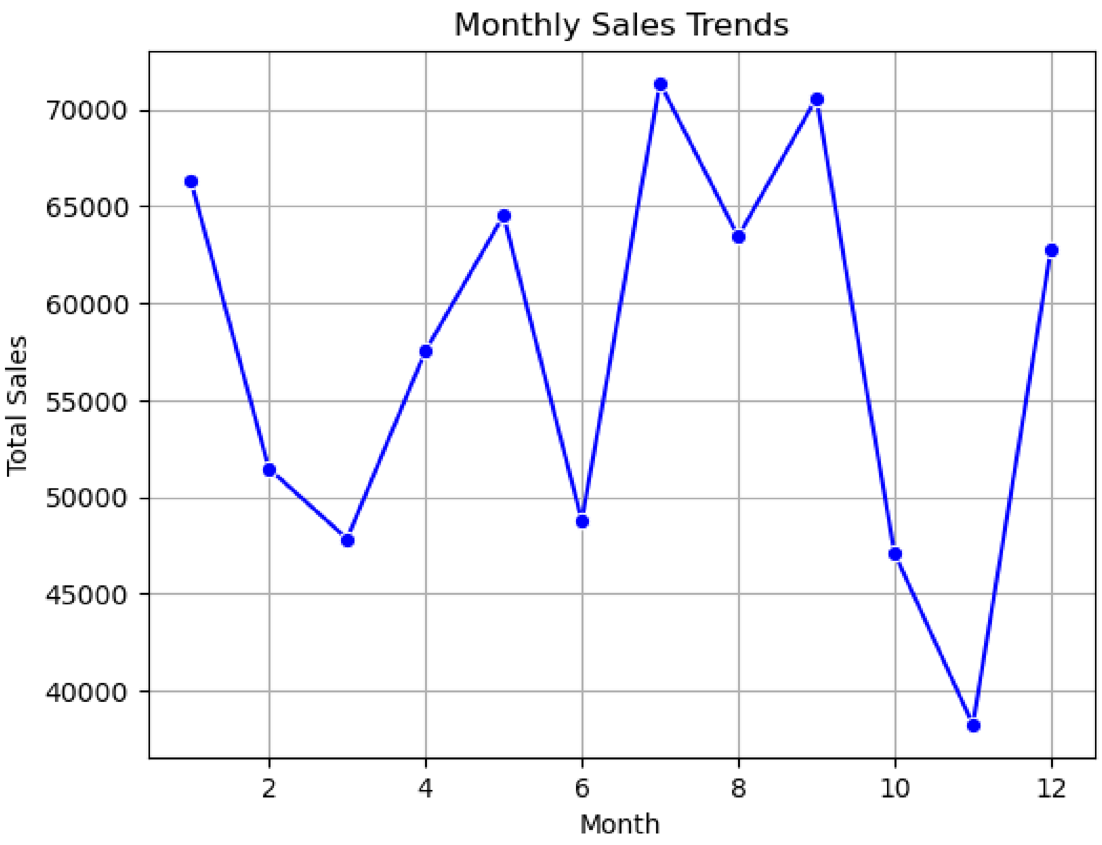


[33]:

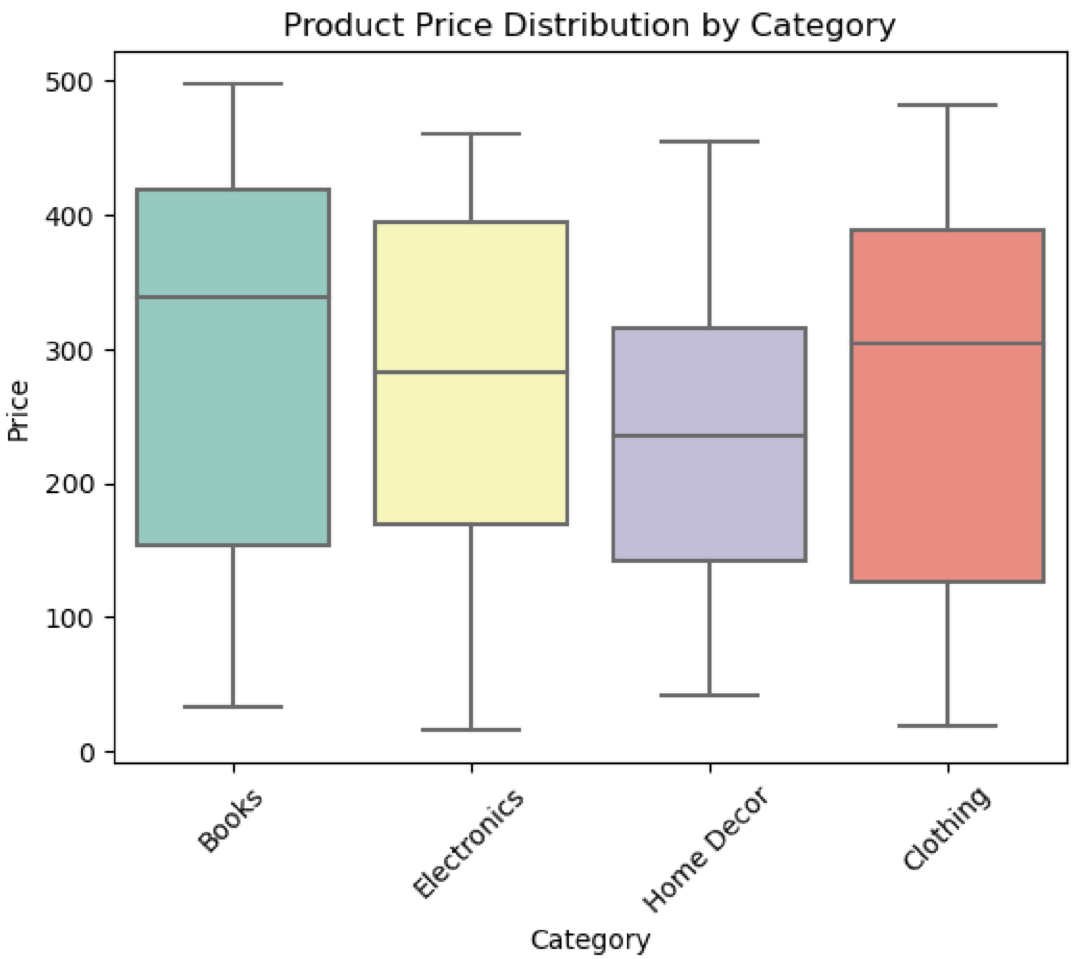
```
region_revenue = merged_data.groupby('Region')['TotalValue'].sum().
  ↪sort_values(ascending=False)
sns.barplot(x=region_revenue.index, y=region_revenue.values, palette='viridis')
plt.title('Region-wise Revenue Contribution')
plt.xlabel('Region')
plt.ylabel('Revenue')
plt.xticks(rotation=45)
plt.show()
```



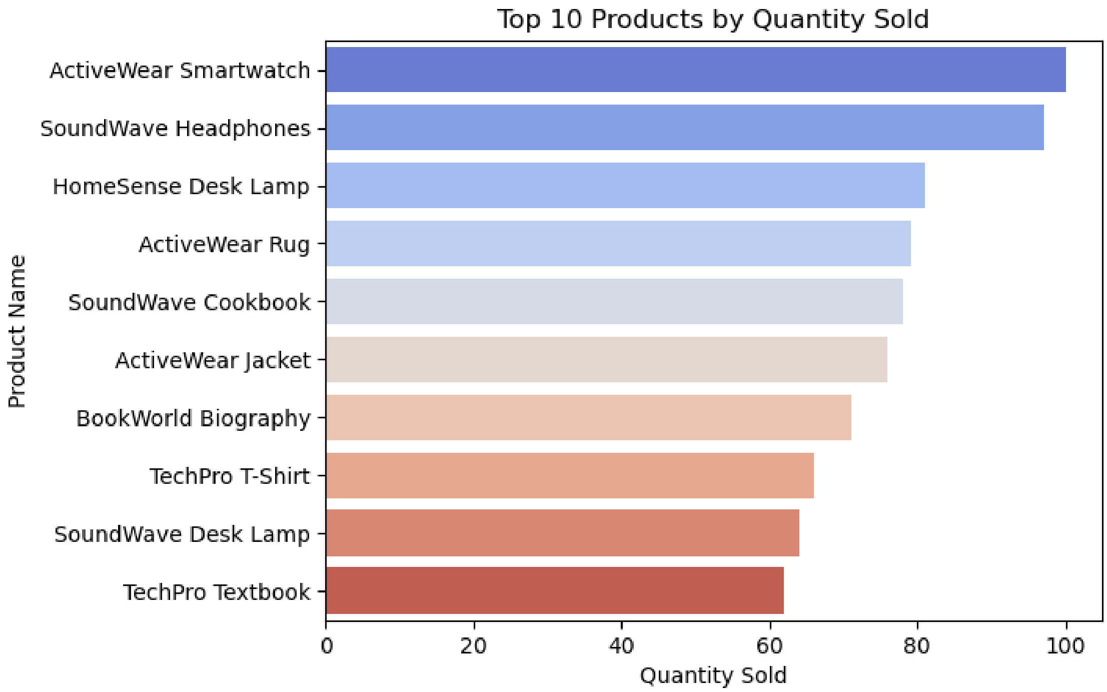
[34]:
```
merged_data['Month'] = merged_data['TransactionDate'].dt.month
monthly_sales = merged_data.groupby('Month')['TotalValue'].sum()
sns.lineplot(x=monthly_sales.index, y=monthly_sales.values, marker='o',␣
  ↪color='blue')
plt.title('Monthly Sales Trends')
plt.xlabel('Month')
plt.ylabel('Total Sales')
plt.grid()
plt.show()
```



[35]:
```
sns.boxplot(x='Category', y='Price', data=products, palette='Set3')
plt.title('Product Price Distribution by Category')
plt.xlabel('Category')
plt.ylabel('Price')
plt.xticks(rotation=45)
plt.show()
```

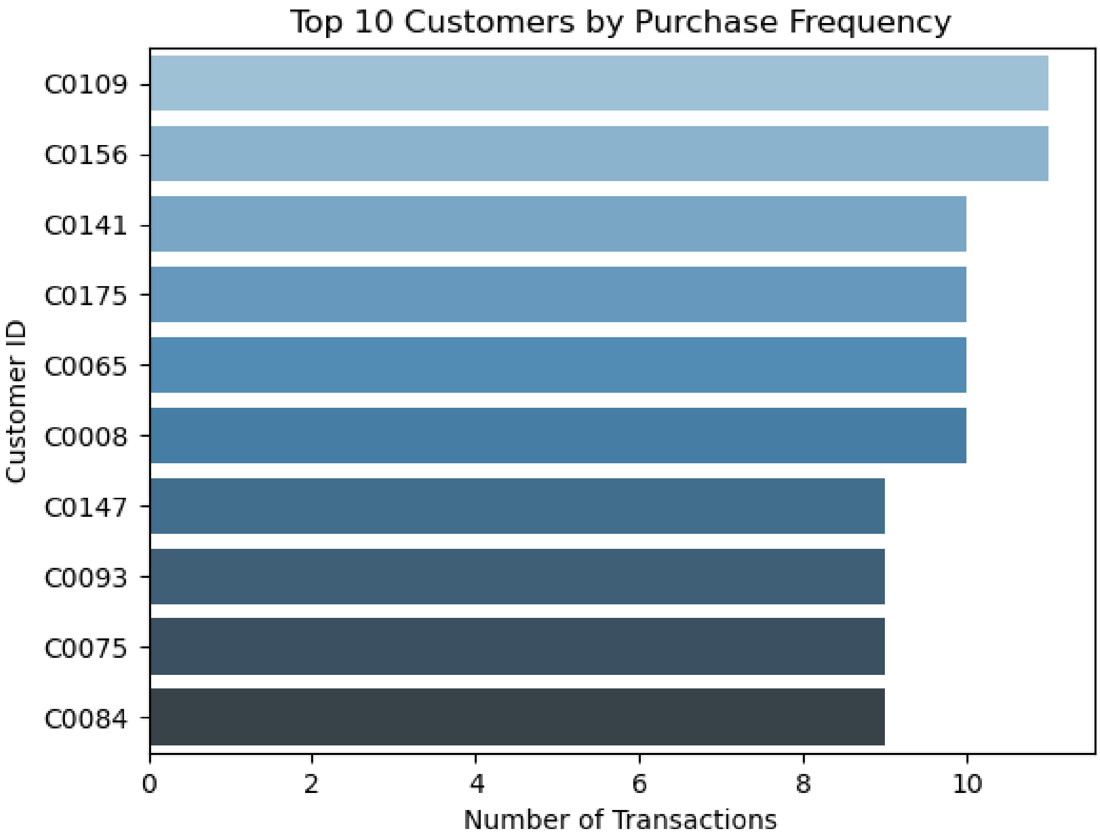


[36]:
```
top_products = merged_data.groupby('ProductName')['Quantity'].sum().
  ↪sort_values(ascending=False).head(10)
sns.barplot(x=top_products.values, y=top_products.index, palette='coolwarm')
plt.title('Top 10 Products by Quantity Sold')
plt.xlabel('Quantity Sold')
plt.ylabel('Product Name')
plt.show()
```



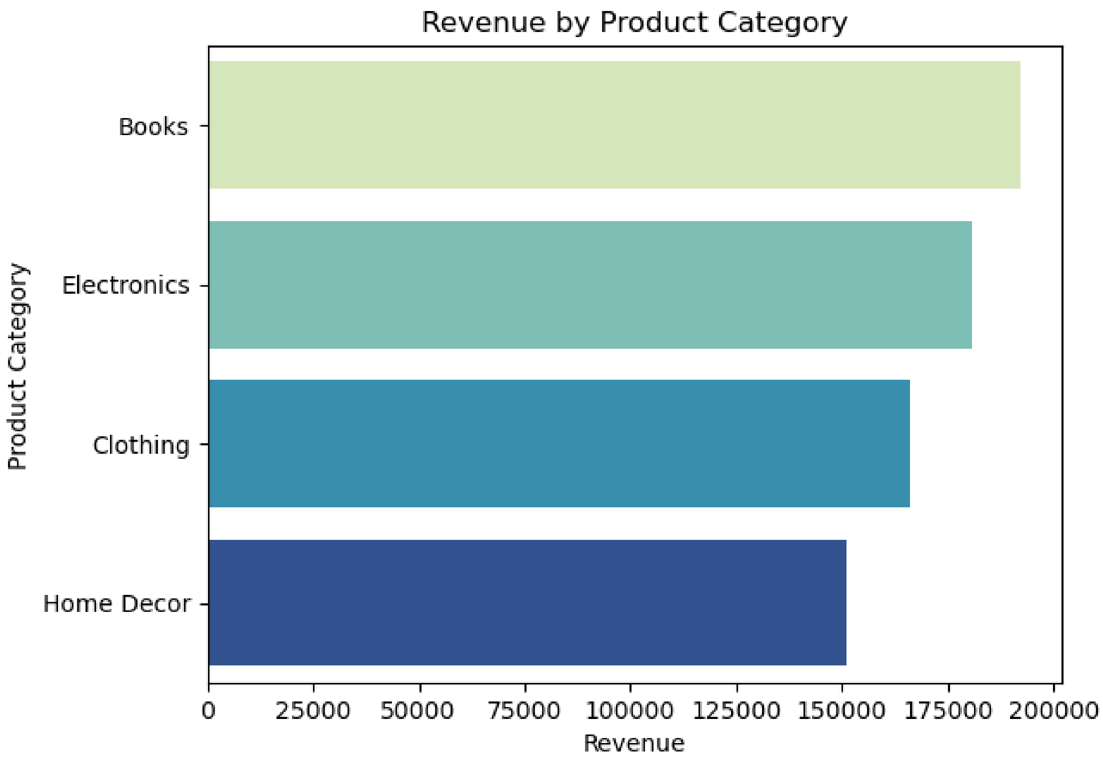
[37]:

```
customer_frequency = transactions['CustomerID'].value_counts().head(10)
sns.barplot(x=customer_frequency.values, y=customer_frequency.index,␣
  ↪palette='Blues_d')
plt.title('Top 10 Customers by Purchase Frequency')
plt.xlabel('Number of Transactions')
plt.ylabel('Customer ID')
plt.show()
```

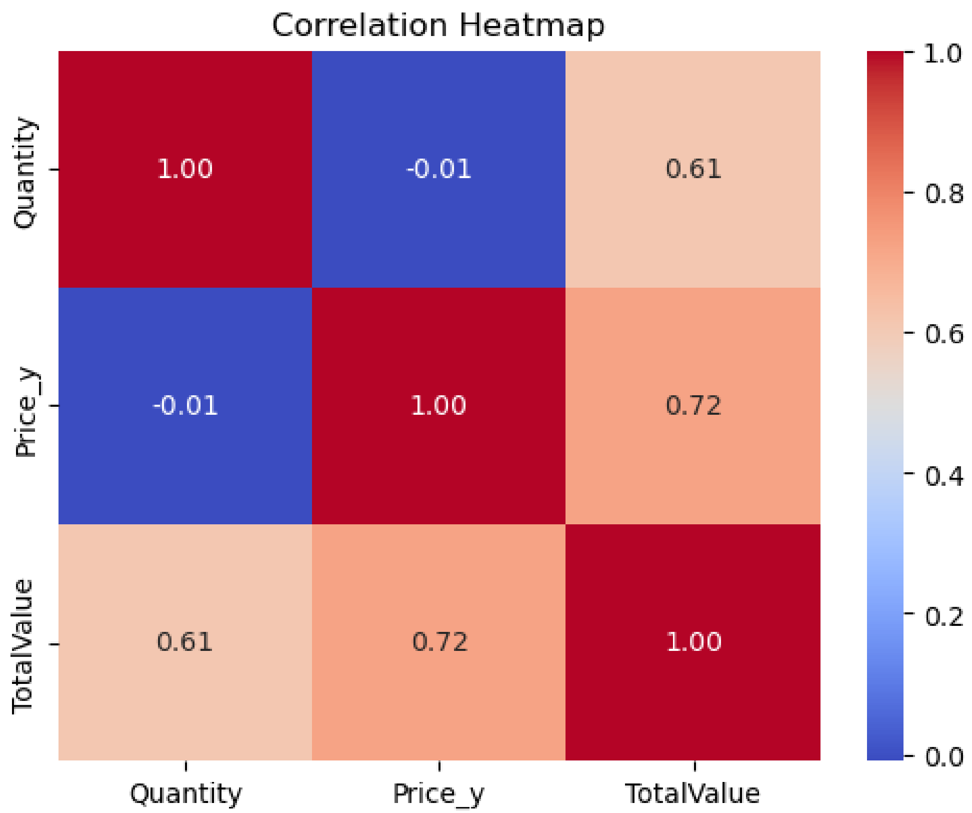


[38]:

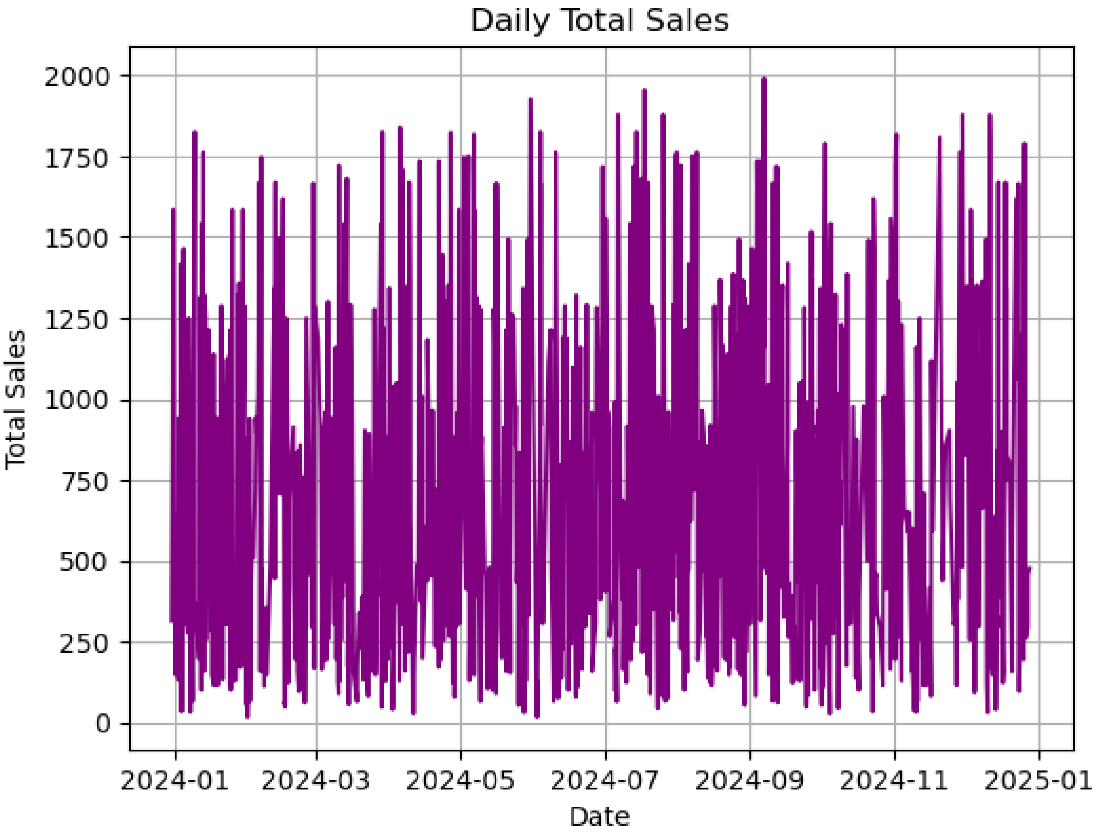
```
category_revenue = merged_data.groupby('Category')['TotalValue'].sum().
  ↪sort_values(ascending=False)
sns.barplot(x=category_revenue.values, y=category_revenue.index,␣
  ↪palette='YlGnBu')
plt.title('Revenue by Product Category')
plt.xlabel('Revenue')
plt.ylabel('Product Category')
plt.show()
```



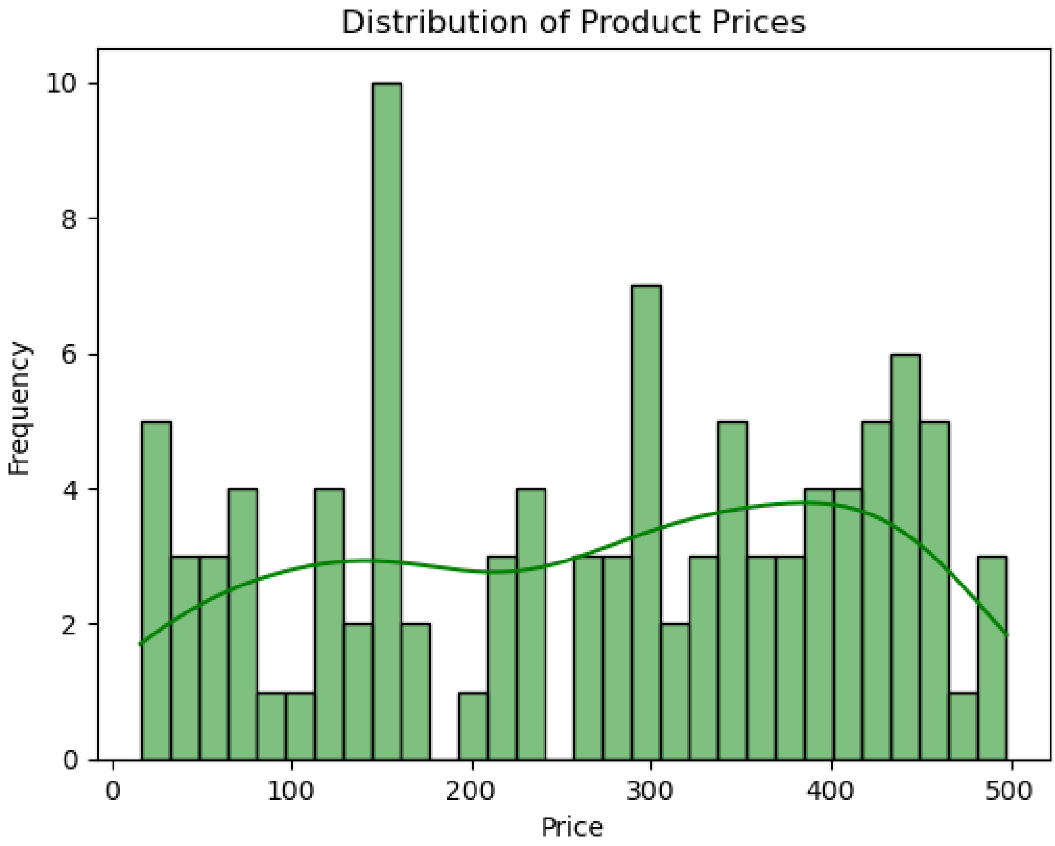
[40]:
```
numeric_data = merged_data[['Quantity', 'Price_y', 'TotalValue']]
corr_matrix = numeric_data.corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.show()
```



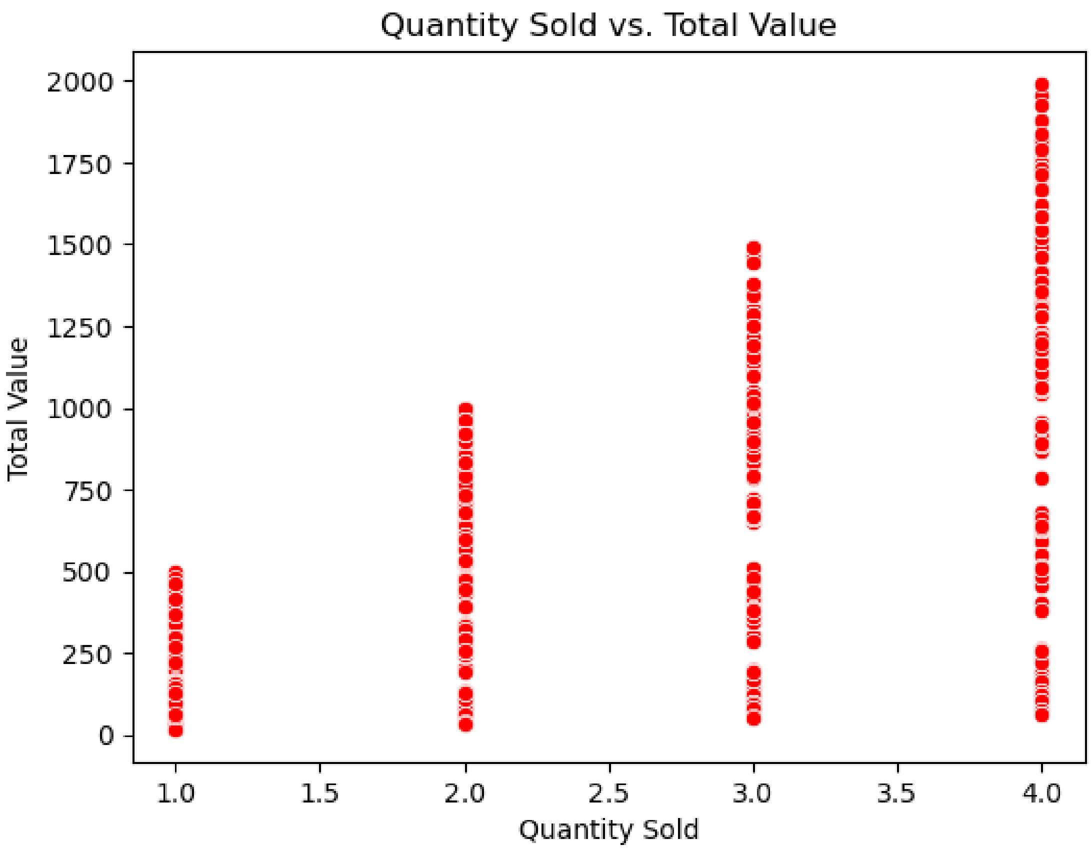
[41]:
```
daily_sales = merged_data.groupby('TransactionDate')['TotalValue'].sum()
sns.lineplot(x=daily_sales.index, y=daily_sales.values, color='purple')
plt.title('Daily Total Sales')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.grid()
plt.show()
```



[57]:
```
sns.histplot(products['Price'], bins=30, kde=True, color='green')
plt.title('Distribution of Product Prices')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.show()
```

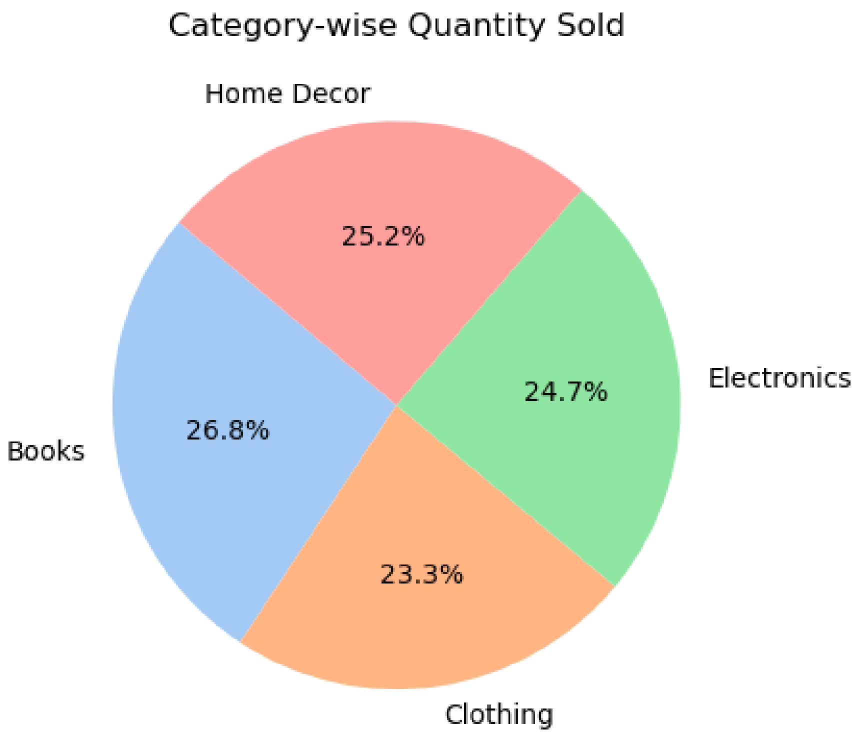


[43]:
```
sns.scatterplot(x='Quantity', y='TotalValue', data=transactions, color='red')
plt.title('Quantity Sold vs. Total Value')
plt.xlabel('Quantity Sold')
plt.ylabel('Total Value')
plt.show()
```



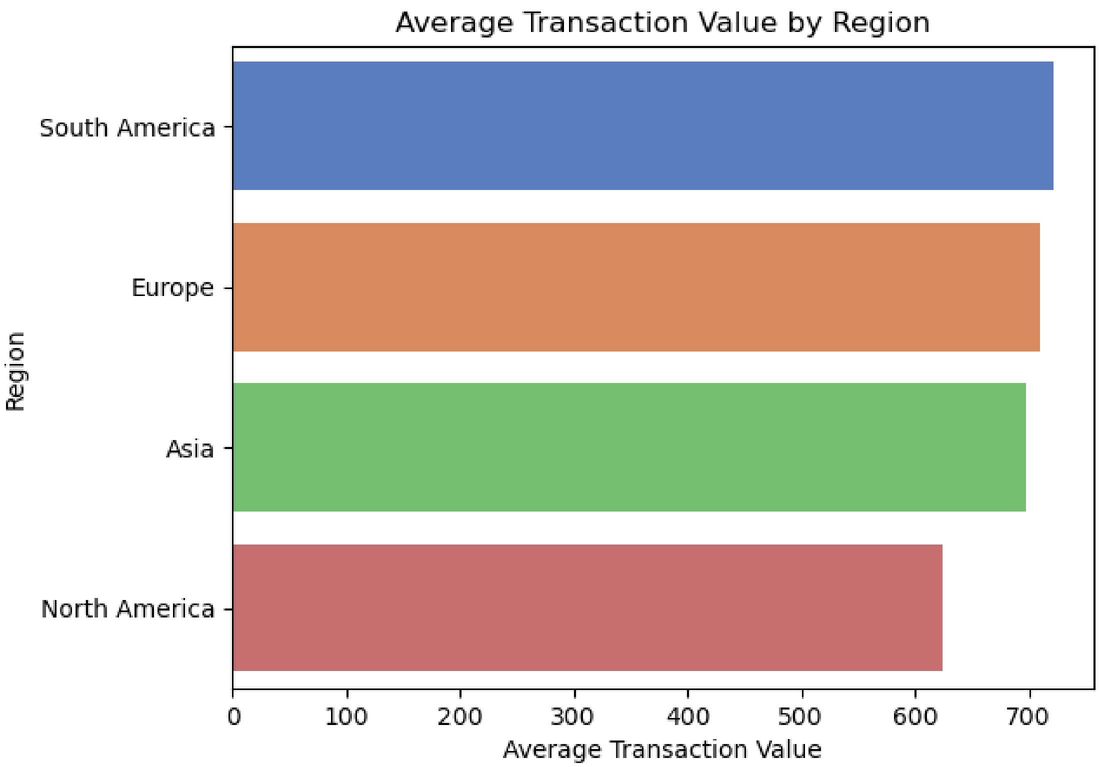
[44]:

```
category_quantity = merged_data.groupby('Category')['Quantity'].sum()
plt.pie(category_quantity, labels=category_quantity.index, autopct='%1.1f%%',␣
  ↪startangle=140, colors=sns.color_palette('pastel'))
plt.title('Category-wise Quantity Sold')
plt.show()
```

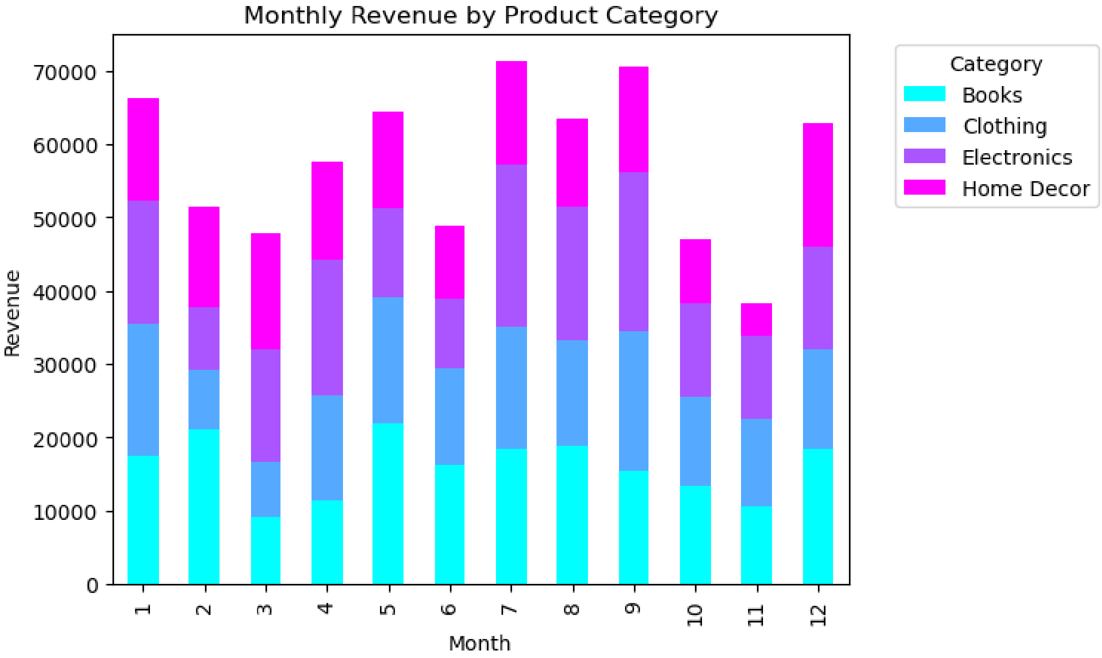


[45]:

```
avg_transaction_value = merged_data.groupby('Region')['TotalValue'].mean().
  ↪sort_values(ascending=False)
sns.barplot(x=avg_transaction_value.values, y=avg_transaction_value.index,␣
  ↪palette='muted')
plt.title('Average Transaction Value by Region')
plt.xlabel('Average Transaction Value')
plt.ylabel('Region')
plt.show()
```



[46]:
```
monthly_category_revenue = merged_data.groupby(['Month',␣
↪'Category'])['TotalValue'].sum().unstack()
monthly_category_revenue.plot(kind='bar', stacked=True, colormap='cool')
plt.title('Monthly Revenue by Product Category')
plt.xlabel('Month')
plt.ylabel('Revenue')
plt.legend(title='Category', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```


Monthly Revenue by Product Category
Revenue
0
10000
20000
30000
40000
50000
60000
70000
Month
1
2
3
4
5
6
7
8
9
10
11
12
Category
Books
Clothing
Electronics
Home Decor

```
[ ]:
```