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

# Load datasets
customers_df = pd.read_csv('C:\\Users\\shaik\\Desktop\\newintern\\
Customers.csv')
transactions_df = pd.read_csv('C:\\Users\\shaik\\Desktop\\newintern\\
Transactions.csv')
products_df = pd.read_csv('C:\\Users\\shaik\\Desktop\\newintern\\
Products.csv')

# Data Cleaning
# Check for missing values
print("Missing values in Customers Data:")
print(customers_df.isnull().sum())
print("Missing values in Transactions Data:")
print(transactions_df.isnull().sum())
print("Missing values in Products Data:")
print(products_df.isnull().sum())

# Drop duplicate records
customers_df.drop_duplicates(inplace=True)
transactions_df.drop_duplicates(inplace=True)
products_df.drop_duplicates(inplace=True)

# Handle missing values (e.g., fill or drop based on analysis)
customers_df.fillna({'Region': 'Unknown'}, inplace=True)
transactions_df.dropna(inplace=True)
products_df.dropna(inplace=True)

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

# Overview of Customers dataset
print("Summary of Customers Data:")
print(customers_df.info())
print(customers_df.describe())
print(customers_df.head())

# Overview of Transactions dataset
print("Summary of Transactions Data:")
print(transactions_df.info())
print(transactions_df.describe())
print(transactions_df.head())

# Overview of Products dataset
```

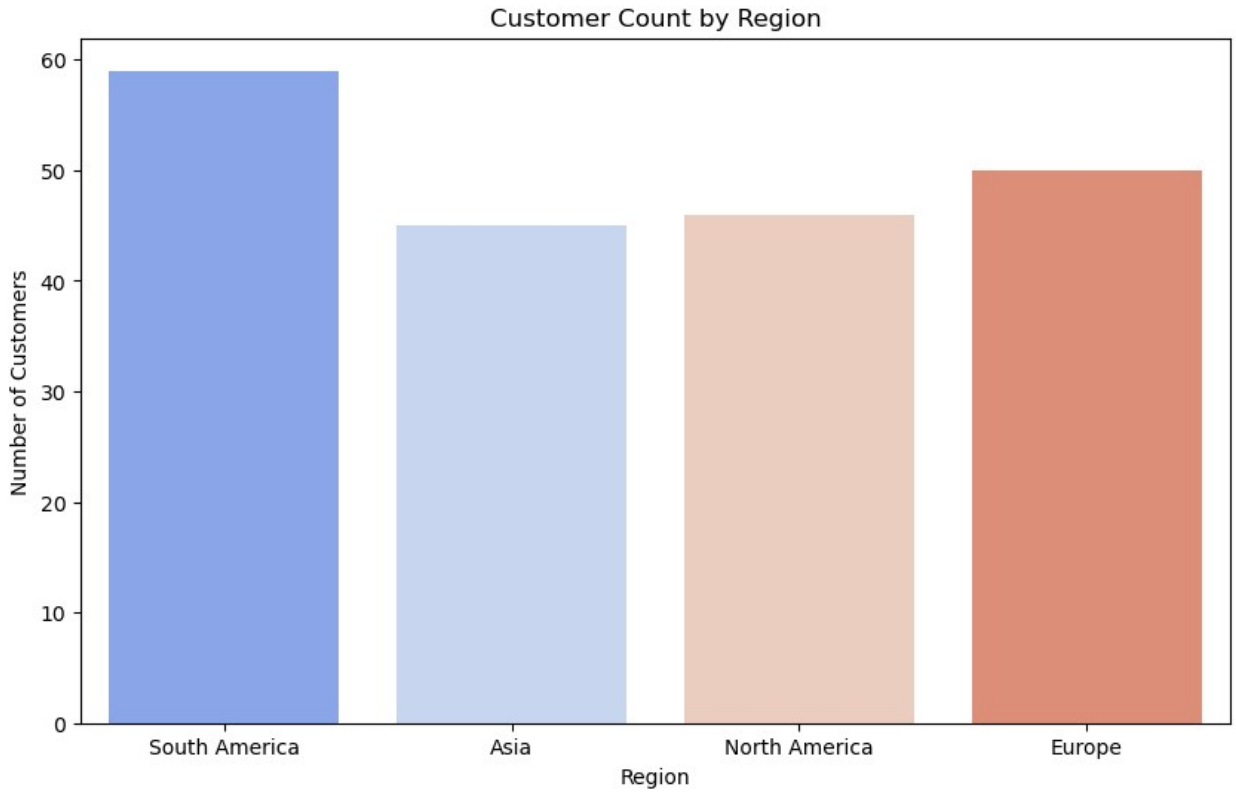
```
print("Summary of Products Data:")
print(products_df.info())
print(products_df.describe())
print(products_df.head())

# Combine customer, transaction, and product data
combined_df = transactions_df.merge(customers_df, on='CustomerID',
how='left')
combined_df = combined_df.merge(products_df, on='ProductID',
how='left')

# Analyze product performance
product_sales = combined_df.groupby('ProductID').agg({'Quantity':
'sum', 'TotalValue': 'sum'}).sort_values(by='TotalValue',
ascending=False)
print("Top Products by Sales Value:")
print(product_sales.head())

# Data Visualization
plt.figure(figsize=(10, 6))
sns.countplot(data=customers_df, x='Region', palette='coolwarm')
plt.title('Customer Count by Region')
plt.xlabel('Region')
plt.ylabel('Number of Customers')
plt.show()

plt.figure(figsize=(10, 6))
sns.histplot(combined_df['TotalValue'], bins=40, kde=True,
color='blue')
plt.title('Transaction Value Distribution')
plt.xlabel('Transaction Value')
plt.ylabel('Frequency')
plt.show()
```

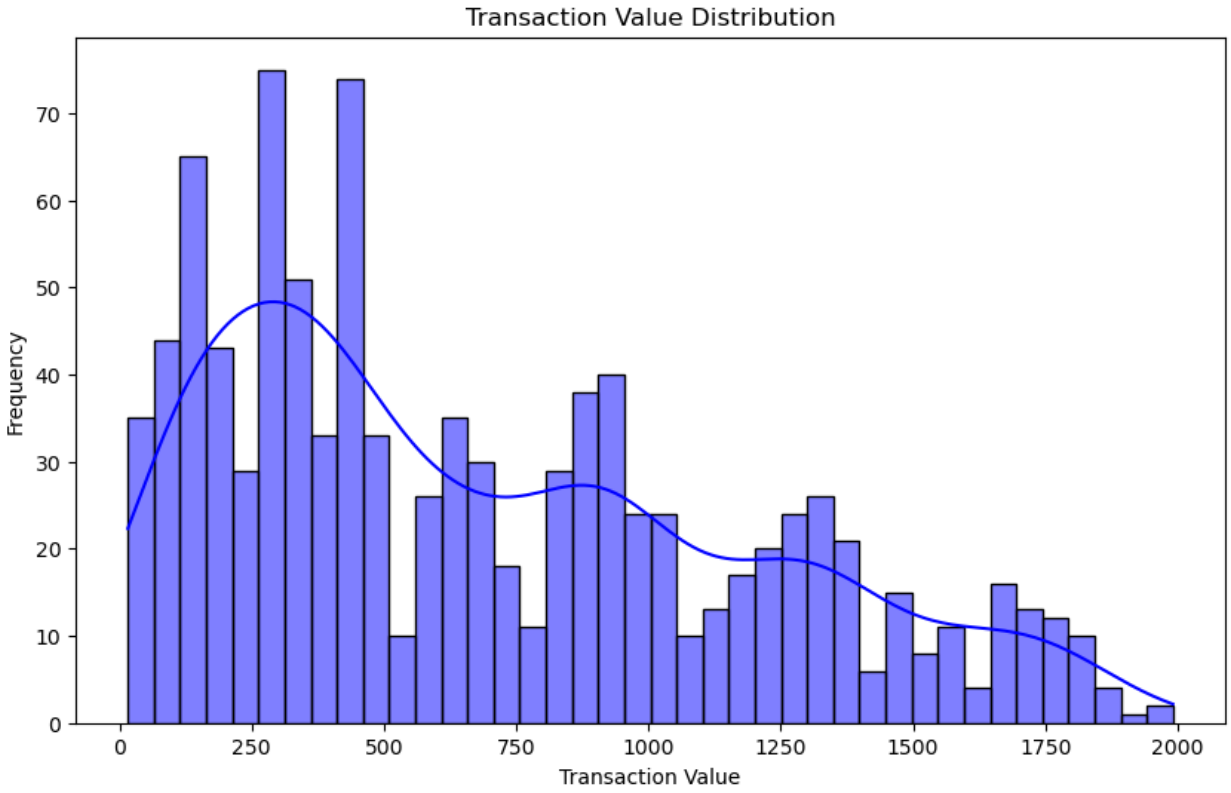
```
Missing values in Customers Data:
CustomerID      0
CustomerName    0
Region          0
SignupDate      0
dtype: int64
Missing values in Transactions Data:
TransactionID      0
CustomerID         0
ProductID          0
TransactionDate    0
Quantity           0
TotalValue         0
```

```
Price              0
dtype: int64
Missing values in Products Data:
ProductID      0
ProductName    0
Category       0
Price          0
dtype: int64
Summary of Customers Data:
<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
                SignupDate
count                  200
mean   2023-07-19 08:31:12
min    2022-01-22 00:00:00
25%    2022-09-26 12:00:00
50%    2023-08-31 12:00:00
75%    2024-04-12 12:00:00
max    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
Summary of Transactions Data:
<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   datetime64[ns]
 4   Quantity         1000 non-null   int64
 5   TotalValue       1000 non-null   float64
 6   Price            1000 non-null   float64
dtypes: datetime64[ns](1), float64(2), int64(1), object(3)
```

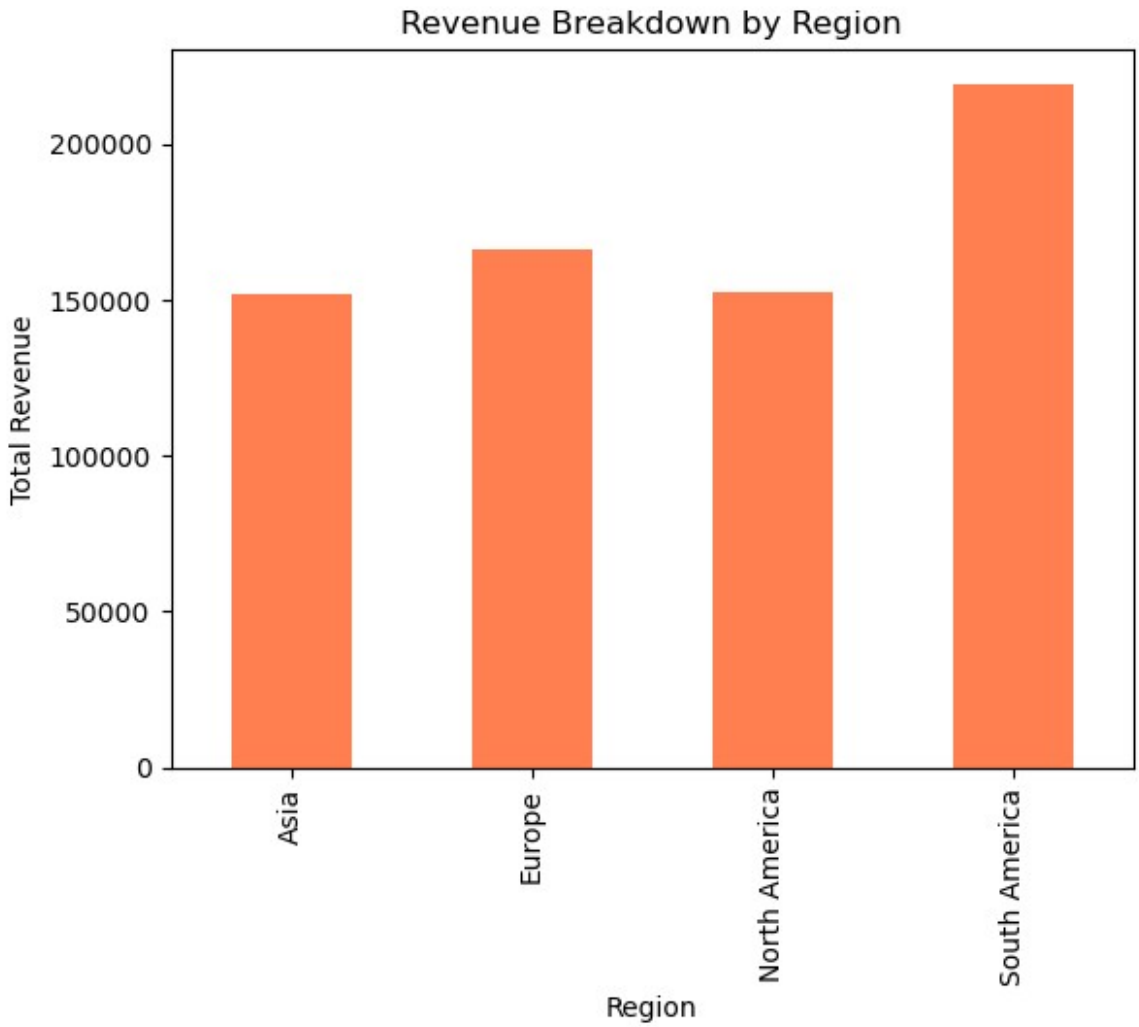
```
memory usage: 54.8+ KB
None
                      TransactionDate     Quantity   TotalValue
Price
count                            1000  1000.000000  1000.000000
1000.00000
mean   2024-06-23 15:33:02.768999936     2.537000   689.995560
272.55407
min              2023-12-30 15:29:12     1.000000    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
  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
Summary of Products Data:
<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
            Price
count  100.000000
mean   267.551700
```

```
std    143.219383
min     16.080000
25%    147.767500
50%    292.875000
75%    397.090000
max    497.760000
  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
Top Products by Sales Value:
           Quantity  TotalValue
ProductID
P029             45    19513.80
P079             43    17946.91
P048             43    17905.20
P020             38    15060.92
P062             39    14592.24
```



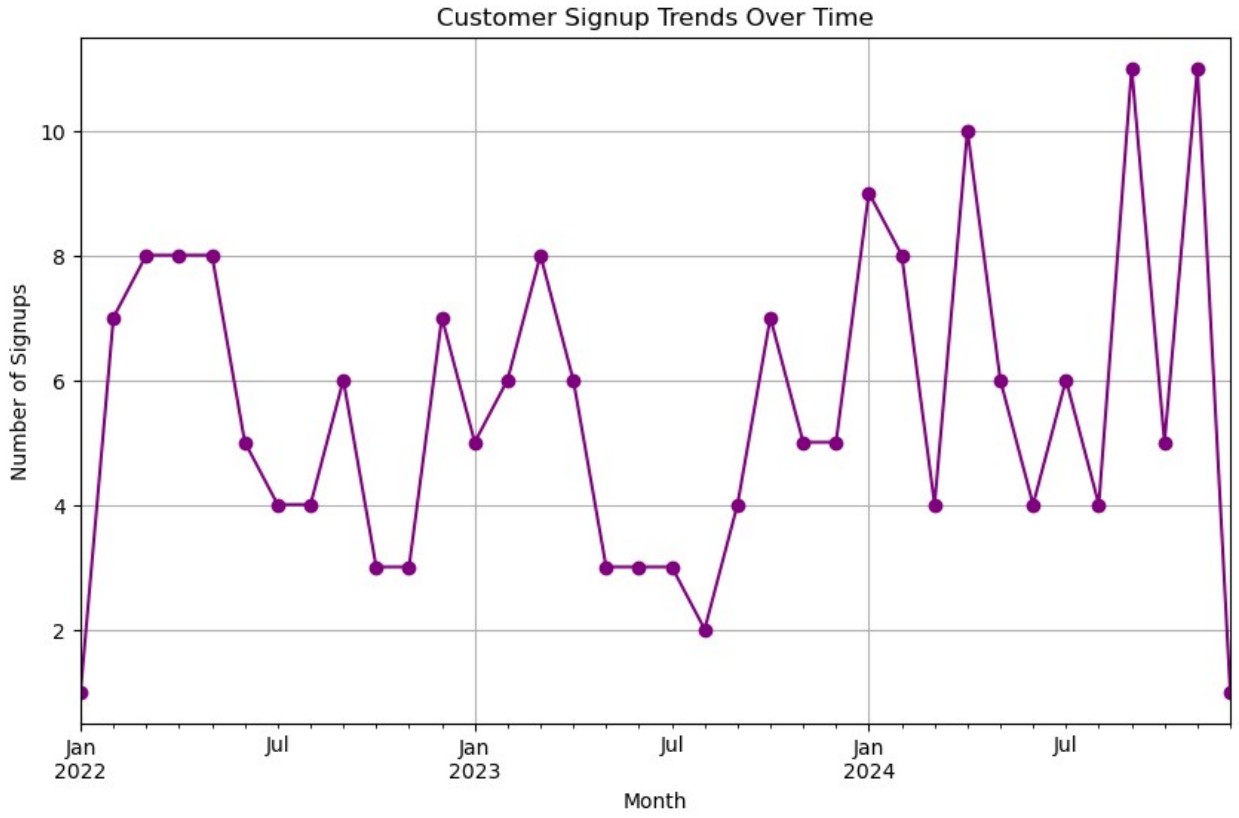



```
# Revenue by region
revenue_by_region = combined_df.groupby('Region')['TotalValue'].sum()
revenue_by_region.plot(kind='bar', title='Revenue Breakdown by
Region', color='coral')
plt.xlabel('Region')
plt.ylabel('Total Revenue')
plt.show()
```

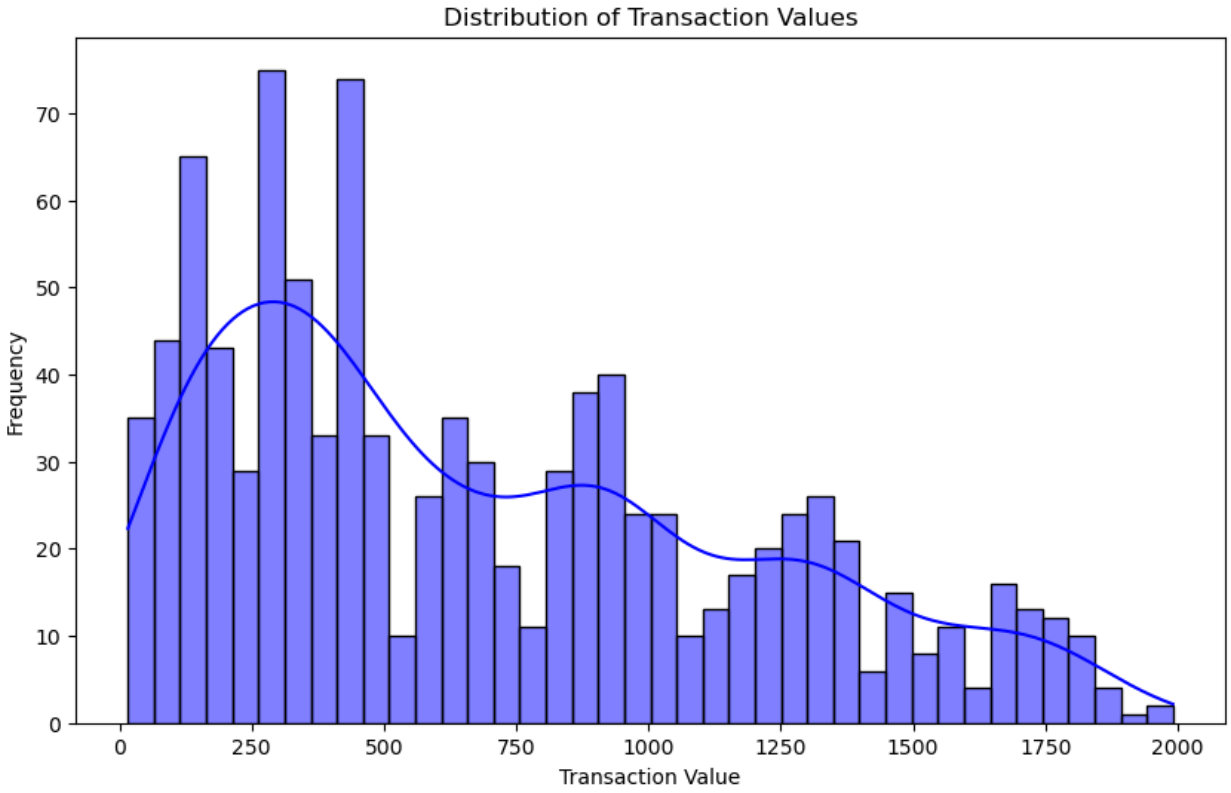



```
# Convert signup date to month period
customers_df['SignupMonth'] =
customers_df['SignupDate'].dt.to_period('M')

# Count customer sign-ups by month
signup_trends = customers_df.groupby('SignupMonth').size()
print("Monthly Customer Signups:")
print(signup_trends)

# Visualize customer sign-up trends
plt.figure(figsize=(10, 6))
signup_trends.plot(marker='o', color='purple')
plt.title('Customer Signup Trends Over Time')
plt.xlabel('Month')
plt.ylabel('Number of Signups')
plt.grid(True)
plt.show()
```

```
Monthly Customer Signups:
SignupMonth
2022-01     1
2022-02     7
2022-03     8
2022-04     8
2022-05     8
2022-06     5
2022-07     4
2022-08     4
2022-09     6
2022-10     3
2022-11     3
2022-12     7
2023-01     5
2023-02     6
2023-03     8
2023-04     6
2023-05     3
2023-06     3
2023-07     3
2023-08     2
2023-09     4
2023-10     7
2023-11     5
2023-12     5
2024-01     9
2024-02     8
2024-03     4
2024-04    10
2024-05     6
2024-06     4
2024-07     6
2024-08     4
2024-09    11
2024-10     5
2024-11    11
2024-12     1
Freq: M, dtype: int64
```

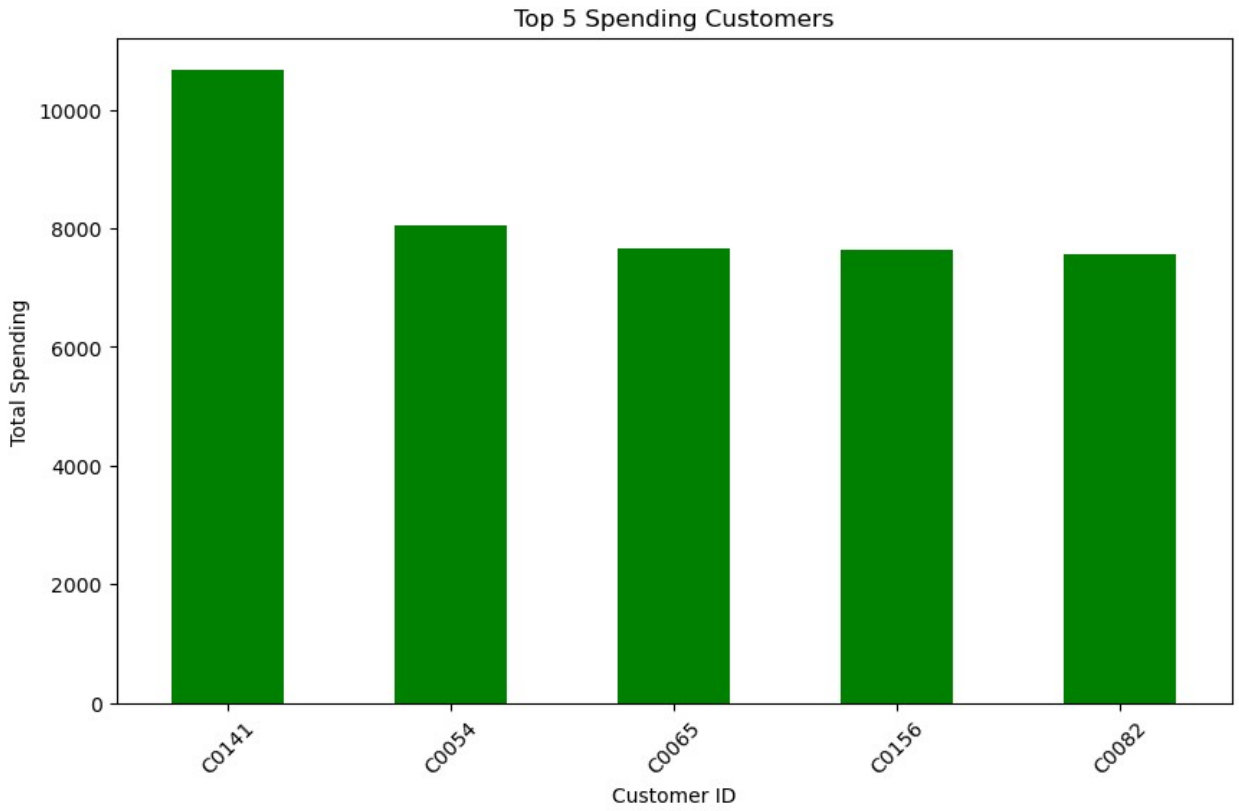



```
# Visualize transaction value distribution
plt.figure(figsize=(10, 6))
sns.histplot(combined_df['TotalValue'], bins=40, kde=True,
color='blue')
plt.title('Distribution of Transaction Values')
plt.xlabel('Transaction Value')
plt.ylabel('Frequency')
plt.show()
```



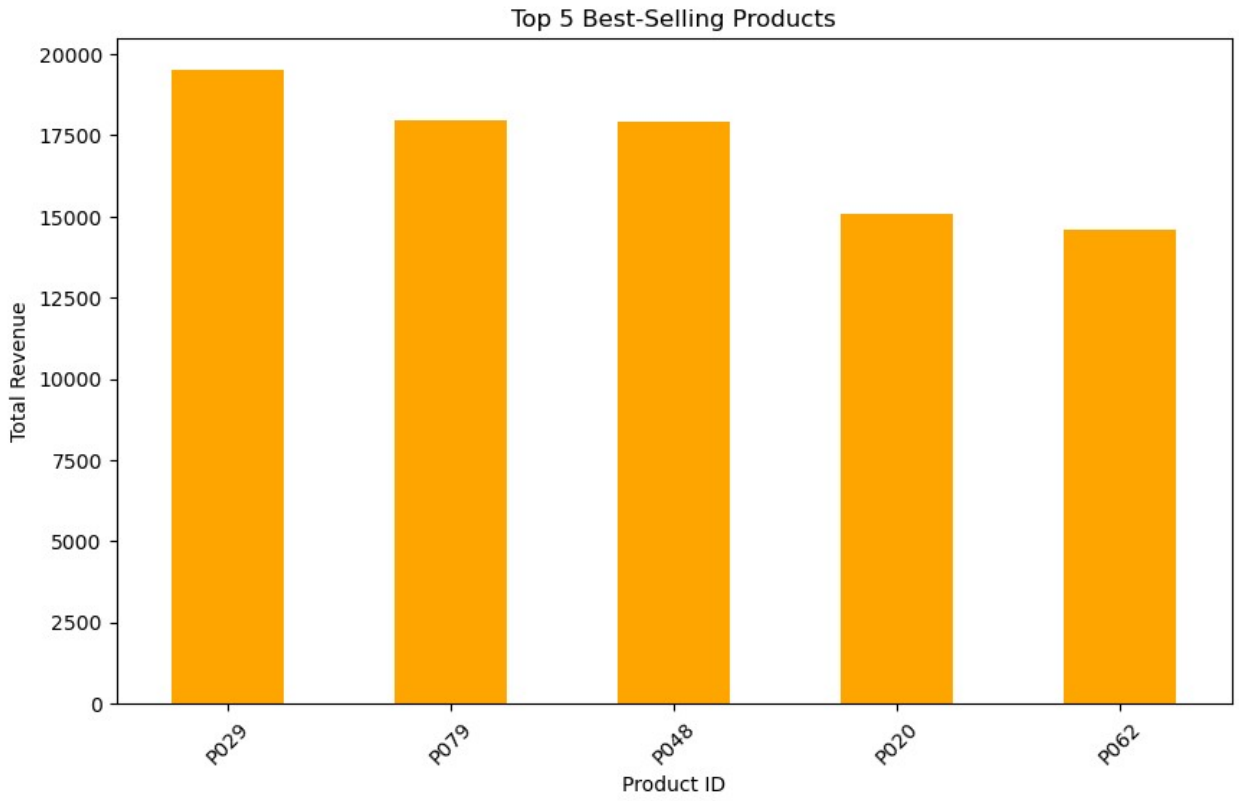

```
# Identify the top 5 spending customers
top_customers = combined_df.groupby('CustomerID')
['TotalValue'].sum().nlargest(5)
print("Top 5 Spending Customers:")
print(top_customers)

# Visualize top spending customers
plt.figure(figsize=(10, 6))
top_customers.plot(kind='bar', color='green')
plt.title('Top 5 Spending Customers')
plt.xlabel('Customer ID')
plt.ylabel('Total Spending')
plt.xticks(rotation=45)
plt.show()
```

```
Top 5 Spending Customers:
CustomerID
C0141    10673.87
C0054     8040.39
C0065     7663.70
C0156     7634.45
C0082     7572.91
Name: TotalValue, dtype: float64
```




```
# Determine best-selling products by total revenue
top_products = combined_df.groupby('ProductID')
['TotalValue'].sum().nlargest(5)
print("Top 5 Best-Selling Products:")
print(top_products)

# Visualize best-selling products
plt.figure(figsize=(10, 6))
top_products.plot(kind='bar', color='orange')
plt.title('Top 5 Best-Selling Products')
plt.xlabel('Product ID')
plt.ylabel('Total Revenue')
plt.xticks(rotation=45)
plt.show()
```

```
Top 5 Best-Selling Products:
ProductID
P029    19513.80
P079    17946.91
P048    17905.20
P020    15060.92
P062    14592.24
Name: TotalValue, dtype: float64
```


Top 5 Best-Selling Products
Total Revenue
20000
17500
15000
12500
10000
7500
5000
2500
0
P029
P079
P048
P020
P062
Product ID