In [7]:

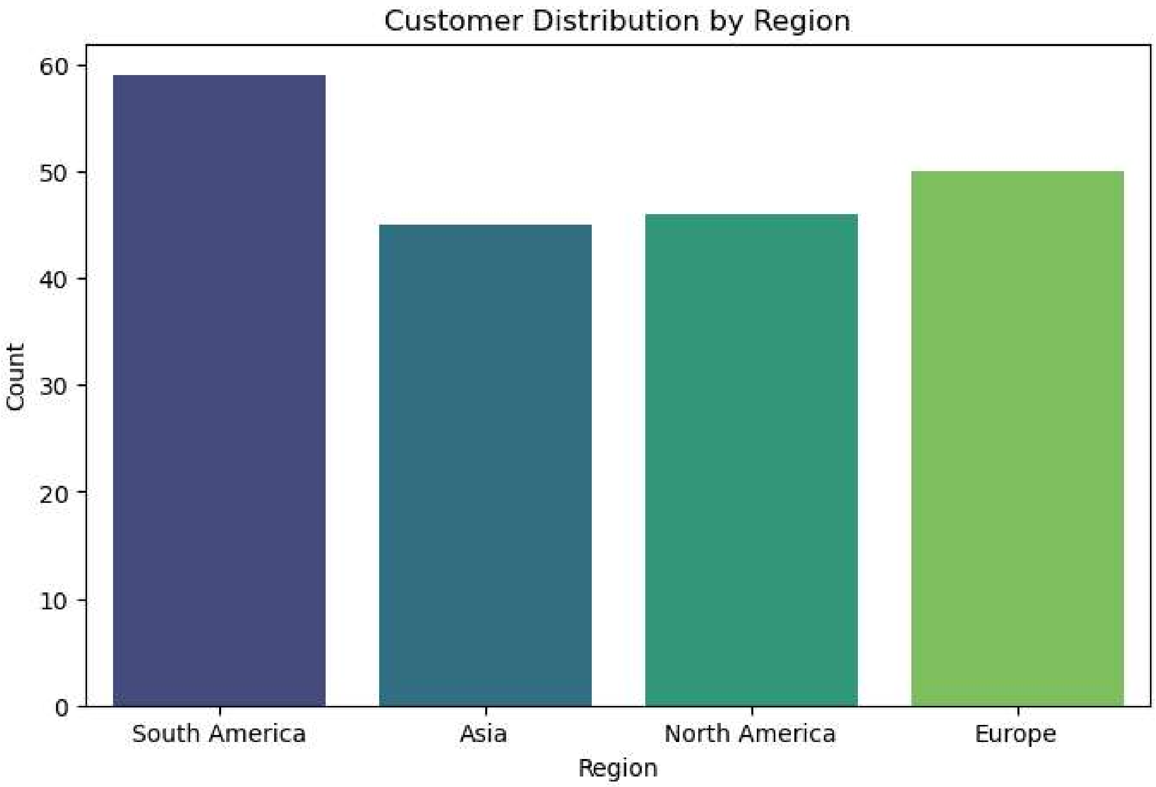
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load datasets
customers = pd.read_csv("Customers.csv")
products = pd.read_csv("Products.csv")
transactions = pd.read_csv("Transactions.csv")

# Display basic information
print(customers.info())
print(products.info())
print(transactions.info())

# Check for missing values
print(customers.isnull().sum())
print(products.isnull().sum())
print(transactions.isnull().sum())

# Convert date columns to datetime
customers['SignupDate'] = pd.to_datetime(customers['SignupDate'])
transactions['TransactionDate'] = pd.to_datetime(transactions['TransactionDat

# Preview datasets
print(customers.head())
print(products.head())
print(transactions.head())
```

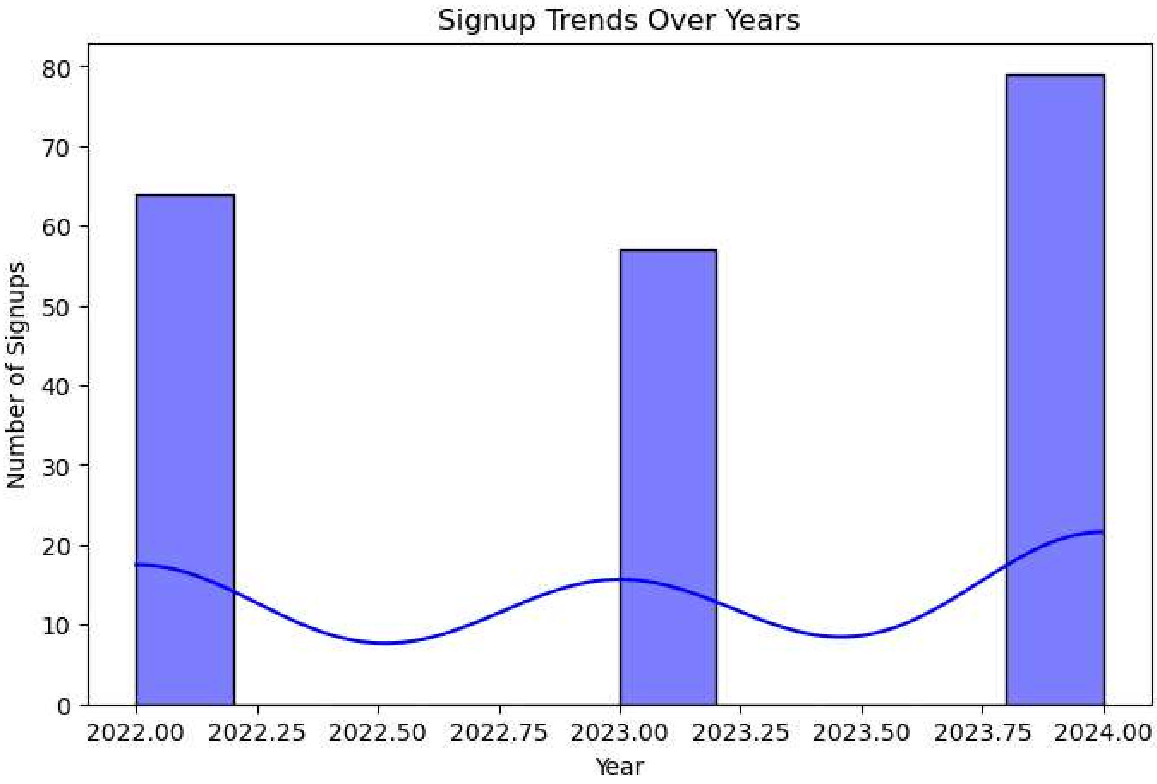
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   CustomerID    200 non-null    object
 1   CustomerName  200 non-null    object
 2   Region        200 non-null    object
 3   SignupDate    200 non-null    object
dtypes: object(4)
memory usage: 6.4+ KB
None
<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
<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
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
```

```
  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 [8]:

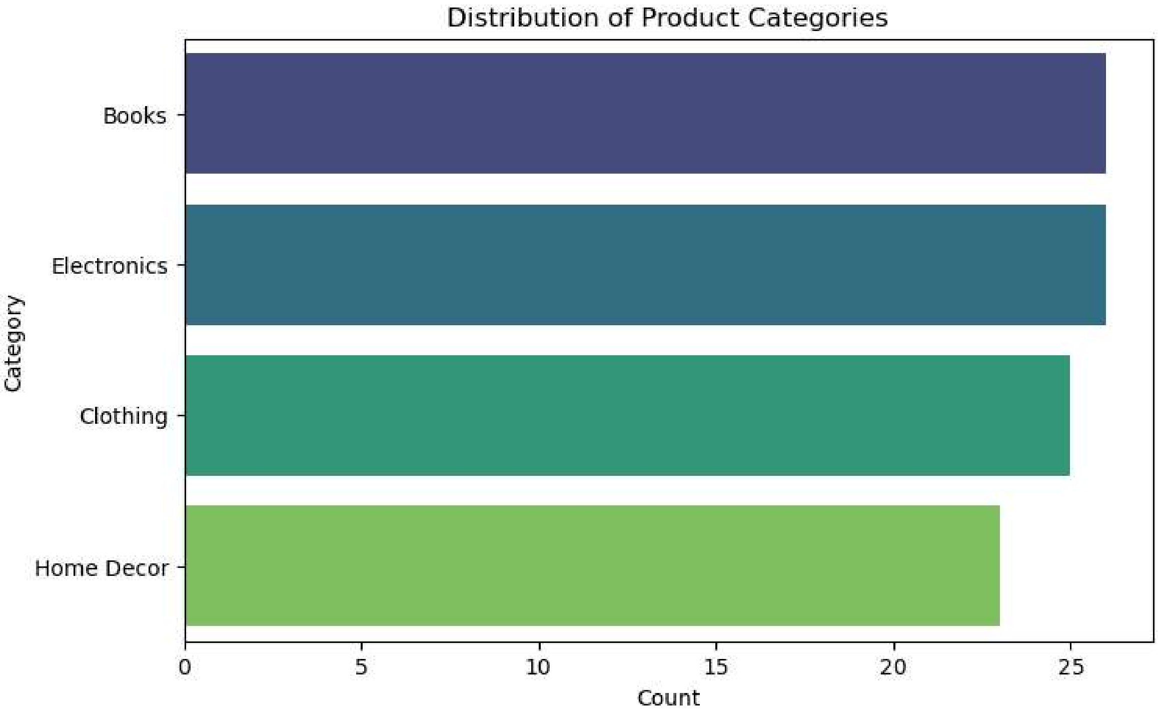
```
# Plot customer distribution by region
plt.figure(figsize=(8, 5))
sns.countplot(data=customers, x='Region', palette='viridis')
plt.title("Customer Distribution by Region")
plt.xlabel("Region")
plt.ylabel("Count")
plt.show()

# Analyze signup trends
customers['SignupYear'] = customers['SignupDate'].dt.year
plt.figure(figsize=(8, 5))
sns.histplot(data=customers, x='SignupYear', bins=10, kde=True, color='blue')
plt.title("Signup Trends Over Years")
plt.xlabel("Year")
plt.ylabel("Number of Signups")
plt.show()
```

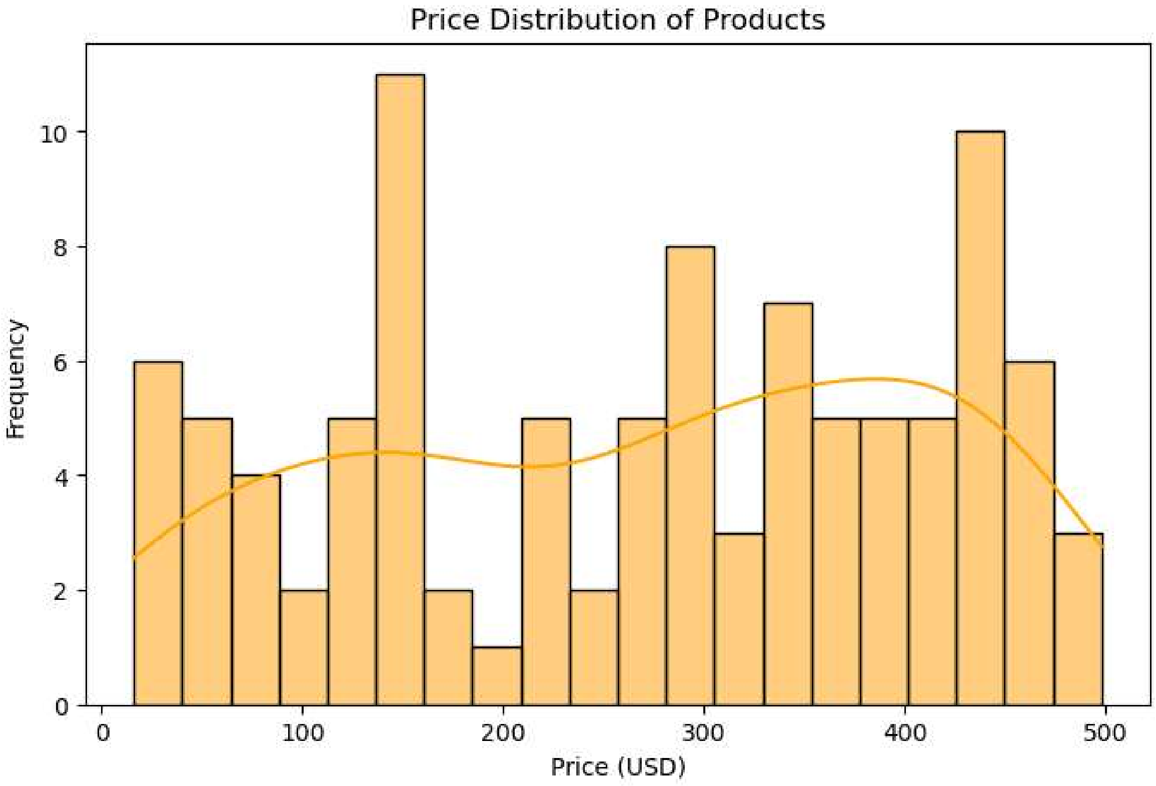



Signup Trends Over Years
Number of Signups
0
10
20
30
40
50
60
70
80
2022.00
2022.25
2022.50
2022.75
2023.00
2023.25
2023.50
2023.75
2024.00
Year

In [9]:

```
# Plot product categories
plt.figure(figsize=(8, 5))
sns.countplot(data=products, y='Category', palette='viridis', order=products[
plt.title("Distribution of Product Categories")
plt.xlabel("Count")
plt.ylabel("Category")
plt.show()

# Analyze price distribution
plt.figure(figsize=(8, 5))
sns.histplot(data=products, x='Price', bins=20, kde=True, color='orange')
plt.title("Price Distribution of Products")
plt.xlabel("Price (USD)")
plt.ylabel("Frequency")
plt.show()
```





In [10]:

```
# Ensure TotalValue is numeric
transactions['TotalValue'] = pd.to_numeric(transactions['TotalValue'], errors

# Ensure Month is a string or period type
transactions['Month'] = transactions['TransactionDate'].dt.to_period('M')

# Group by Month and recalculate monthly_sales
monthly_sales = transactions.groupby('Month')['TotalValue'].sum().reset_index

# Verify data types again
print(monthly_sales.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13 entries, 0 to 12
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype    
---  ------      --------------  -----    
 0   Month       13 non-null     period[M]
 1   TotalValue  13 non-null     float64  
dtypes: float64(1), period[M](1)
memory usage: 336.0 bytes
None
```

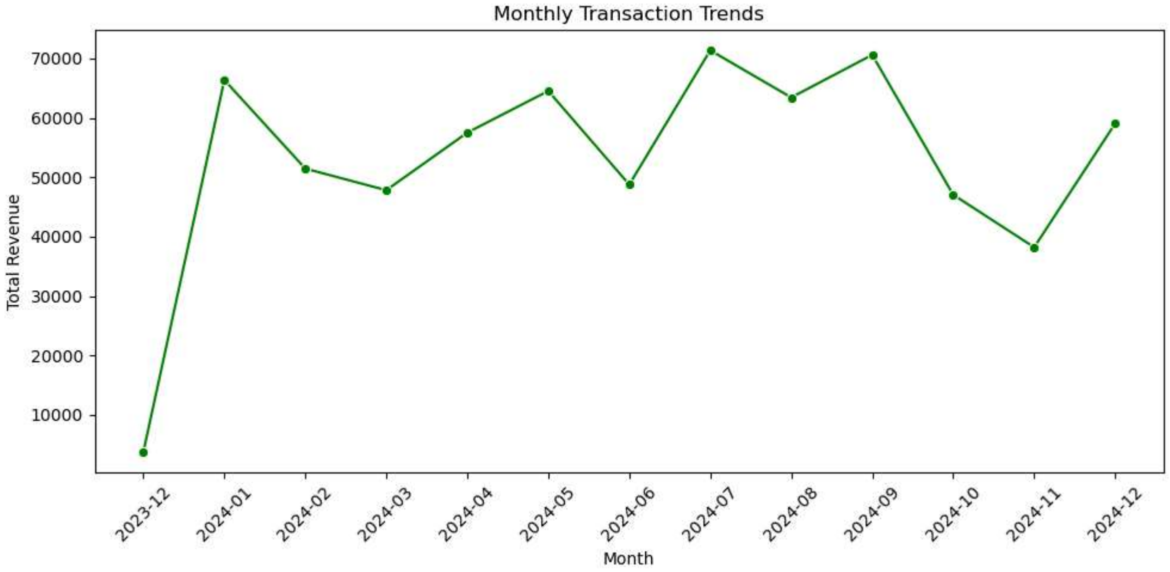
In [11]:

```
# Check for NaN values
print(monthly_sales.isnull().sum())

# Drop rows with NaN if any
monthly_sales = monthly_sales.dropna()

# Alternatively, fill missing values with 0
# monthly_sales['TotalValue'] = monthly_sales['TotalValue'].fillna(0)
```

```
Month         0
TotalValue    0
dtype: int64
```

In [12]:

```
# Convert Month to string for plotting
monthly_sales['Month'] = monthly_sales['Month'].astype(str)

# Plot Monthly Transaction Trends
plt.figure(figsize=(10, 5))
sns.lineplot(data=monthly_sales, x='Month', y='TotalValue', marker='o', color
plt.title("Monthly Transaction Trends")
plt.xlabel("Month")
plt.ylabel("Total Revenue")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [ ]: