# cdoqr0nvn

January 27, 2025

[4]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[5]:

```python
customers = pd.read_csv('Customers.csv')
products = pd.read_csv('Products.csv')
transactions = pd.read_csv('Transactions.csv')
```

[6]:

```python
print("Customers Dataset Info:")
print(customers.info(), "\n")
print("Products Dataset Info:")
print(products.info(), "\n")
print("Transactions Dataset Info:")
print(transactions.info(), "\n")
```

```
Customers Dataset 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
None

Products Dataset 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: 3.2+ KB
None

Transactions Dataset 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
None
```

[7]:

```
print("First few rows of Customers Dataset:")
print(customers.head(), "\n")

print("First few rows of Products Dataset:")
print(products.head(), "\n")

print("First few rows of Transactions Dataset:")
print(transactions.head(), "\n")
```

```
First few rows of Customers Dataset:
  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

First few rows of Products Dataset:
  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
```

```
First few rows of Transactions Dataset:
  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
```

[8]:

```python
print("Missing Values in Customers Dataset:")
print(customers.isnull().sum(), "\n")

print("Missing Values in Products Dataset:")
print(products.isnull().sum(), "\n")

print("Missing Values in Transactions Dataset:")
print(transactions.isnull().sum(), "\n")
```

```
Missing Values in Customers Dataset:
CustomerID      0
CustomerName    0
Region          0
SignupDate      0
dtype: int64

Missing Values in Products Dataset:
ProductID      0
ProductName    0
Category       0
Price          0
dtype: int64

Missing Values in Transactions Dataset:
TransactionID      0
CustomerID         0
ProductID          0
TransactionDate    0
Quantity           0
TotalValue         0
Price              0
```

```
dtype: int64
```

[9]:

```
customers['SignupDate'] = pd.to_datetime(customers['SignupDate'])
transactions['TransactionDate'] = pd.
 ↪to_datetime(transactions['TransactionDate'])
```

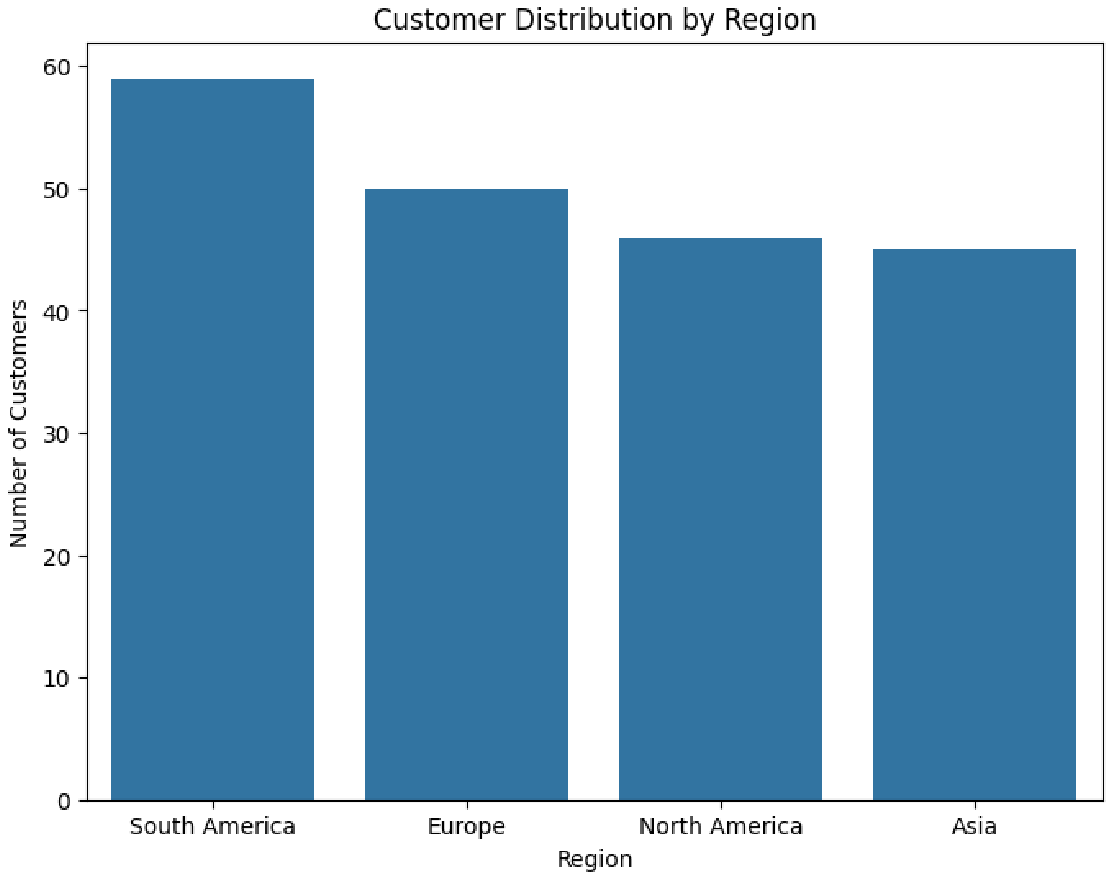
[10]:

```
print("Descriptive statistics for Customers Dataset:")
print(customers.describe(include='all'), "\n")

print("Descriptive statistics for Products Dataset:")
print(products.describe(include='all'), "\n")

print("Descriptive statistics for Transactions Dataset:")
print(transactions.describe(), "\n")
```

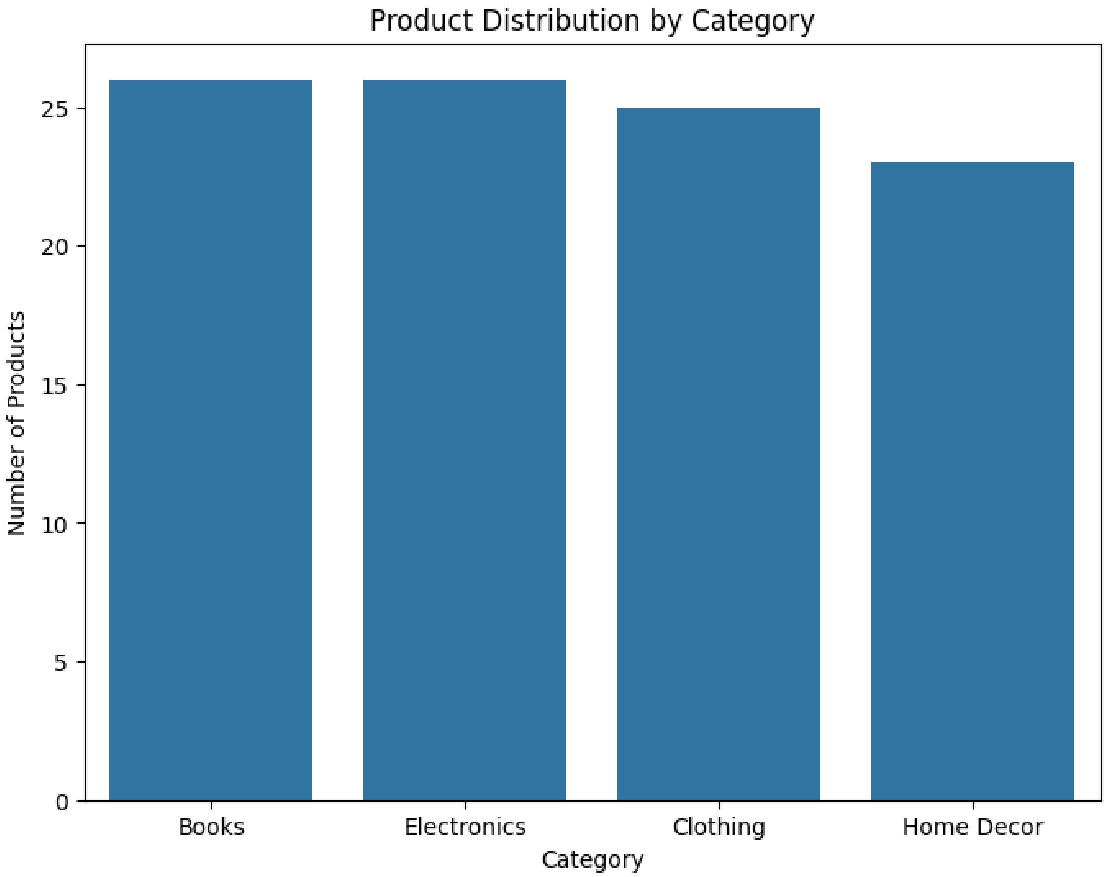
```
Descriptive statistics for Customers Dataset:
       CustomerID      CustomerName         Region           SignupDate
count         200               200            200                  200
unique        200               200              4                  NaN
top         C0001  Lawrence Carroll  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

Descriptive statistics for Products Dataset:
       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

Descriptive statistics for Transactions Dataset:
                     TransactionDate     Quantity   TotalValue       Price
count                           1000  1000.000000  1000.000000  1000.00000
mean   2024-06-23 15:33:02.768999936     2.537000   689.995560   272.55407
min              2023-12-30 15:29:12     1.000000    16.080000    16.08000
```

```
25%         2024-03-25 22:05:34.500000     2.000000    295.295000    147.95000
50%         2024-06-26 17:21:52.500000     3.000000    588.880000    299.93000
75%                2024-09-19 14:19:57     4.000000   1011.660000    404.40000
max                2024-12-28 11:00:00     4.000000   1991.040000    497.76000
std                                NaN     1.117981    493.144478    140.73639
```

[11]:
```
plt.figure(figsize=(8, 6))
sns.countplot(data=customers, x='Region', order=customers['Region'].
  ↪value_counts().index)
plt.title('Customer Distribution by Region')
plt.xlabel('Region')
plt.ylabel('Number of Customers')
plt.show()
```

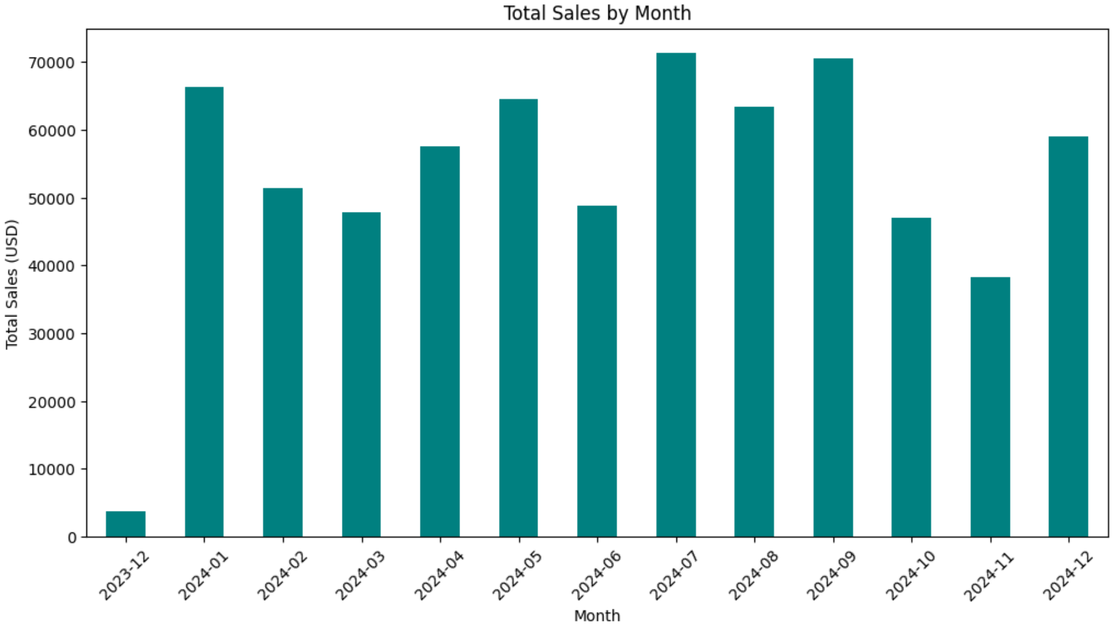


[12]:
```
plt.figure(figsize=(8, 6))
sns.countplot(data=products, x='Category', order=products['Category'].
  ↪value_counts().index)
plt.title('Product Distribution by Category')
plt.xlabel('Category')
plt.ylabel('Number of Products')
plt.show()
```



[13]:
```
transactions['Month'] = transactions['TransactionDate'].dt.to_period('M')
monthly_sales = transactions.groupby('Month')['TotalValue'].sum()
```

[14]:
```
plt.figure(figsize=(12, 6))
monthly_sales.plot(kind='bar', color='teal')
plt.title('Total Sales by Month')
plt.xlabel('Month')
plt.ylabel('Total Sales (USD)')
plt.xticks(rotation=45)
plt.show()
```

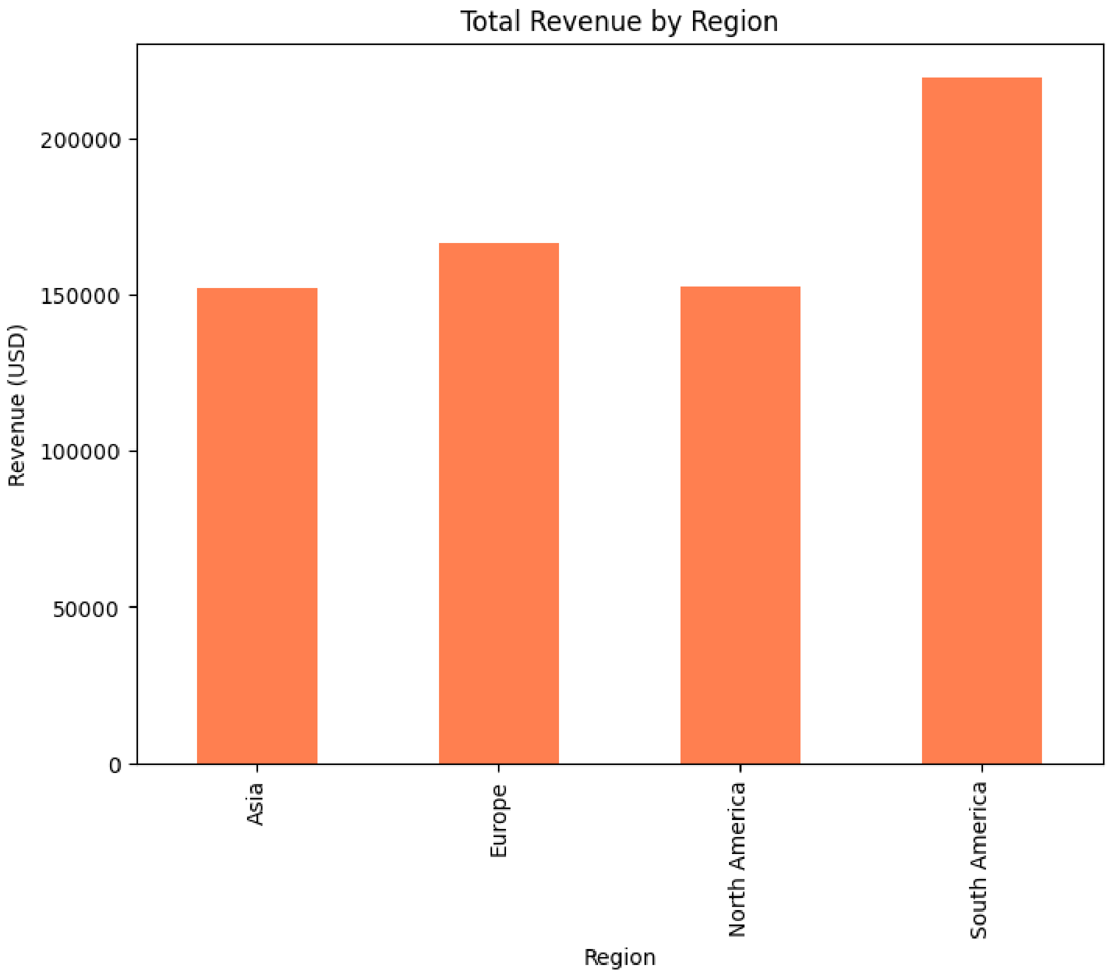


[15]:

```
region_sales = transactions.merge(customers, on='CustomerID').
  ↪groupby('Region')['TotalValue'].sum()
```

[16]:

```
plt.figure(figsize=(8, 6))
region_sales.plot(kind='bar', color='coral')
plt.title('Total Revenue by Region')
plt.xlabel('Region')
plt.ylabel('Revenue (USD)')
plt.show()
```



[17]:

```
customer_region_counts = customers['Region'].value_counts()
top_region = customer_region_counts.idxmax()
top_region_count = customer_region_counts.max()
print(f"Insight 1: The region with the highest number of customers is␣
  ↪{top_region} with {top_region_count} customers.")
```

```
Insight 1: The region with the highest number of customers is South America with
59 customers.
```

[18]:

```
top_product_quantity = transactions.groupby('ProductID')['Quantity'].sum().
  ↪sort_values(ascending=False).head(1)
top_product_id = top_product_quantity.index[0]
top_product_name = products[products['ProductID'] ==␣
  ↪top_product_id]['ProductName'].values[0]
top_product_quantity_value = top_product_quantity.values[0]
print(f"Insight 2: The top-selling product by quantity is '{top_product_name}'␣
  ↪with {top_product_quantity_value} units sold.")
```

```
Insight 2: The top-selling product by quantity is 'SoundWave Jeans' with 46
units sold.
```

[19]:

```
top_month = monthly_sales.idxmax()
top_month_sales = monthly_sales.max()
print(f"Insight 3: The month with the highest total sales is {top_month} with␣
  ↪${top_month_sales:.2f} in revenue.")
```

```
Insight 3: The month with the highest total sales is 2024-07 with $71366.39 in
revenue.
```

[20]:

```
customer_revenue = transactions.groupby('CustomerID')['TotalValue'].sum()
top_10_customers_revenue = customer_revenue.sort_values(ascending=False).
  ↪head(10).sum()
total_revenue = transactions['TotalValue'].sum()
top_10_percentage = (top_10_customers_revenue / total_revenue) * 100
print(f"Insight 4: The top 10 customers contribute {top_10_percentage:.2f}% of␣
  ↪the total revenue.")
```

```
Insight 4: The top 10 customers contribute 11.02% of the total revenue.
```

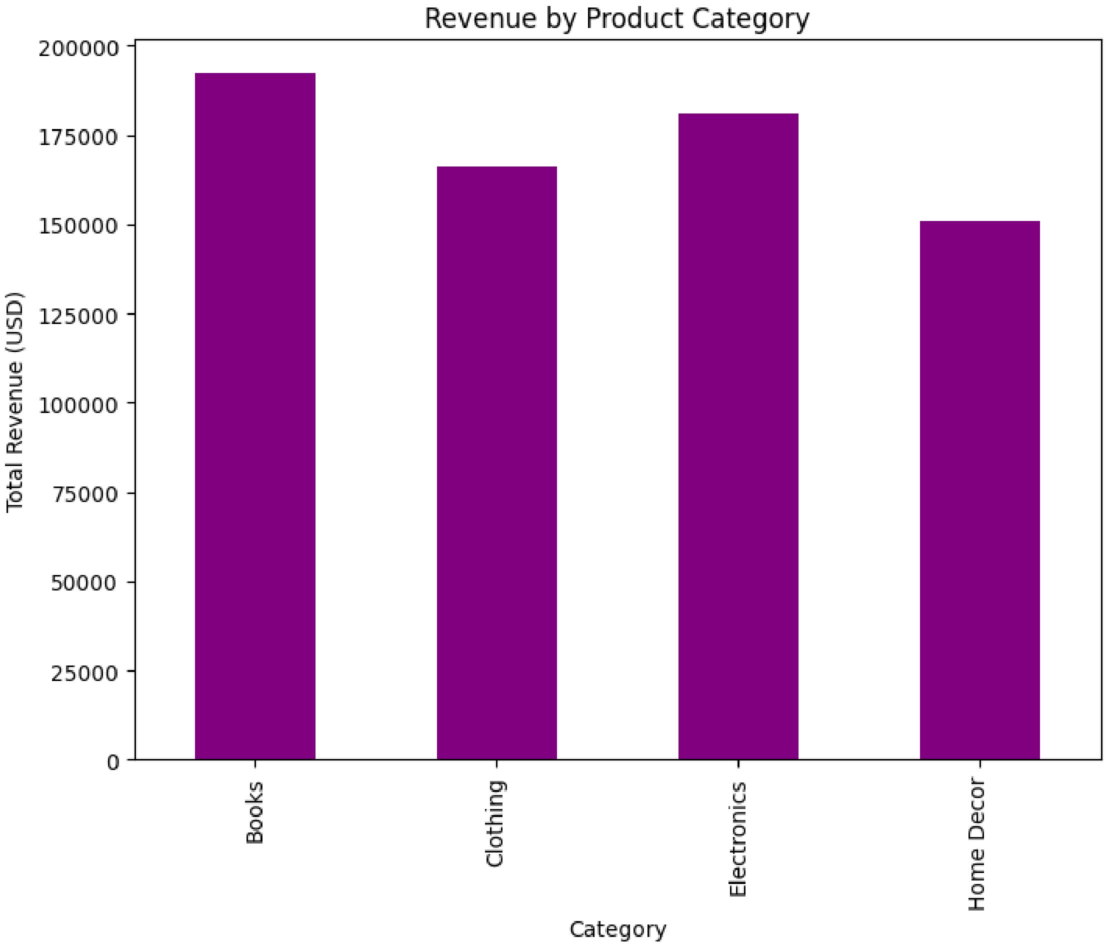
[21]:

```
category_sales = transactions.merge(products, on='ProductID').
  ↪groupby('Category')['TotalValue'].sum()
category_quantity = transactions.merge(products, on='ProductID').
  ↪groupby('Category')['Quantity'].sum()
average_purchase_value = category_sales / category_quantity
print(f"Insight 5: Average purchase value by category:
  ↪\n{average_purchase_value}\n")
```

```
Insight 5: Average purchase value by category:
Category
Books          282.154875
Clothing       281.645186
Electronics    288.330941
Home Decor     236.140736
dtype: float64
```

[22]:

```
plt.figure(figsize=(8, 6))
category_sales.plot(kind='bar', color='purple')
plt.title('Revenue by Product Category')
plt.xlabel('Category')
plt.ylabel('Total Revenue (USD)')
plt.show()
```


Revenue by Product Category
200000
175000
150000
125000
100000
75000
50000
25000
0
Total Revenue (USD)
Books
Clothing
Electronics
Home Decor
Category

[ ]: