Import Libraries

```
import numpy as np
import pandas as pd

# Data Visualization
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings("ignore")
```

Loadind Customer DataSet

```
customer = pd.read_csv("Customers.csv")
```

Loading Products Dataset

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

Loading Transcations DataSet

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

```
customer.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.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
```

```
transcation.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  
```

Task 1: **Exploratory Data Analysis** (EDA) and Business Insights

```
# Load the combined dataset
combined_data = pd.read_csv('KishoreReddy_V_Combined_Data.csv')

combined_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  
```

```
combined_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 14 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_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
 13  Month            1000 non-null   int32
dtypes: datetime64[ns](1), float64(3), int32(1), int64(1), object(8)
memory usage: 105.6+ KB
```

```
combined_data.describe()
```

```
                     TransactionDate     Quantity   TotalValue
Price_x  \
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

          Price_y        Month
count  1000.00000  1000.000000
mean    272.55407     6.288000
min      16.08000     1.000000
25%     147.95000     3.000000
50%     299.93000     6.000000
75%     404.40000     9.000000
max     497.76000    12.000000
std     140.73639     3.437859
```

```
combined_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
```

```
combined_data.columns
```

```
Index(['TransactionID', 'CustomerID', 'ProductID', 'TransactionDate',
       'Quantity', 'TotalValue', 'Price_x', 'CustomerName', 'Region',
       'SignupDate', 'ProductName', 'Category', 'Price_y', 'Month'],
      dtype='object')
```

```
# Data Cleaning
combined_data['TransactionDate'] =
pd.to_datetime(combined_data['TransactionDate'])
combined_data['TotalValue'] = combined_data['Quantity'] *
combined_data['Price_y']
```

```
combined_data['TransactionDate']
```

```
0     2024-08-25 12:38:23
1     2024-05-27 22:23:54
2     2024-04-25 07:38:55
3     2024-03-26 22:55:37
4     2024-03-21 15:10:10
              ...
995   2024-10-24 08:30:27
996   2024-06-04 02:15:24
997   2024-04-05 13:05:32
998   2024-09-29 10:16:02
999   2024-04-21 10:52:24
Name: TransactionDate, Length: 1000, dtype: datetime64[ns]
```

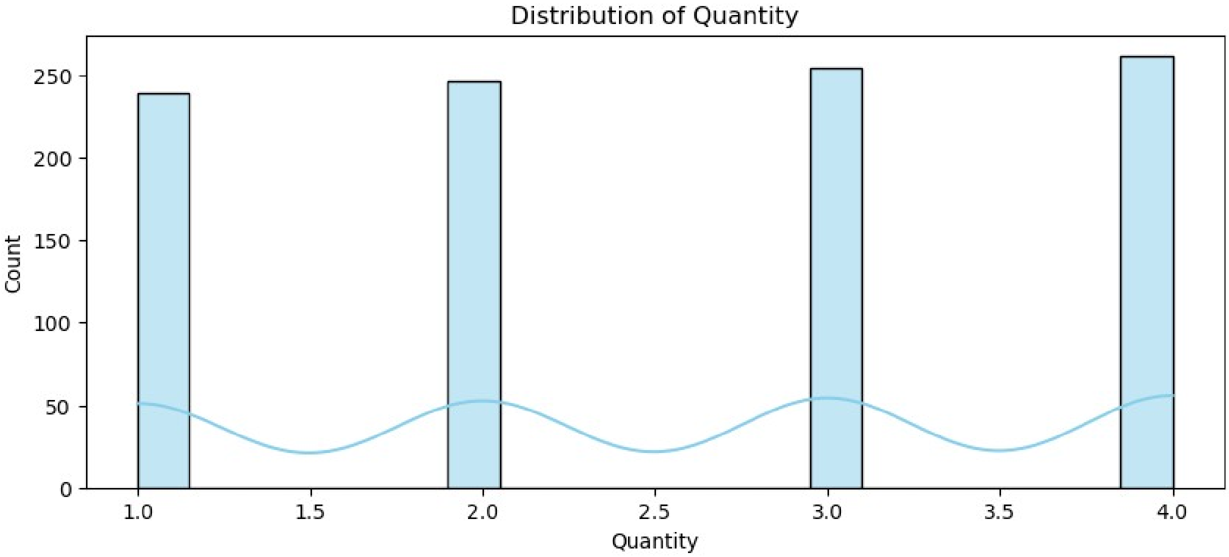
```
print(combined_data['TotalValue'])
```

```
0       300.68
1       300.68
2       300.68
3       601.36
4       902.04
         ...
```

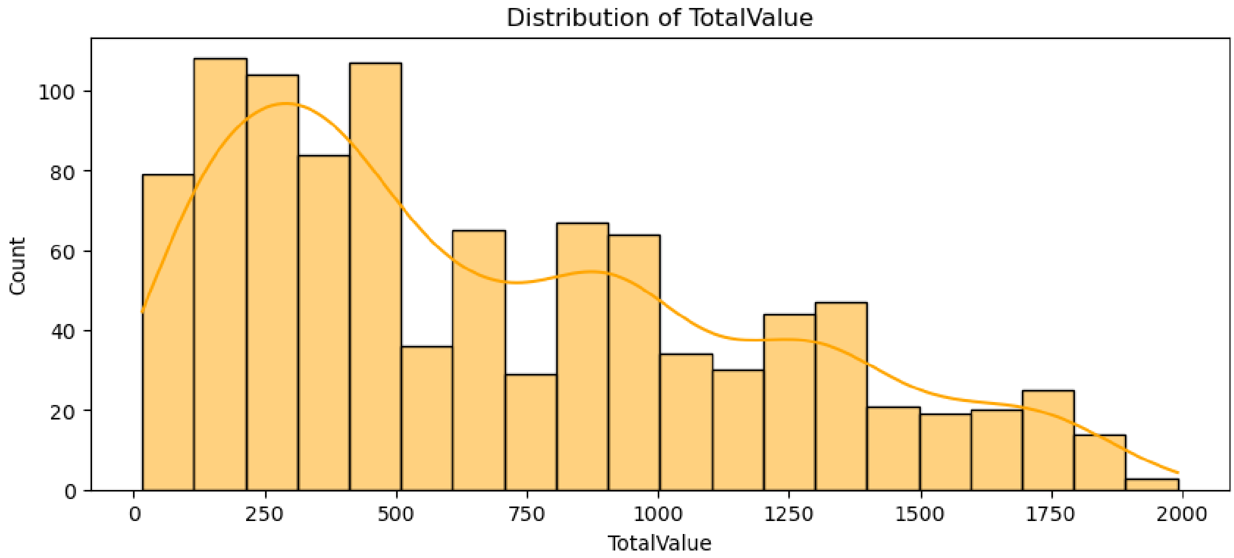
```
995     459.86
996    1379.58
997    1839.44
998     919.72
999     459.86
Name: TotalValue, Length: 1000, dtype: float64
```

1. Univariate Analysis

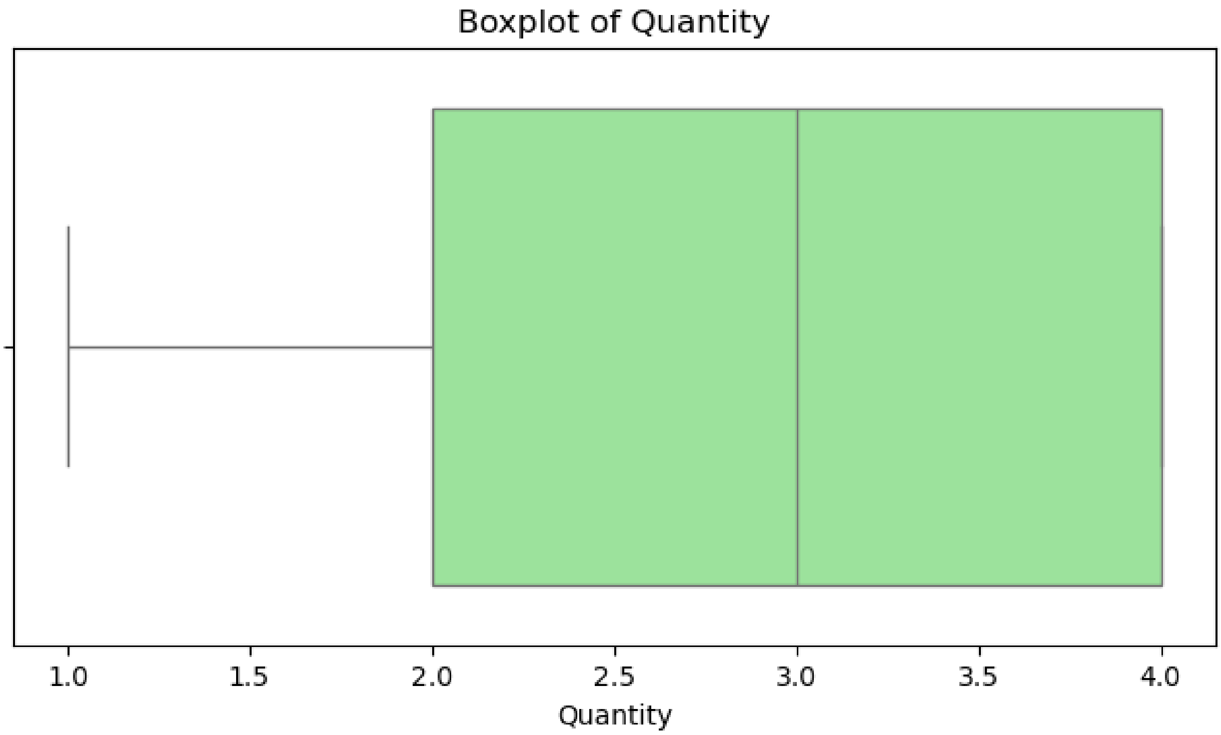
```
# Distribution of Quantity and TotalValue
plt.figure(figsize=(10, 4))
sns.histplot(combined_data['Quantity'], kde=True, bins=20,
color='skyblue')
plt.title('Distribution of Quantity')
plt.show()
```



```
plt.figure(figsize=(10, 4))
sns.histplot(combined_data['TotalValue'], kde=True, bins=20,
color='orange')
plt.title('Distribution of TotalValue')
plt.show()
```

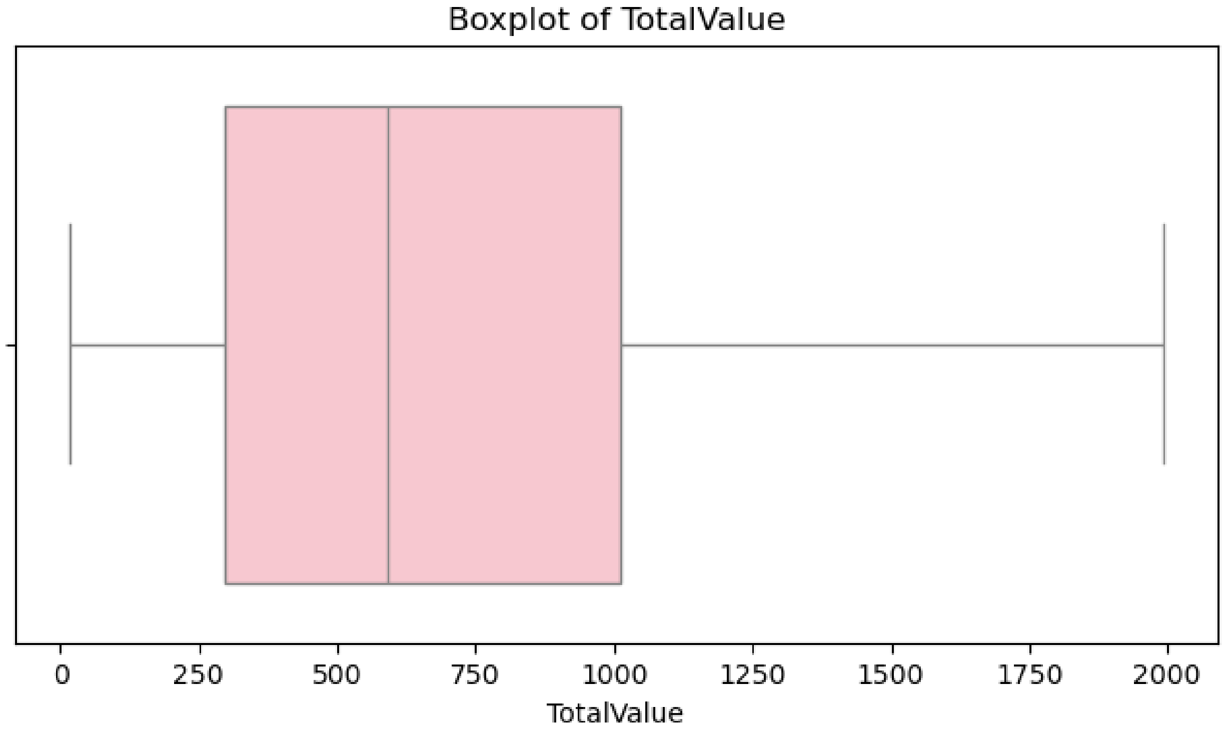


```
plt.figure(figsize=(8, 4))
sns.boxplot(x=combined_data['Quantity'], color='lightgreen')
plt.title('Boxplot of Quantity')
plt.show()
```

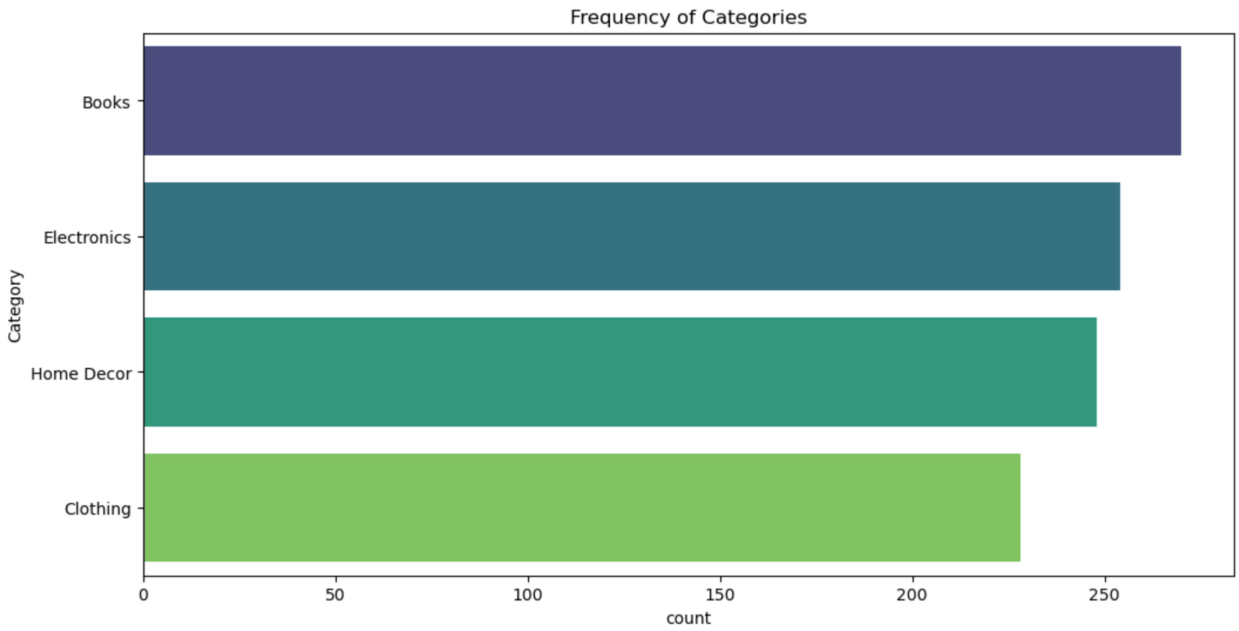


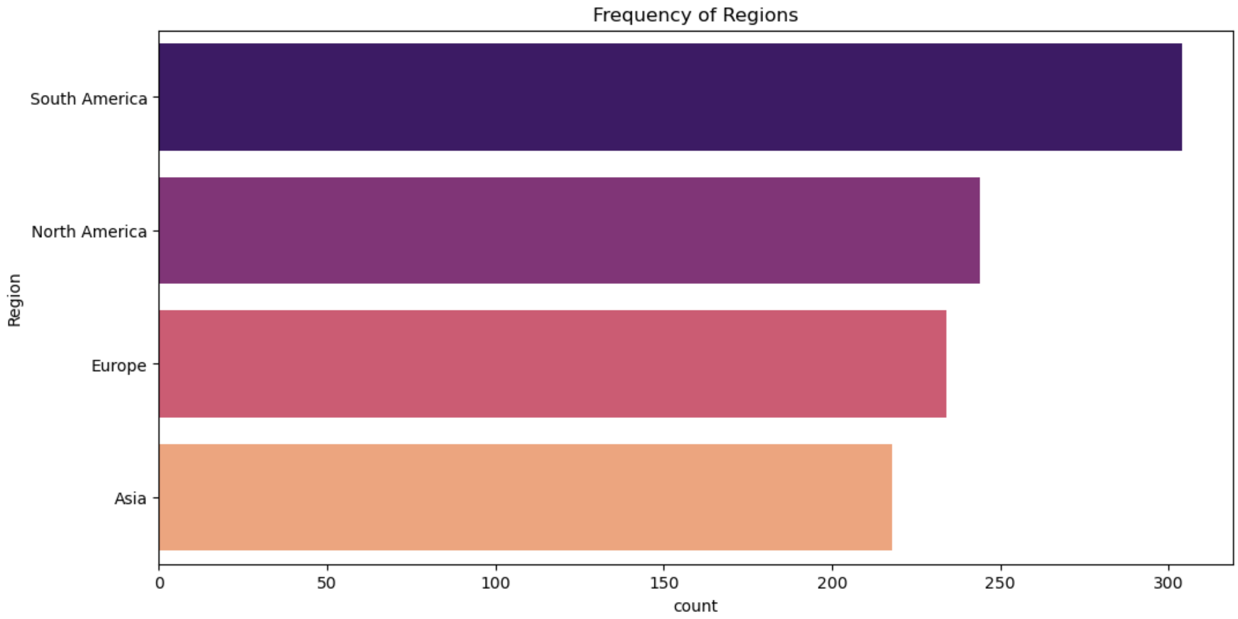
```
plt.figure(figsize=(8, 4))
sns.boxplot(x=combined_data['TotalValue'], color='pink')
```

```
plt.title('Boxplot of TotalValue')
plt.show()
```



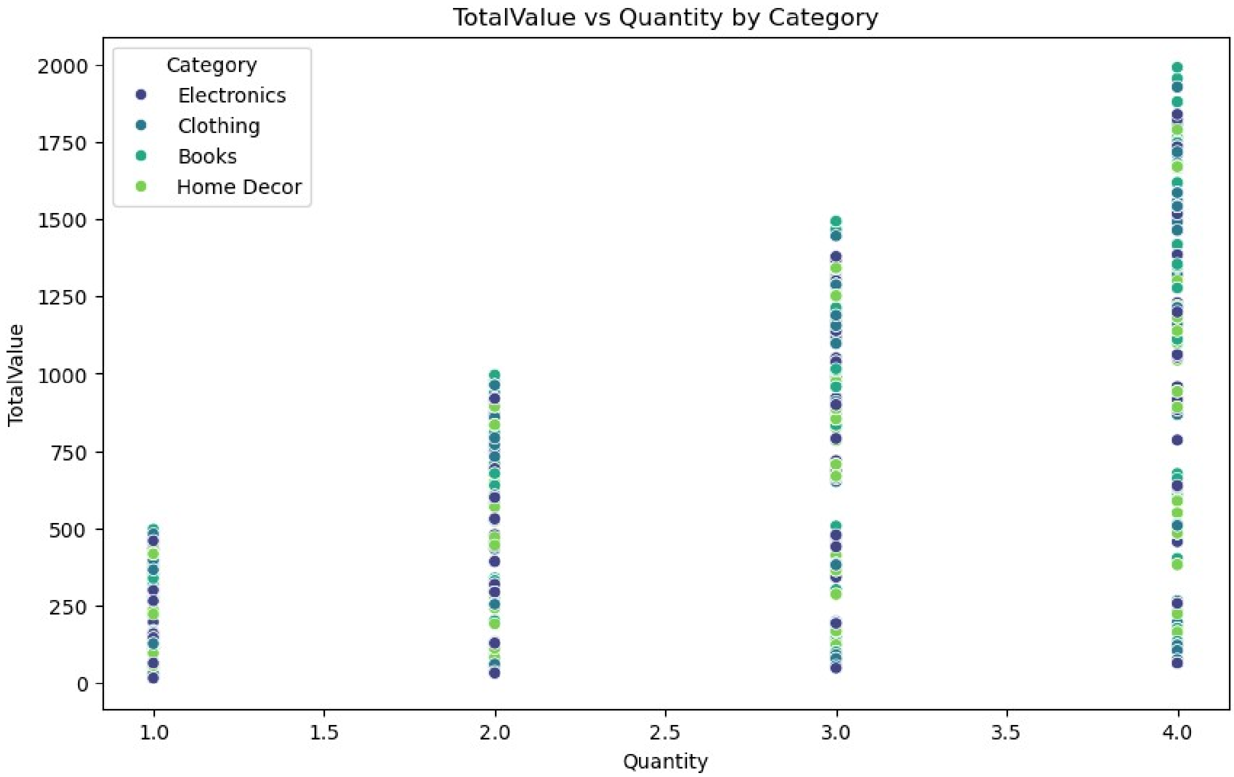
```
# Bar plot for Category and Region
plt.figure(figsize=(12, 6))
sns.countplot(y='Category', data=combined_data,
order=combined_data['Category'].value_counts().index,
palette='viridis')
plt.title('Frequency of Categories')
plt.show()

plt.figure(figsize=(12, 6))
sns.countplot(y='Region', data=combined_data,
order=combined_data['Region'].value_counts().index, palette='magma')
plt.title('Frequency of Regions')
plt.show()
```



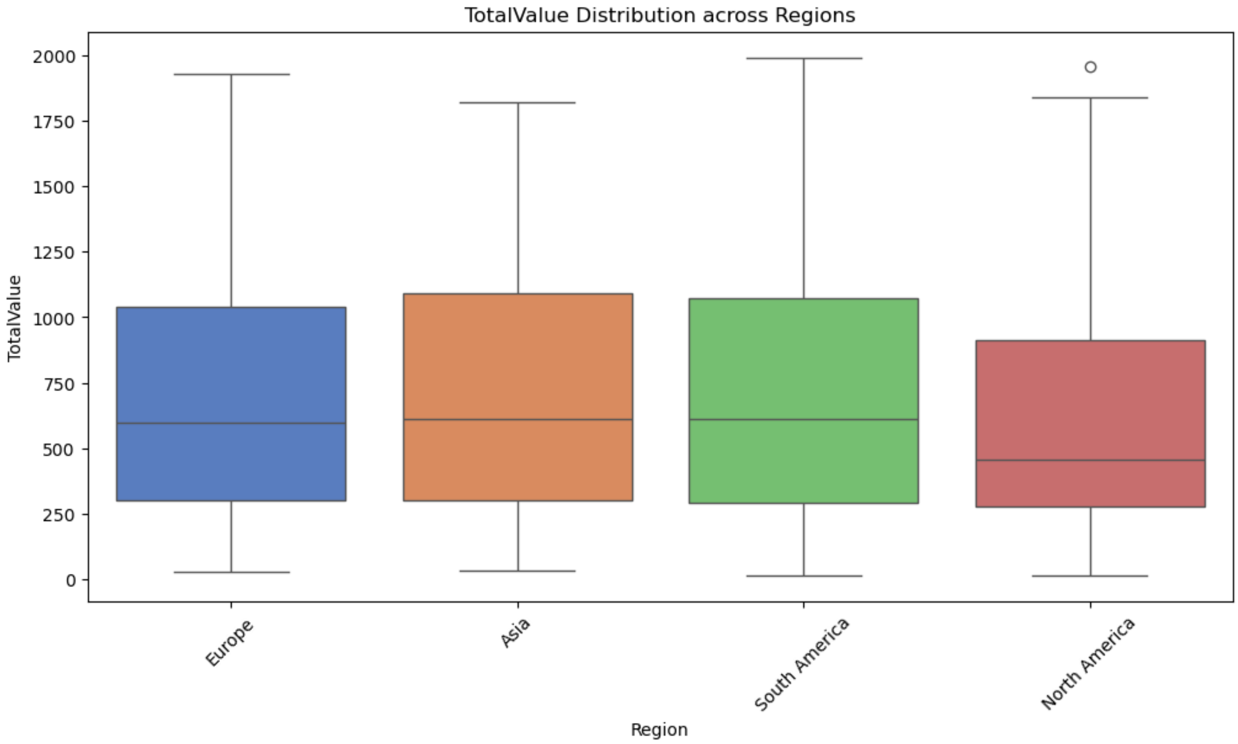


2.Bivariate Analysis

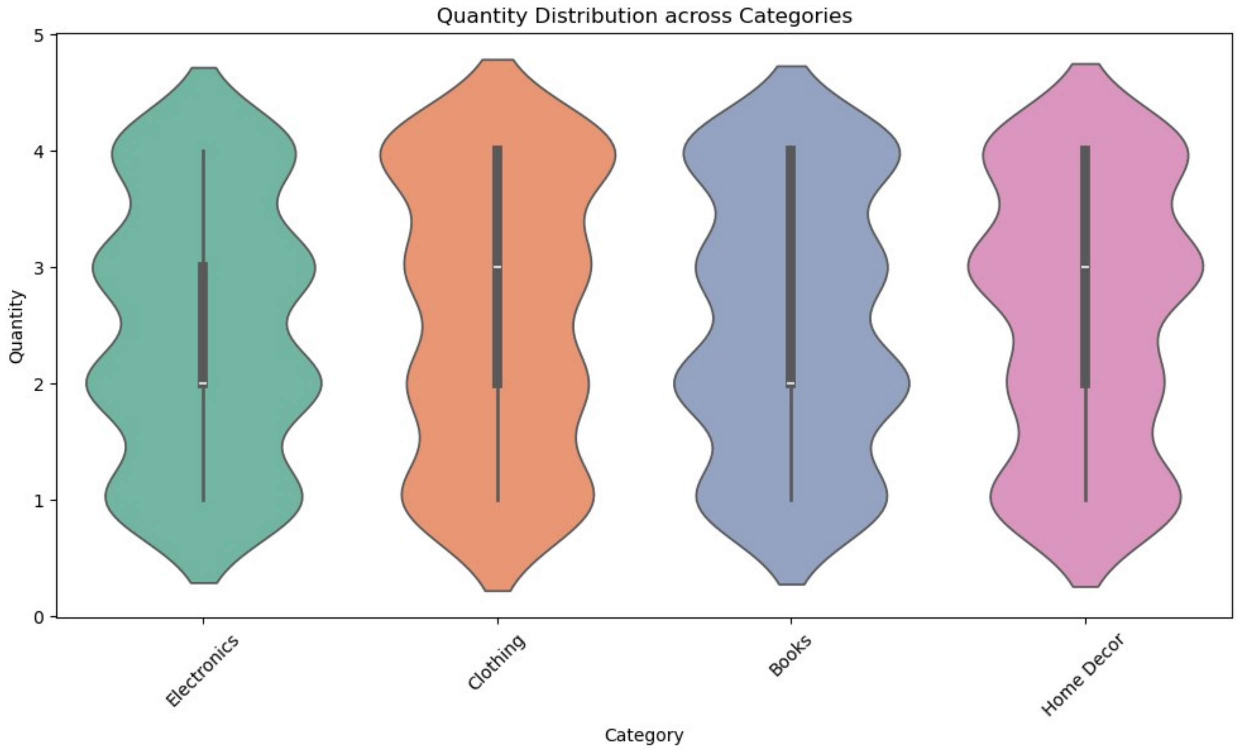
```
# TotalValue vs Quantity
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Quantity', y='TotalValue', data=combined_data,
hue='Category', palette='viridis')
plt.title('TotalValue vs Quantity by Category')
plt.show()
```



```python
# TotalValue across Regions
plt.figure(figsize=(12, 6))
sns.boxplot(x='Region', y='TotalValue', data=combined_data,
palette='muted')
plt.title('TotalValue Distribution across Regions')
plt.xticks(rotation=45)
plt.show()
```

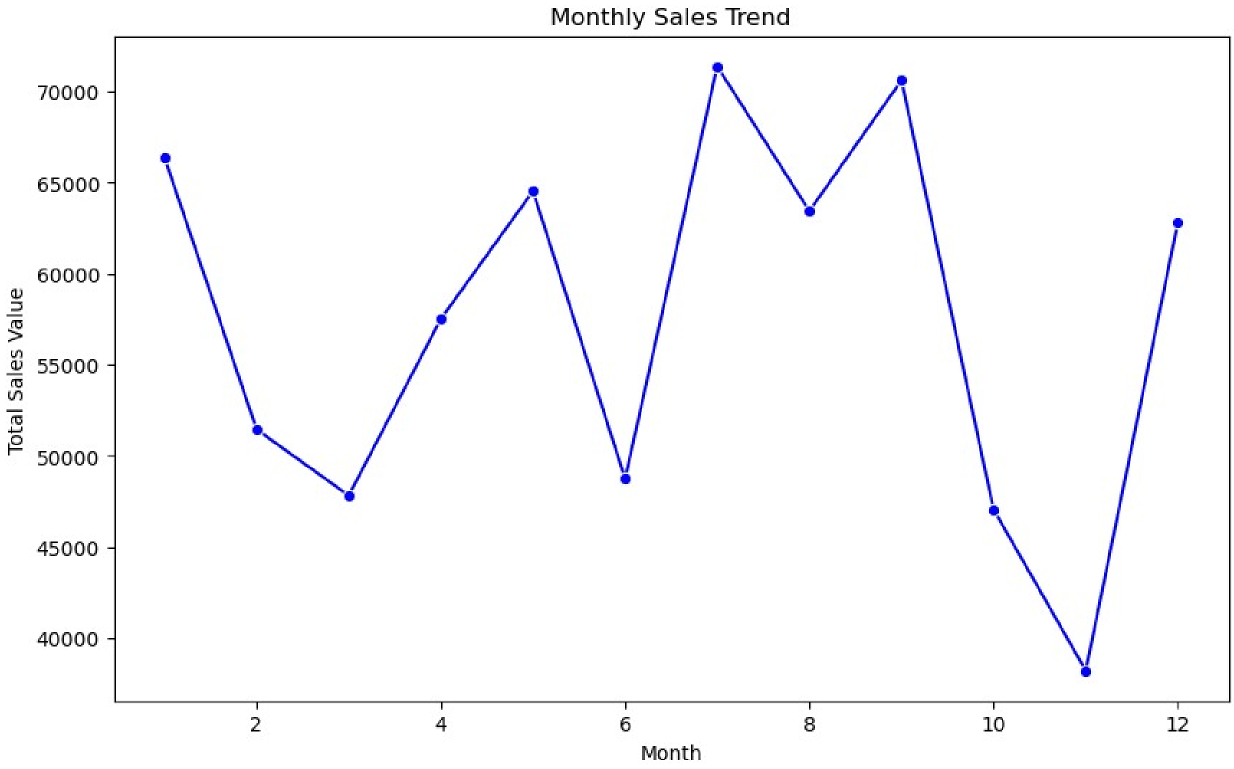


```
# Quantity across Categories
plt.figure(figsize=(12, 6))
sns.violinplot(x='Category', y='Quantity', data=combined_data,
palette='Set2')
plt.title('Quantity Distribution across Categories')
plt.xticks(rotation=45)
plt.show()
```

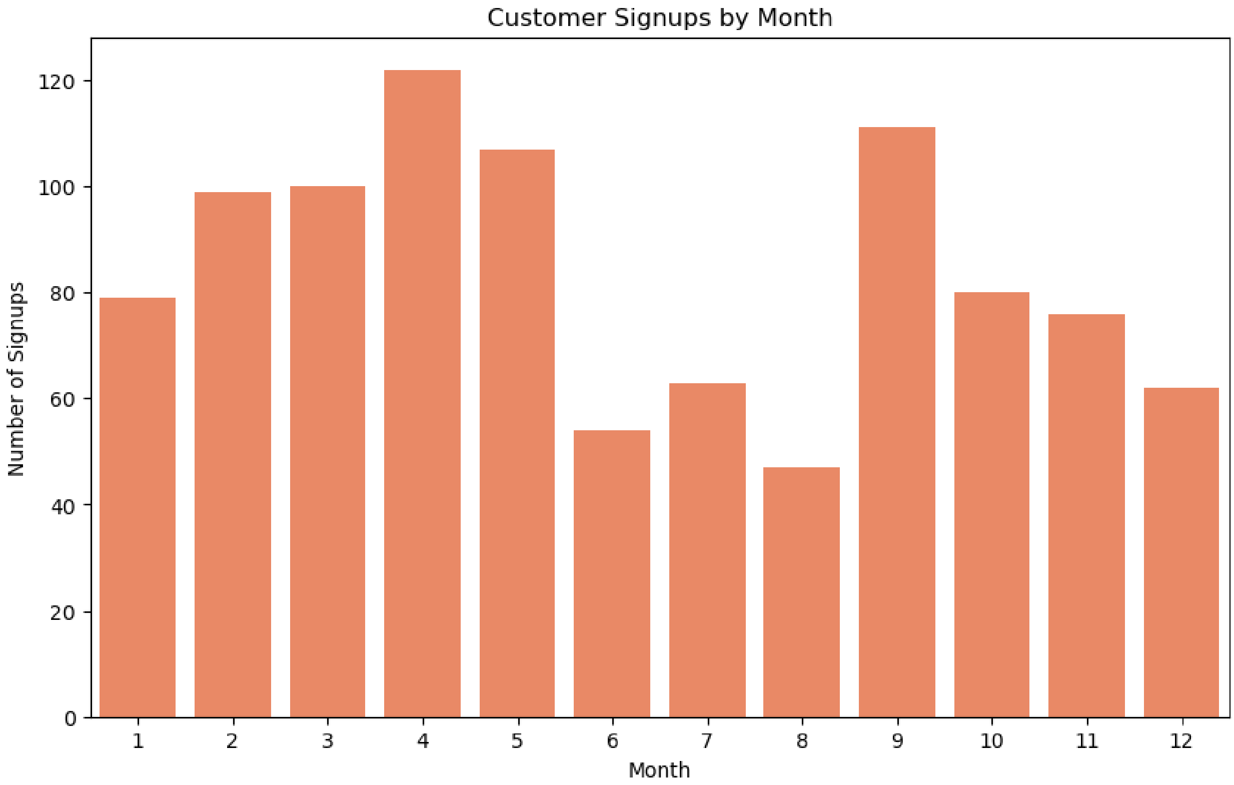


3. Time-Based Analysis

```
# Convert TransactionDate to datetime if not already
combined_data['TransactionDate'] =
pd.to_datetime(combined_data['TransactionDate'])

# Group data by Month
monthly_data = combined_data.groupby('Month')
['TotalValue'].sum().reset_index()

plt.figure(figsize=(10, 6))
sns.lineplot(x='Month', y='TotalValue', data=monthly_data, marker='o',
color='blue')
plt.title('Monthly Sales Trend')
plt.xlabel('Month')
plt.ylabel('Total Sales Value')
plt.show()
```

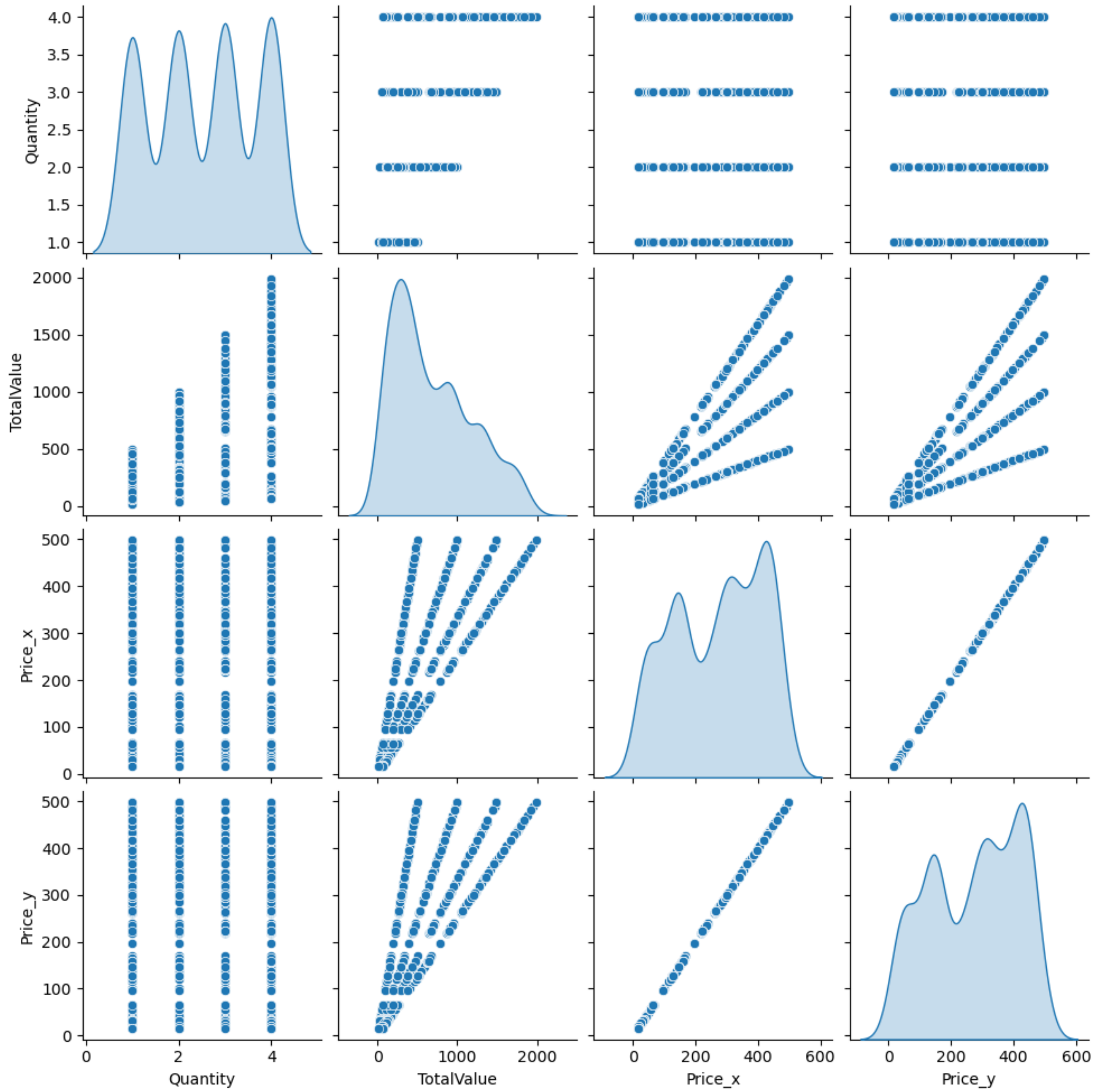


```
# Signup trends by Region
combined_data['SignupDate'] =
pd.to_datetime(combined_data['SignupDate'])
signup_trend =
combined_data.groupby(combined_data['SignupDate'].dt.month)
['CustomerID'].count().reset_index()

plt.figure(figsize=(10, 6))
sns.barplot(x='SignupDate', y='CustomerID', data=signup_trend,
color='coral')
plt.title('Customer Signups by Month')
plt.xlabel('Month')
plt.ylabel('Number of Signups')
plt.show()
```



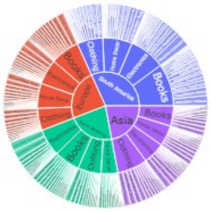
4.Advanced Visualizations

```
# Pair Plot for Numerical Variables
sns.pairplot(combined_data[['Quantity', 'TotalValue', 'Price_x',
'Price_y']], diag_kind='kde', palette='coolwarm')
plt.show()
```



```
import plotly.express as px

fig = px.sunburst(combined_data, path=['Region', 'Category',
'ProductName'], values='TotalValue', title='Sales Breakdown')
fig.show()
```

Sales Breakdown


South America
Clothing
Home Decor
Electronics
Books
Europe
Books
Electronics
Home Decor
Clothing
Asia
Books
Home Decor
Electronics
Clothing
North America
Electronics
Books
Clothing
Home Decor