```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import davies_bouldin_score
from sklearn.neighbors import NearestNeighbors
```

```
# Load Data
customers = pd.read_csv('/content/Customers.csv')
products = pd.read_csv('/content/Products.csv')
transactions = pd.read_csv('/content/Transactions.csv')
```

```
customers.shape
```

```
(200, 4)
```

```
products.shape
```

```
(100, 4)
```

```
transactions.shape
```

```
(1000, 7)
```

```
customers.head()
```

	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

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

```
products.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

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

```
transactions.head()
```

```
  TransactionID CustomerID ProductID     TransactionDate  Quantity  TotalValue   Price
0        T00001      C0199      P067 2024-08-25 12:38:23         1      300.68  300.68
1        T00112      C0146      P067 2024-05-27 22:23:54         1      300.68  300.68
2        T00166      C0127      P067 2024-04-25 07:38:55         1      300.68  300.68
3        T00272      C0087      P067 2024-03-26 22:55:37         2      601.36  300.68
4        T00363      C0070      P067 2024-03-21 15:10:10         3      902.04  300.68
```

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

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

```
              0
CustomerID    0
CustomerName  0
Region        0
SignupDate    0

dtype: int64
```

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

```
             0
ProductID    0
ProductName  0
Category     0
Price        0

dtype: int64
```

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

	0
TransactionID	0
CustomerID	0
ProductID	0
TransactionDate	0
Quantity	0
TotalValue	0
Price	0

dtype: int64

```
# Task 1: Exploratory Data Analysis (EDA)
# Merge datasets
data = transactions.merge(customers, on='CustomerID').merge(products, on='ProductID')
```

```
# Basic statistics
print(data.describe())
print(data.info())
```

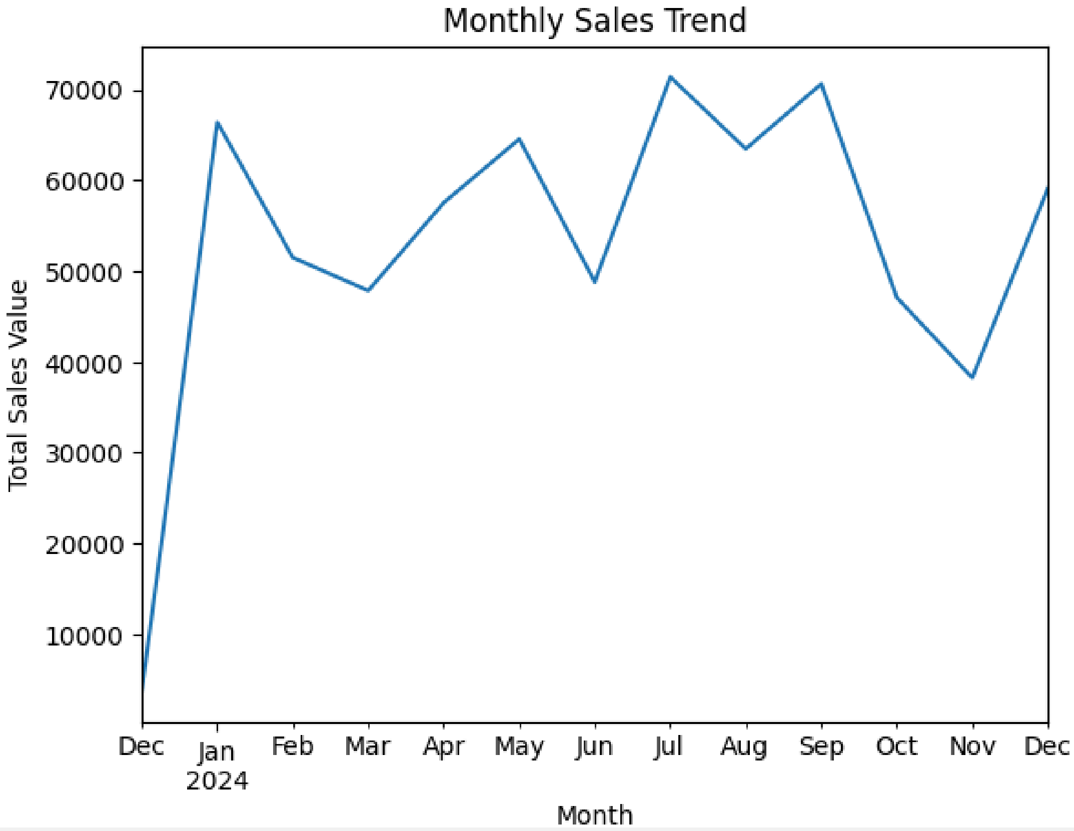
```
          Quantity   TotalValue     Price_x     Price_y
count  1000.000000  1000.000000  1000.00000  1000.00000
mean      2.537000   689.995560   272.55407   272.55407
std       1.117981   493.144478   140.73639   140.73639
min       1.000000    16.080000    16.08000    16.08000
25%       2.000000   295.295000   147.95000   147.95000
50%       3.000000   588.880000   299.93000   299.93000
75%       4.000000  1011.660000   404.40000   404.40000
```

```
max       4.000000  1991.040000   497.76000   497.76000
<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   object 
 10  ProductName      1000 non-null   object 
 11  Category         1000 non-null   object 
 12  Price_y          1000 non-null   float64
dtypes: float64(3), int64(1), object(9)
memory usage: 101.7+ KB
None
```

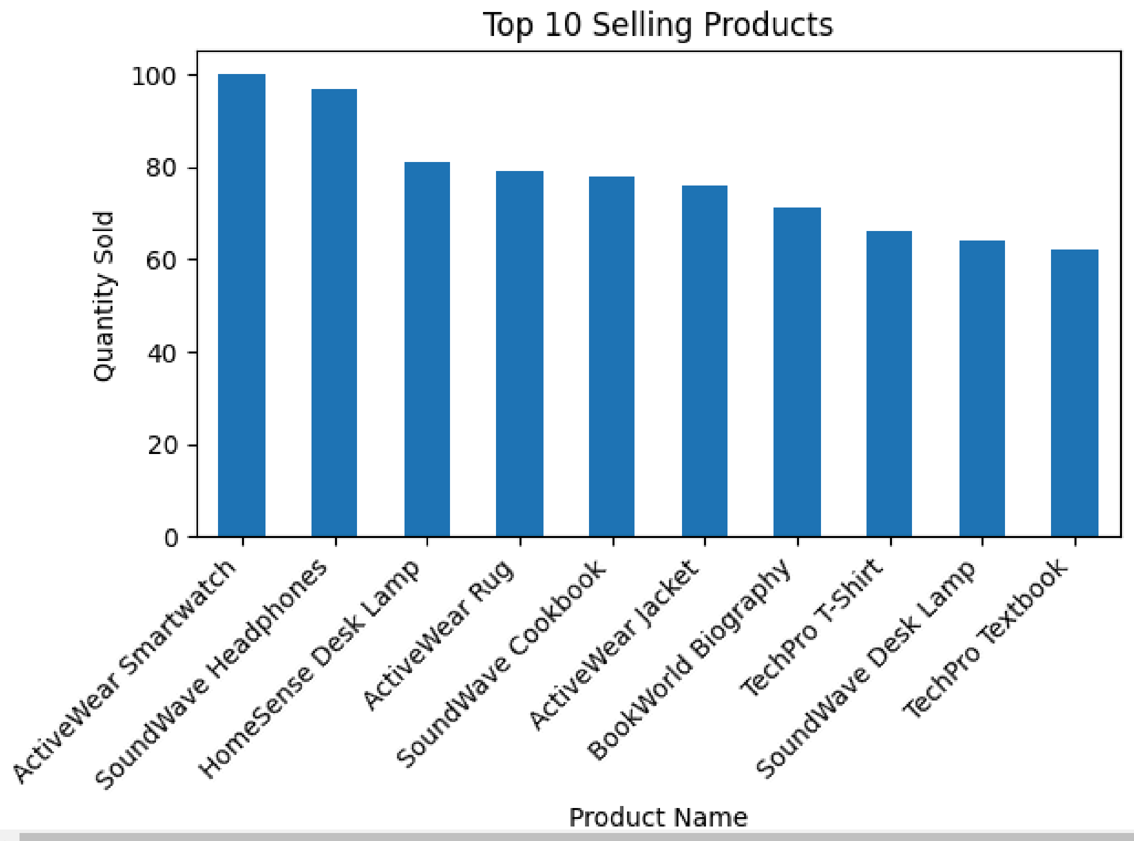
```
# Check for missing values
print(data.isnull().sum())
```

```
TransactionID      0
CustomerID         0
ProductID          0
TransactionDate    0
Quantity           0
TotalValue         0
Price_x            0
CustomerName       0
Region             0
SignupDate         0
ProductName        0
Category           0
Price_y            0
dtype: int64
```

```
import matplotlib.pyplot as plt

# Ensure TransactionDate is in datetime format
data['TransactionDate'] = pd.to_datetime(data['TransactionDate'])

# Group by month and sum 'TotalValue'
monthly_sales = data.groupby(data['TransactionDate'].dt.to_period('M'))['TotalValue'].sum()

# Plot the monthly sales trend
monthly_sales.plot(title='Monthly Sales Trend')
plt.xlabel('Month')
plt.ylabel('Total Sales Value')
plt.show()
```



```
# Group by ProductName and sum only the 'Quantity' column
top_products = data.groupby('ProductName')['Quantity'].sum().sort_values(ascending=False).head(10)

# Plot the top-selling products
top_products.plot(kind='bar', title='Top 10 Selling Products')
plt.xlabel('Product Name')
plt.ylabel('Quantity Sold')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



Business insights

1. Peak sales periods
2. Most popular products
3. Customer distribution by region
4. Average transaction value per region

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.