

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.preprocessing import StandardScaler, OneHotEncoder
```

```
customers = pd.read_csv("Customers.csv")
print(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

```
products = pd.read_csv("Products.csv")
```

```
print(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

```
transactions = pd.read_csv("Transactions.csv")
```

```
print(transactions.head())
```

	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

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

```
print(customers.isnull().sum())
print(products.isnull().sum())
print(transactions.isnull().sum())
```

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	

```
print(customers.duplicated().sum())
print(products.duplicated().sum())
print(transactions.duplicated().sum())
```

0  
0  
0

```
merged_data = transactions.merge(customers, on="CustomerID").merge(products, on="ProductID")
print(merged_data.head())
```

	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_x	CustomerName	Region	SignupDate	\
0	300.68	300.68	Andrea Jenkins	Europe	2022-12-03	
1	300.68	300.68	Brittany Harvey	Asia	2024-09-04	
2	300.68	300.68	Kathryn Stevens	Europe	2024-04-04	
3	601.36	300.68	Travis Campbell	South America	2024-04-11	
4	902.04	300.68	Timothy Perez	Europe	2022-03-15	

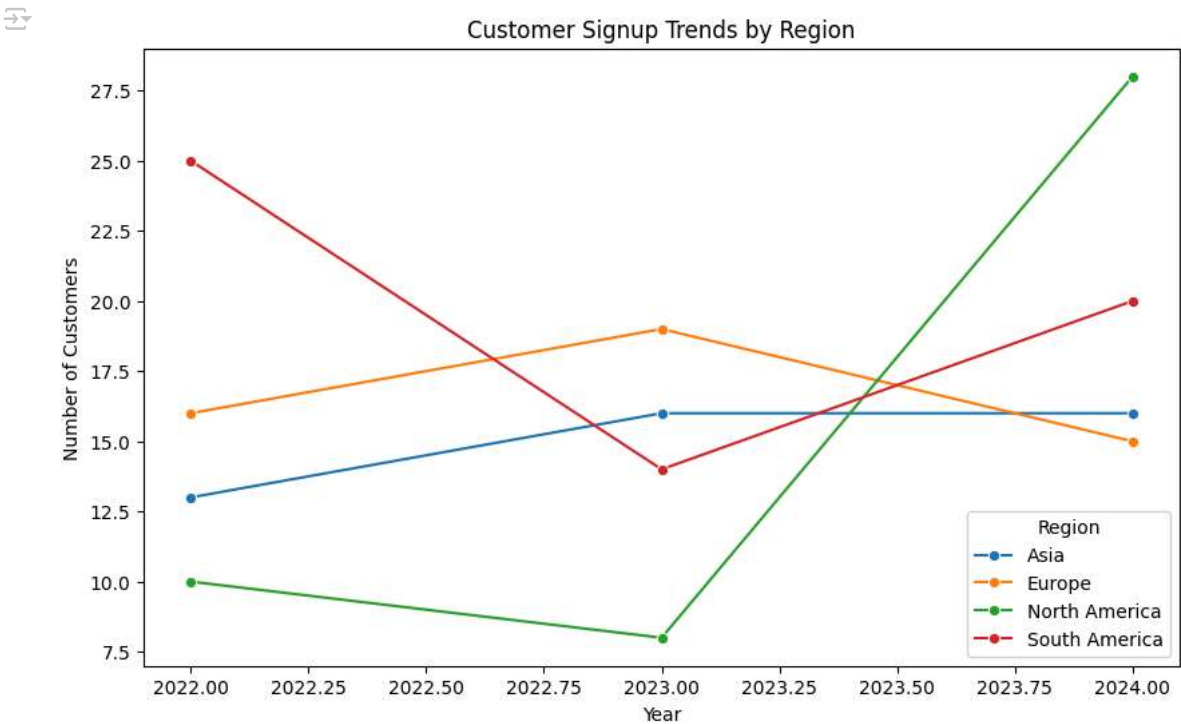
  

	ProductName	Category	Price_y
0	ComfortLiving Bluetooth Speaker	Electronics	300.68
1	ComfortLiving Bluetooth Speaker	Electronics	300.68
2	ComfortLiving Bluetooth Speaker	Electronics	300.68
3	ComfortLiving Bluetooth Speaker	Electronics	300.68
4	ComfortLiving Bluetooth Speaker	Electronics	300.68

Customer Growth by Region

```
# Group by Signup Year and Region
customers['SignupYear'] = customers['SignupDate'].dt.year
signup_trends = customers.groupby(['SignupYear', 'Region']).size().reset_index(name='CustomerCount')

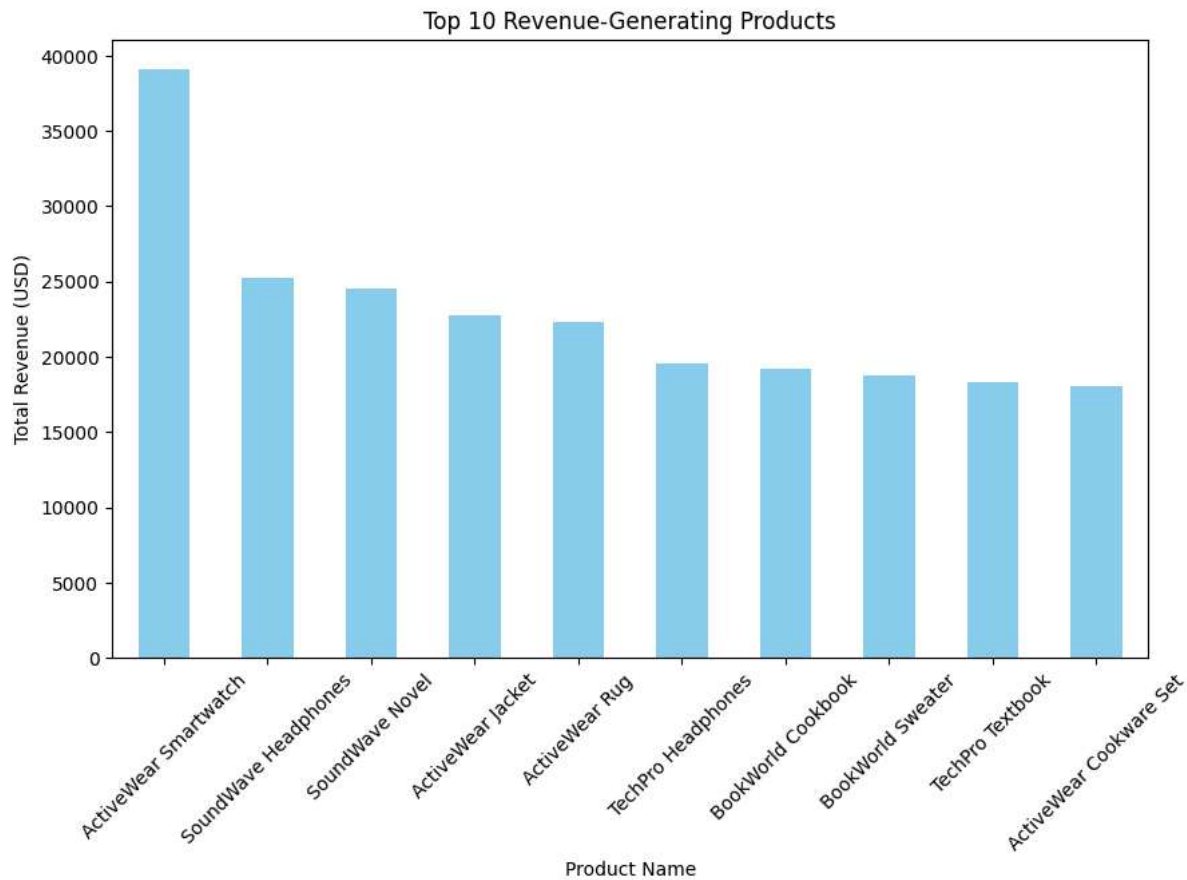
# Plot trends
plt.figure(figsize=(10, 6))
sns.lineplot(data=signup_trends, x='SignupYear', y='CustomerCount', hue='Region', marker='o')
plt.title('Customer Signup Trends by Region')
plt.xlabel('Year')
plt.ylabel('Number of Customers')
plt.legend(title='Region')
plt.show()
```



Top Revenue-Generating Products

```
# Total revenue per product
product_revenue = merged_data.groupby('ProductName')['TotalValue'].sum().sort_values(ascending=False).head(10)

# Plot top 10 products by revenue
plt.figure(figsize=(10, 6))
product_revenue.plot(kind='bar', color='skyblue')
plt.title('Top 10 Revenue-Generating Products')
plt.xlabel('Product Name')
plt.ylabel('Total Revenue (USD)')
plt.xticks(rotation=45)
plt.show()
```

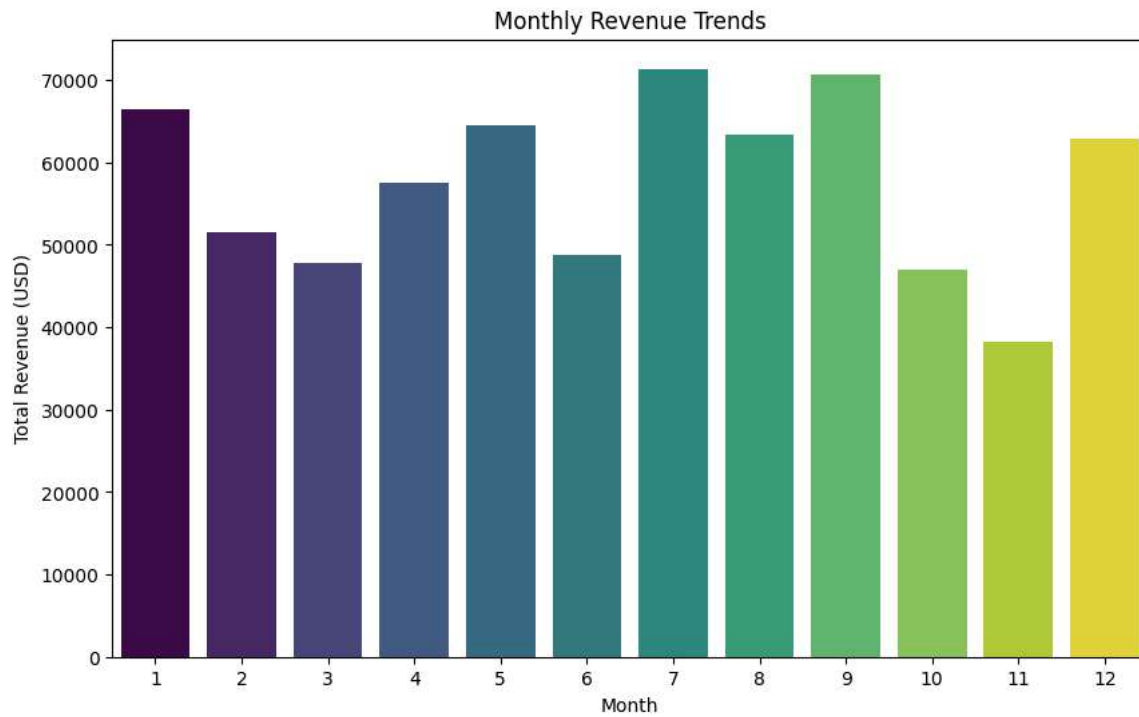


### Seasonal Trends

```
# Extract month from transaction date
merged_data['TransactionMonth'] = merged_data['TransactionDate'].dt.month

# Monthly revenue
monthly_revenue = merged_data.groupby('TransactionMonth')['TotalValue'].sum().reset_index()

# Plot monthly revenue
plt.figure(figsize=(10, 6))
sns.barplot(data=monthly_revenue, x='TransactionMonth', y='TotalValue', hue='TransactionMonth', palette='viridis', legend = False)
plt.title('Monthly Revenue Trends')
plt.xlabel('Month')
plt.ylabel('Total Revenue (USD)')
plt.show()
```



### High-Value Customers

```
# Total revenue per customer
customer_revenue = merged_data.groupby('CustomerID')['TotalValue'].sum().sort_values(ascending=False)

# Calculate cumulative contribution
customer_revenue_cumsum = customer_revenue.cumsum() / customer_revenue.sum()

# Plot the Pareto distribution
plt.figure(figsize=(10, 6))
customer_revenue_cumsum.plot(drawstyle='steps-post', color='purple')
plt.axhline(y=0.8, color='r', linestyle='--', label='80% of Revenue')
plt.title('Cumulative Revenue Contribution by Customers')
plt.xlabel('Customer Rank')
plt.ylabel('Cumulative Revenue Contribution')
plt.legend()
plt.show()
```



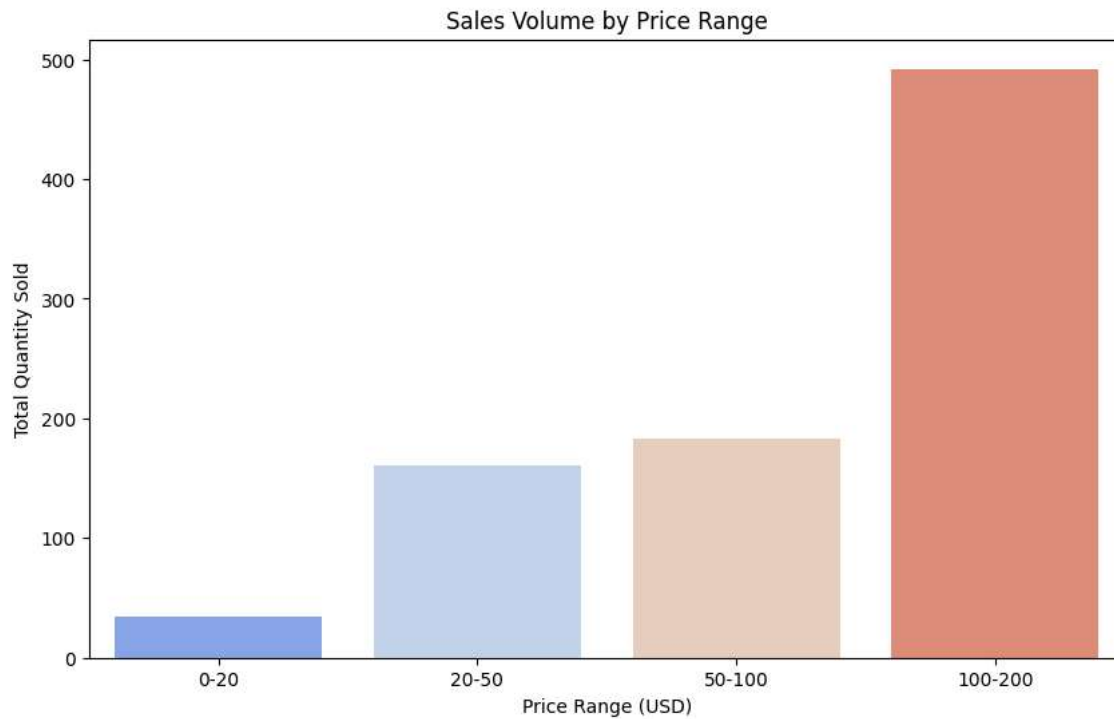
## Cumulative Revenue Contribution by Customers

### Product Pricing Impact



```
# Group products by price ranges
merged_data.rename(columns={"Price_y": "Price"}, inplace=True)
merged_data['PriceRange'] = pd.cut(merged_data['Price'], bins=[0, 20, 50, 100, 200], labels=['0-20', '20-50', '50-100', '100-200'])
price_sales = merged_data.groupby('PriceRange', observed=False)['Quantity'].sum().reset_index()
```

```
# Plot price range vs sales volume
plt.figure(figsize=(10, 6))
sns.barplot(data=price_sales, x='PriceRange', y='Quantity', hue='PriceRange', dodge=False, palette='coolwarm', legend=False)
plt.title('Sales Volume by Price Range')
plt.xlabel('Price Range (USD)')
plt.ylabel('Total Quantity Sold')
plt.show()
```



Start coding or generate with AI.