In [7]:

```
'''import pandas as pd

# Reading the CSV files

productions = pd.read_csv('Products.csv.csv')



# Display the first few rows of each dataset

print(productions.head())

''''
```

```
  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
```

In [10]:

```
# Inspect the first few rows of each dataset
'''print(customers.head())
print(productions.head())
print(transactions.head())

# Check for missing values
print(customers.isnull().sum())
print(productions.isnull().sum())
print(transactions.isnull().sum())'''
```

```
  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
  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
  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 07:38:55         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
CustomerID      0
CustomerName    0
Region          0
SignupDate      0
dtype: int64
ProductID      0
ProductName    0
Category       0
Price          0
dtype: int64
TransactionID      0
CustomerID         0
ProductID          0
TransactionDate    0
Quantity           0
TotalValue         0
Price              0
dtype: int64
```

In [11]:

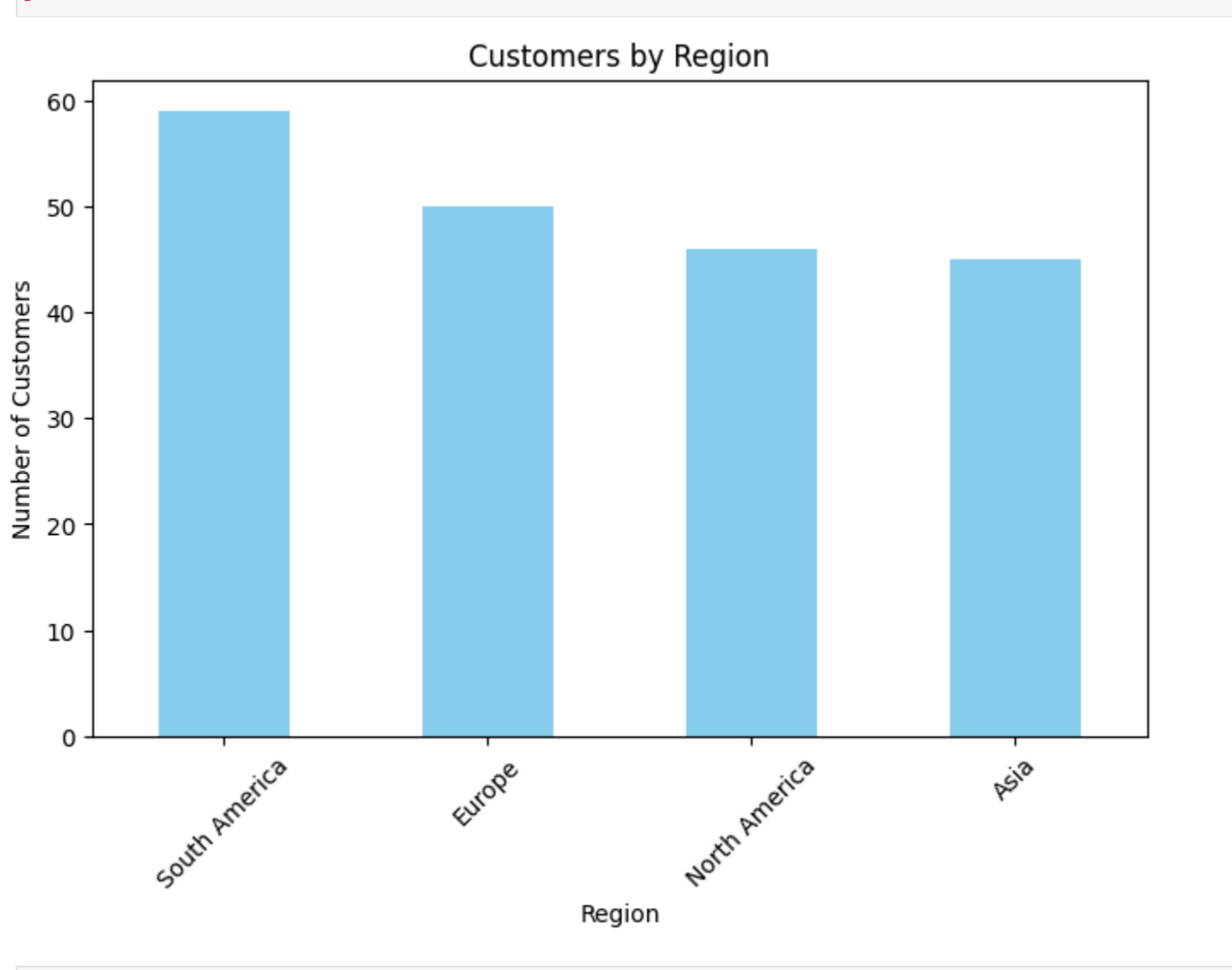
```
# Fill or drop missing values based on your analysis
'''customers.fillna('Unknown', inplace=True)  # Example: filling missing values with 'Unknown'''
```

In [12]:

```
# Drop duplicate rows if any
'''customers.drop_duplicates(inplace=True)'''
```

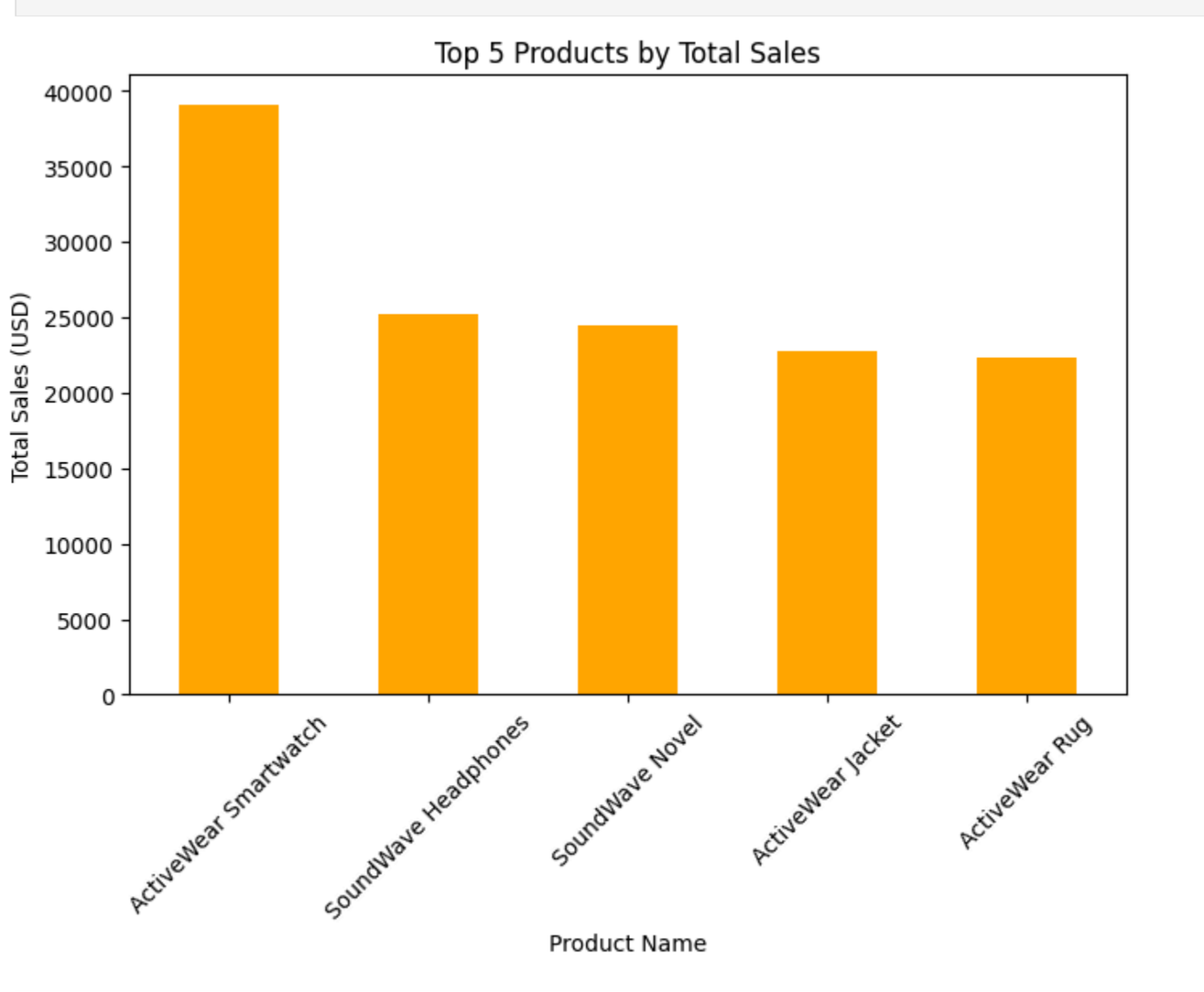
In [13]:

```
''''import matplotlib.pyplot as plt
import seaborn as sns

# Count customers by region
region_counts = customers['Region'].value_counts()

# Bar chart
plt.figure(figsize=(8, 5))
region_counts.plot(kind='bar', color='skyblue')
plt.title("Customers by Region")
plt.xlabel("Region")
plt.ylabel("Number of Customers")
plt.xticks(rotation=45)
plt.show() '''
```



In [15]:

```
# Merge transactions with products
'''merged_data = transactions.merge(productions, on="ProductID")

# Total sales by product
top_products = merged_data.groupby("ProductName")['TotalValue'].sum().nlargest(5)

# Bar chart
top_products.plot(kind='bar', figsize=(8, 5), color='orange')
plt.title("Top 5 Products by Total Sales")
plt.xlabel("Product Name")
plt.ylabel("Total Sales (USD)")
plt.xticks(rotation=45)
plt.show()
'''
```



In [20]:

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

# Load the datasets (update file names if necessary)
customers_df = pd.read_csv('Customers.csv.csv')
products_df = pd.read_csv('Products.csv.csv')
transactions_df = pd.read_csv('Transactions.csv.csv')

# Convert date columns to datetime format
customers_df['SignupDate'] = pd.to_datetime(customers_df['SignupDate'])
transactions_df['TransactionDate'] = pd.to_datetime(transactions_df['TransactionDate'])

# Display summary of Customers dataset
print("Customers Dataset Summary:")
print(customers_df.info())
print(customers_df.describe(include='all'))

# Optionally, display the first few rows to inspect the data
print(customers_df.head())
print(products_df.head())
print(transactions_df.head())
```

```
Customers Dataset Summary:
<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    datetime64[ns]
dtypes: datetime64[ns](1), object(3)
memory usage: 6.4+ KB
None
       CustomerID      CustomerName         Region           SignupDate
count         200               200            200                  200
unique        200               200              4                  NaN
top         C0001  Lawrence Carroll  South America                  NaN
freq            1                 1             59                  NaN
mean          NaN               NaN            NaN  2023-07-19 08:31:12
min           NaN               NaN            NaN  2022-01-22 00:00:00
25%           NaN               NaN            NaN  2022-09-26 12:00:00
50%           NaN               NaN            NaN  2023-08-31 12:00:00
75%           NaN               NaN            NaN  2024-04-12 12:00:00
max           NaN               NaN            NaN  2024-12-28 00:00:00
  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
  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
  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 07:38:55         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
```

In [ ]:

In [21]:

```
# Summary of Products dataset
print("\nProducts Dataset Summary:")
print(products_df.info())
print(products_df.describe(include='all'))
```

```
Products Dataset Summary:
<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
None
       ProductID            ProductName Category       Price
count        100                    100      100  100.000000
unique       100                     66        4         NaN
top         P001  ActiveWear Smartwatch    Books         NaN
freq           1                      4       26         NaN
mean         NaN                    NaN      NaN  267.551700
std          NaN                    NaN      NaN  143.219383
min          NaN                    NaN      NaN   16.080000
25%          NaN                    NaN      NaN  147.767500
50%          NaN                    NaN      NaN  292.875000
75%          NaN                    NaN      NaN  397.090000
max          NaN                    NaN      NaN  497.760000
```

In [ ]:

In [23]:

```
# Load the Transactions dataset
transactions_df = pd.read_csv('Transactions.csv.csv')

# Convert any necessary columns (e.g., dates) to datetime format if required
# transactions_df['YourDateColumn'] = pd.to_datetime(transactions_df['YourDateColumn'])

# Display summary of Transactions dataset
print("\nTransactions Dataset Summary:")
print(transactions_df.info())
print(transactions_df.describe(include='all'))

# Optionally, display the first few rows to inspect the data
print(transactions_df.head())
```

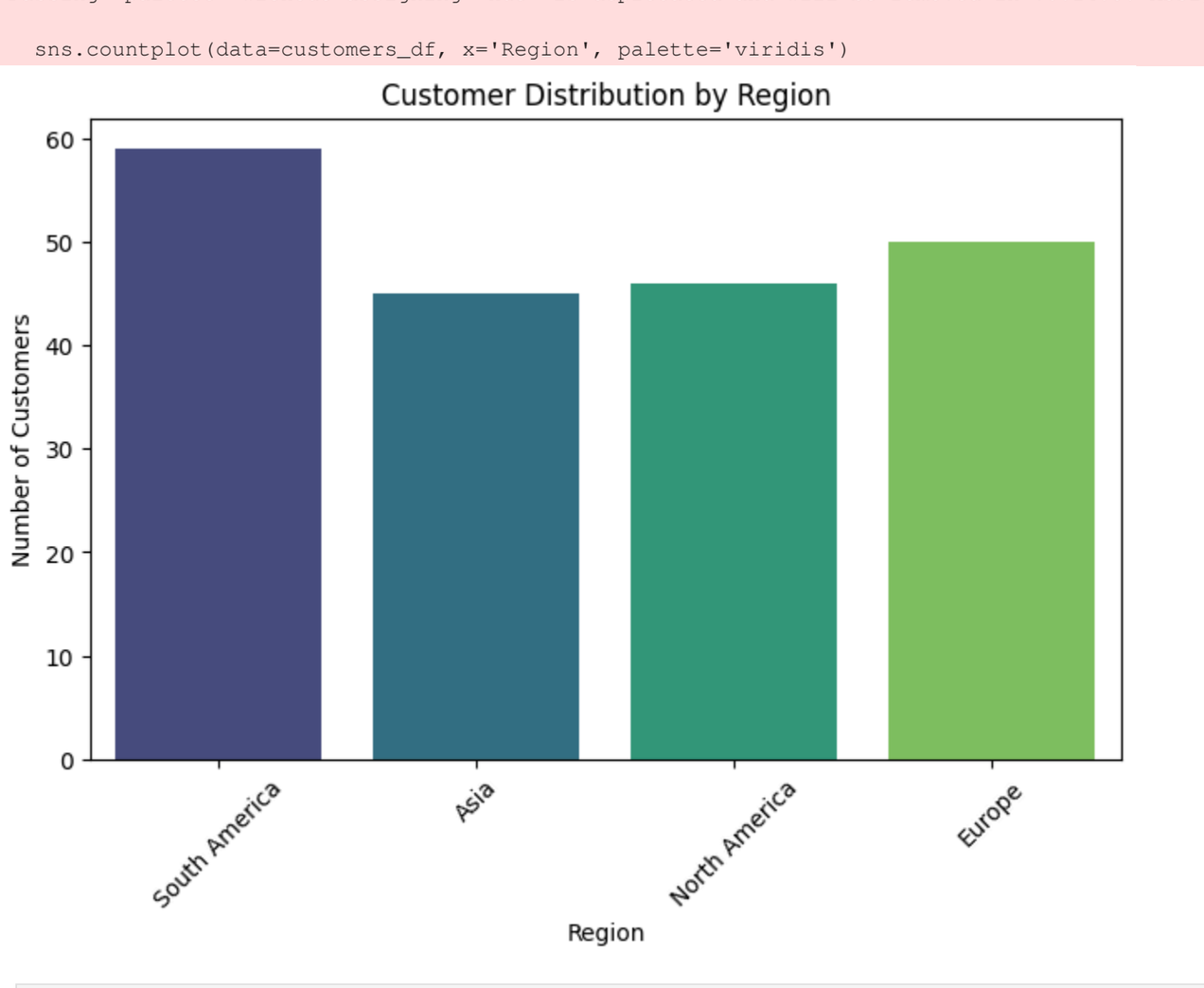
```
Transactions Dataset Summary:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   TransactionID    1000 non-null   object
 1   CustomerID       1000 non-null   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
None
       TransactionID CustomerID ProductID      TransactionDate     Quantity  \
count           1000       1000      1000                 1000  1000.000000
unique          1000        199       100                 1000          NaN
top           T00992      C0109      P059  2024-04-21 10:52:24          NaN
freq               1         11        19                    1          NaN
mean             NaN        NaN       NaN                  NaN     2.537000
std              NaN        NaN       NaN                  NaN     1.117981
min              NaN        NaN       NaN                  NaN     1.000000
25%              NaN        NaN       NaN                  NaN     2.000000
50%              NaN        NaN       NaN                  NaN     3.000000
75%              NaN        NaN       NaN                  NaN     4.000000
max              NaN        NaN       NaN                  NaN     4.000000

         TotalValue       Price
count   1000.000000  1000.00000
unique          NaN         NaN
top             NaN         NaN
freq            NaN         NaN
mean     689.995560   272.55407
std      493.144478   140.73639
min       16.080000    16.08000
25%      295.295000   147.95000
50%      588.880000   299.93000
75%     1011.660000   404.40000
max     1991.040000   497.76000
  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 07:38:55         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
```

In [24]:

```
# EDA - Customers dataset
print("\nCustomer Region Distribution:")
# Display value counts for the 'Region' column
print(customers_df['Region'].value_counts())

# Plot the customer distribution by region using Seaborn's countplot
plt.figure(figsize=(8, 5))
sns.countplot(data=customers_df, x='Region', palette='viridis')
plt.title("Customer Distribution by Region")
plt.ylabel("Number of Customers")
plt.xlabel("Region")
plt.xticks(rotation=45)
plt.show()
```

```
Customer Region Distribution:
Region
South America    59
Europe           50
North America    46
Asia             45
Name: count, dtype: int64
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_13340\2431198478.py:8: 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=False` for the same effect.

  sns.countplot(data=customers_df, x='Region', palette='viridis')
```

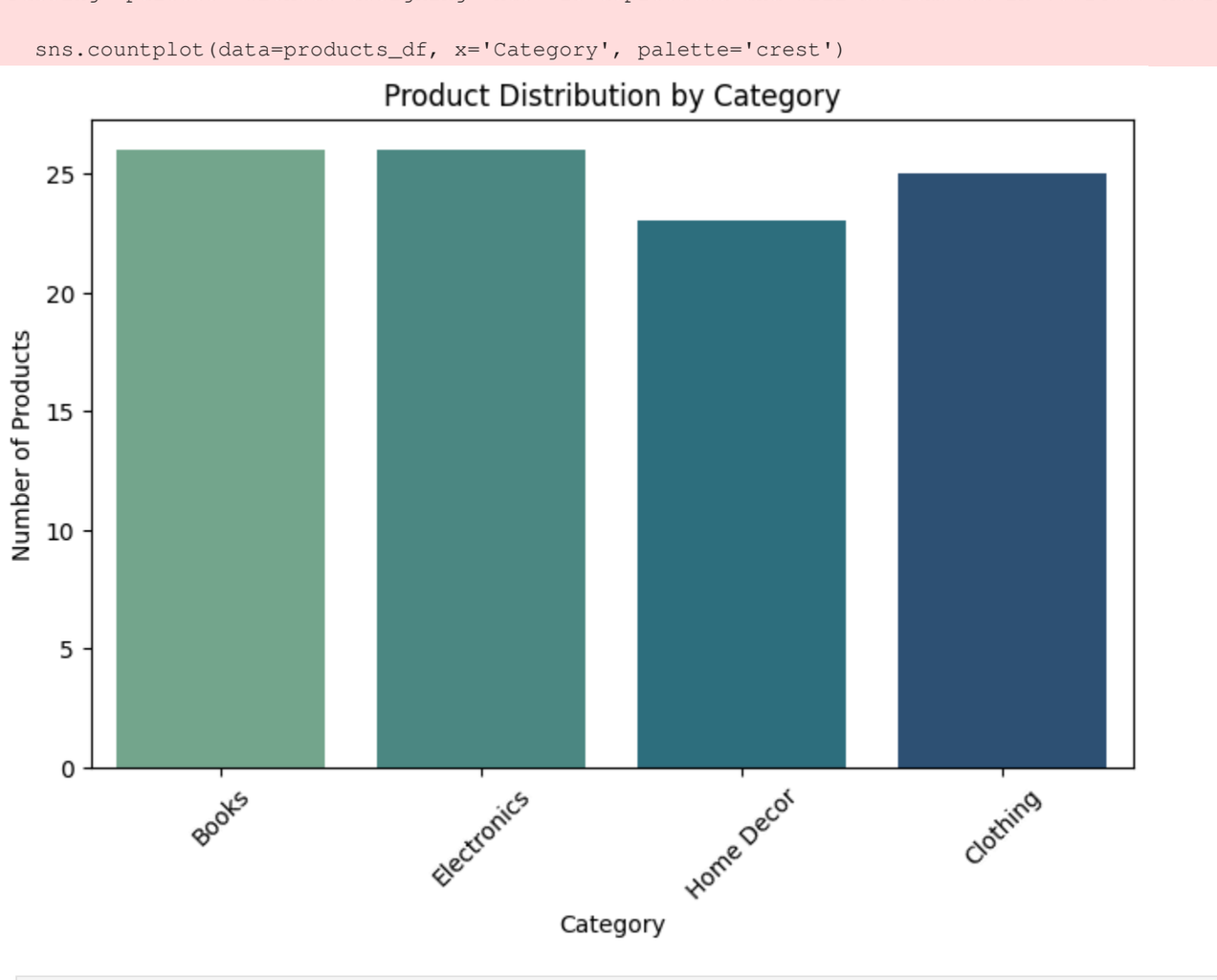


In [25]:

```
# EDA - Products dataset
print("\nProduct Category Distribution:")
# Display value counts for the 'Category' column
print(products_df['Category'].value_counts())

# Plot the product distribution by category using Seaborn's countplot
plt.figure(figsize=(8, 5))
sns.countplot(data=products_df, x='Category', palette='crest')
plt.title("Product Distribution by Category")
plt.ylabel("Number of Products")
plt.xlabel("Category")
plt.xticks(rotation=45)
plt.show()
```

```
Product Category Distribution:
Category
Books          26
Electronics    26
Clothing       25
Home Decor     23
Name: count, dtype: int64
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_13340\2508497260.py:8: 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=False` for the same effect.

  sns.countplot(data=products_df, x='Category', palette='crest')
```



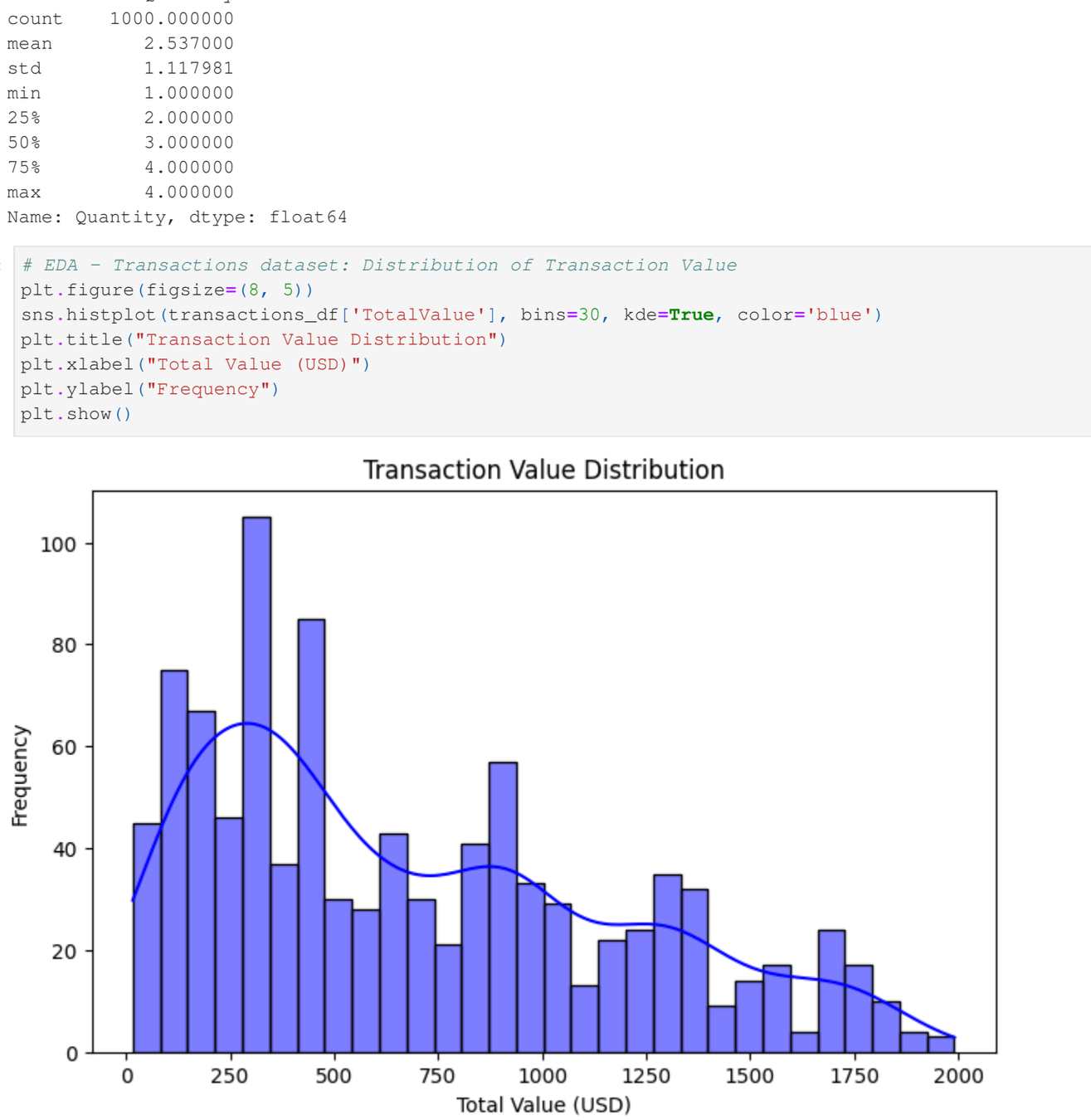
In [26]:

```
# EDA - Transactions dataset
print("\nTransaction Value Statistics:")
# Display statistics for the 'TotalValue' column
print(transactions_df['TotalValue'].describe())

print("\nTransaction Quantity Statistics:")
# Display statistics for the 'Quantity' column
print(transactions_df['Quantity'].describe())
```

```
Transaction Value Statistics:
count    1000.000000
mean      689.995560
std       493.144478
min        16.080000
25%       295.295000
50%       588.880000
75%      1011.660000
max      1991.040000
Name: TotalValue, dtype: float64

Transaction Quantity Statistics:
count    1000.000000
mean        2.537000
std         1.117981
min         1.000000
25%         2.000000
50%         3.000000
75%         4.000000
max         4.000000
Name: Quantity, dtype: float64
```

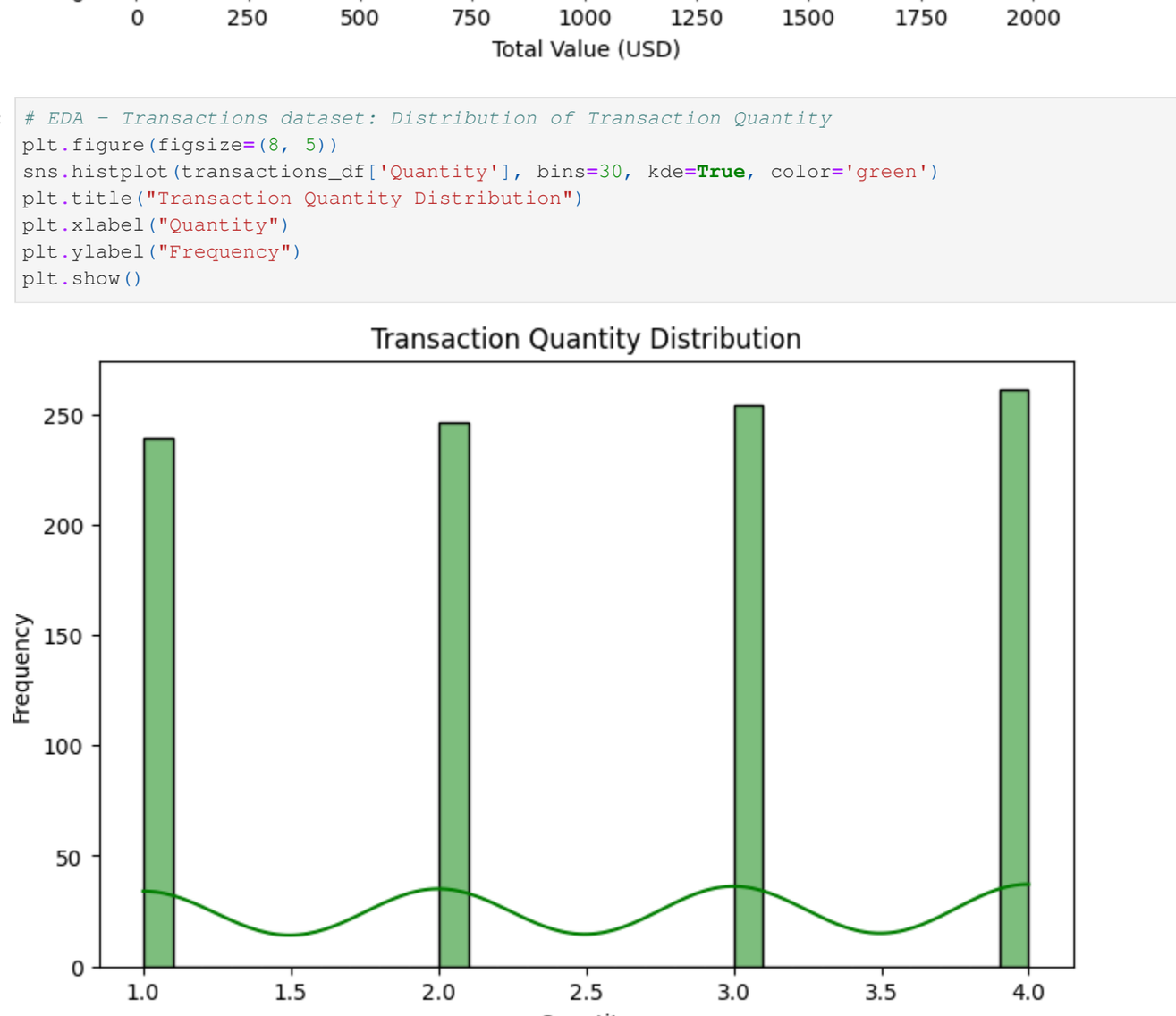
In [27]:

```
# EDA - Transactions dataset: Distribution of Transaction Value
plt.figure(figsize=(8, 5))
sns.histplot(transactions_df['TotalValue'], bins=30, kde=True, color='blue')
plt.title("Transaction Value Distribution")
plt.xlabel("Total Value (USD)")
plt.ylabel("Frequency")
plt.show()
```



In [28]:

```
# EDA - Transactions dataset: Distribution of Transaction Quantity
plt.figure(figsize=(8, 5))
sns.histplot(transactions_df['Quantity'], bins=30, kde=True, color='green')
plt.title("Transaction Quantity Distribution")
plt.xlabel("Quantity")
plt.ylabel("Frequency")
plt.show()
```



In [29]:

```
# Merge datasets for further analysis
merged_df = transactions_df.merge(customers_df, on='CustomerID').merge(products_df, on='ProductID')

# Insights from merged dataset
print("\nMerged Dataset Summary:")
print(merged_df.info())
```

```
Merged Dataset Summary:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 13 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_x          1000 non-null   float64
 7   CustomerName     1000 non-null   object
 8   Region           1000 non-null   object
 9   SignupDate       1000 non-null   datetime64[ns]
 10  ProductName      1000 non-null   object
 11  Category         1000 non-null   object
 12  Price_y          1000 non-null   float64
dtypes: datetime64[ns](1), float64(3), int64(1), object(8)
memory usage: 101.7+ KB
None
```

In [30]:

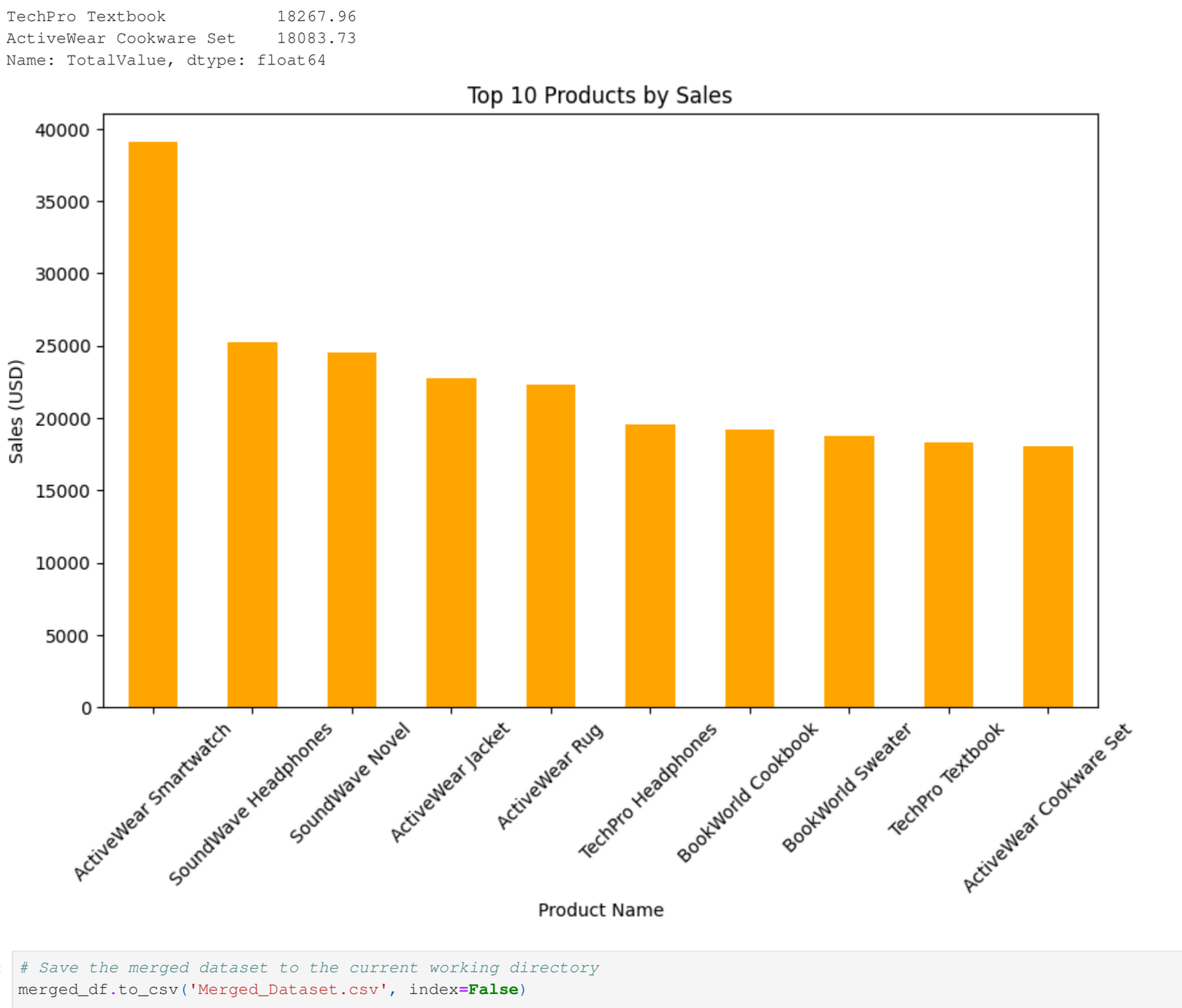
```
# Example Insight: Sales by Region
region_sales = merged_df.groupby('Region')['TotalValue'].sum().sort_values(ascending=False)

# Print the total sales by region
print("\nTotal Sales by Region:")
print(region_sales)

# Visualize the sales by region
plt.figure(figsize=(8, 5))
region_sales.plot(kind='bar', color='teal')
plt.title("Total Sales by Region")
plt.ylabel("Sales (USD)")
plt.xlabel("Region")
plt.xticks(rotation=45)
plt.show()
```

```
Total Sales by Region:
Region
South America    219352.56
Europe           166254.63
North America    152313.40
Asia             152074.97
Name: TotalValue, dtype: float64
```

In [31]:

```
# Example Insight: Top 10 Products by Sales
product_sales = merged_df.groupby('ProductName')['TotalValue'].sum().sort_values(ascending=False).head(10)

# Print the top 10 products by sales
print("\nTop 10 Products by Sales:")
print(product_sales)

# Visualize the top 10 products by sales
plt.figure(figsize=(10, 6))
product_sales.plot(kind='bar', color='orange')
plt.title("Top 10 Products by Sales")
plt.ylabel("Sales (USD)")
plt.xlabel("Product Name")
plt.xticks(rotation=45)
plt.show()
```

```
Top 10 Products by Sales:
ProductName
ActiveWear Smartwatch      39096.97
SoundWave Headphones       25211.64
SoundWave Novel            24507.90
ActiveWear Jacket          22712.56
ActiveWear Rug             22314.43
TechPro Headphones         19513.80
BookWorld Cookbook         19221.99
BookWorld Sweater          18743.79
TechPro Textbook           18267.96
ActiveWear Cookware Set    18083.73
Name: TotalValue, dtype: float64
```



In [35]:

```
# Save the merged dataset to the current working directory
merged_df.to_csv('Merged_Dataset.csv', index=False)
```

In [1]:

```
import os
os.getcwd()
```

```
<div id="1" class="1" data-value="0" data-checked="off" checked_description="
```