```
import pandas as pd

customers = pd.read_csv("/content/Customers.csv")
products = pd.read_csv("/content/Products.csv")
transactions = pd.read_csv("/content/Transactions.csv")

customer_transactions = pd.merge(transactions, customers,
on="CustomerID", how="inner")

merged_data = pd.merge(customer_transactions, products,
on="ProductID", how="inner")

print(merged_data.head())

merged_data.to_csv("Merged_Dataset.csv", index=False)
```

```
  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
```

```
merged_data.isnull().sum()
```

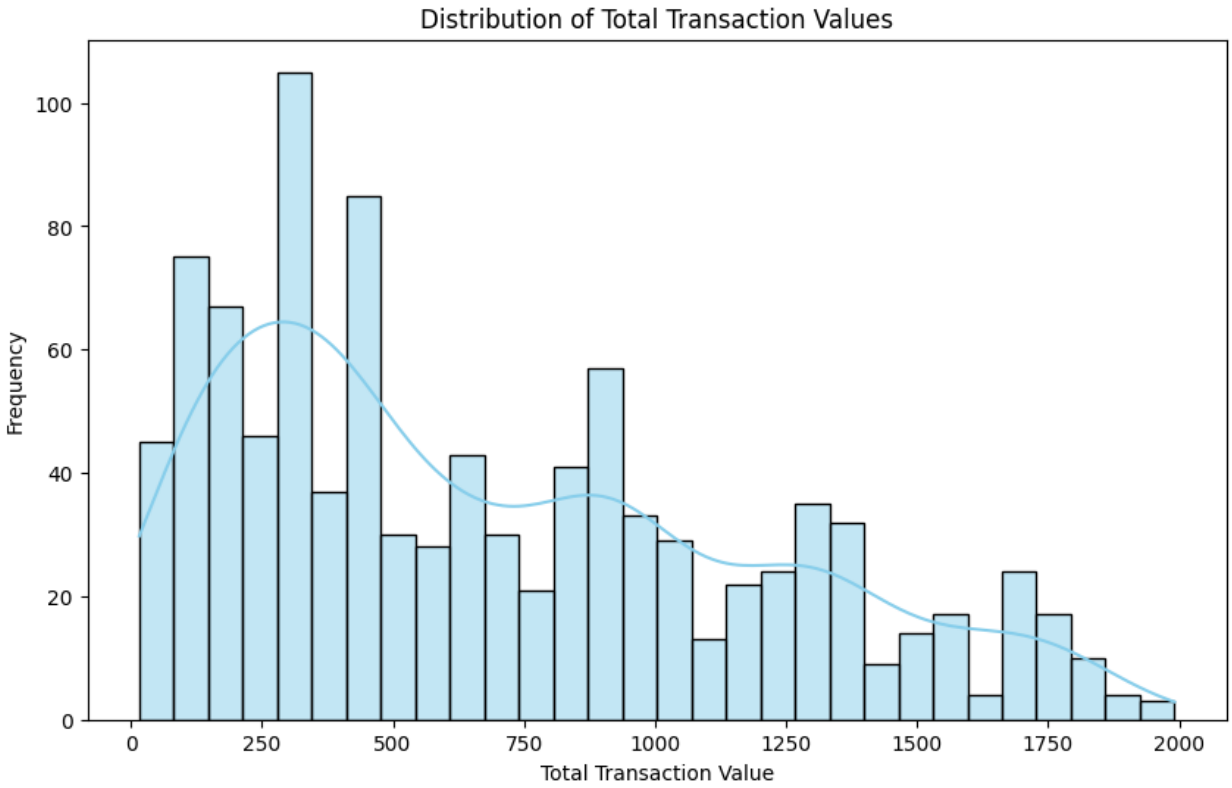
```
TransactionID      0
CustomerID         0
ProductID          0
TransactionDate    0
Quantity           0
TotalValue         0
Price_x            0
CustomerName       0
Region             0
SignupDate         0
ProductName        0
Category           0
```

```
Price_y             0
dtype: int64
```

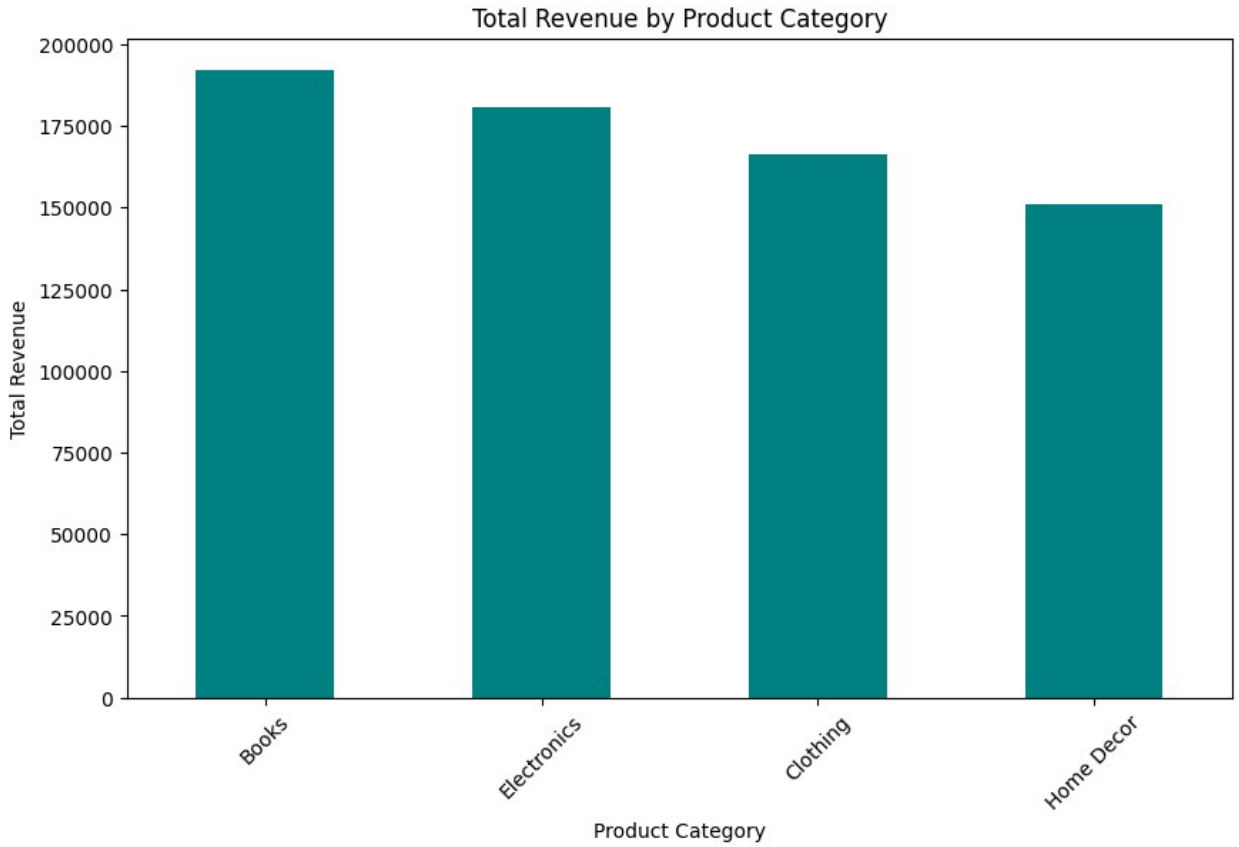
```
merged_data.describe()
```


```
{"summary":"{\n  \"name\": \"merged_data\",\n  \"rows\": 8,\n
\"fields\": [\n    {\n      \"column\": \"Quantity\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
352.66353426013046,\n        \"min\": 1.0,\n        \"max\": 1000.0,\n
\"num_unique_values\": 7,\n        \"samples\": [\n          1000.0,\n
2.537,\n          3.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"TotalValue\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 598.9454831884048,\n        \"min\":
16.08,\n        \"max\": 1991.04,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          689.9955600000001,\n          588.88,\n
1000.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Price_x\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 305.1609156198964,\n        \"min\": 16.08,\n
\"max\": 1000.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          272.55407,\n          299.93,\n
1000.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Price_y\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 305.1609156198964,\n        \"min\": 16.08,\n
\"max\": 1000.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          272.55407,\n          299.93,\n
1000.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

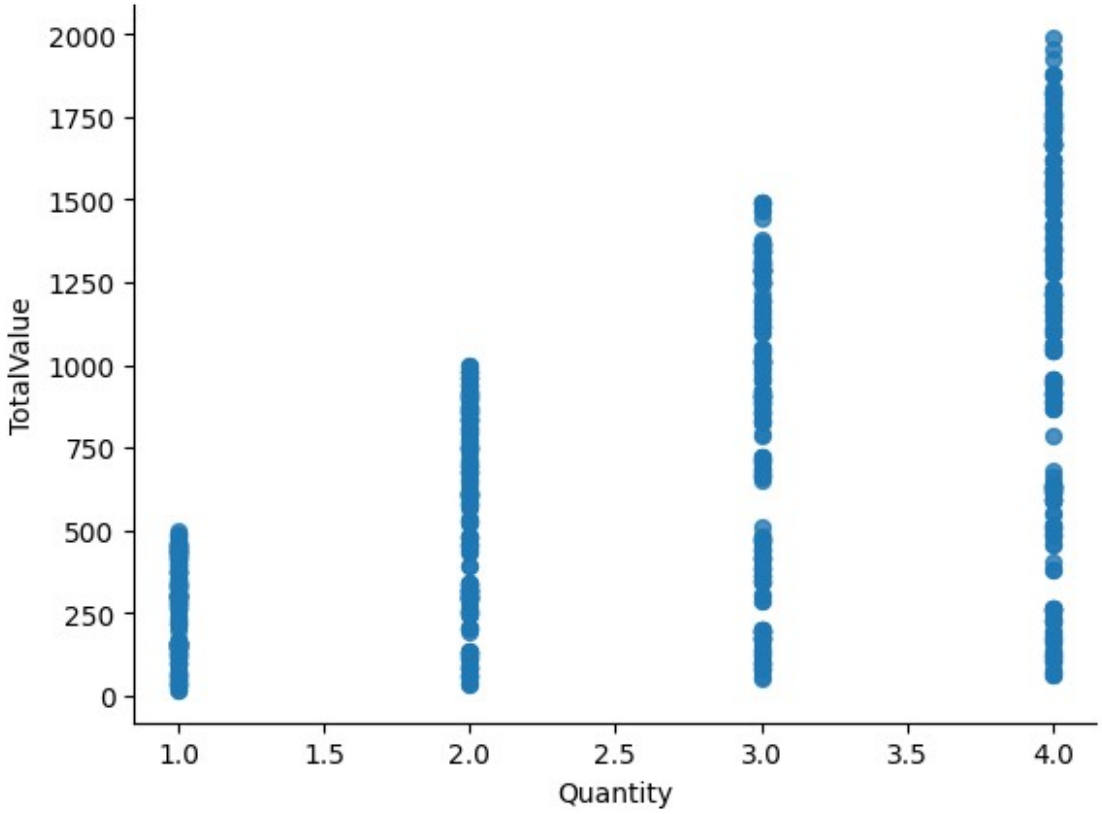

```
plt.figure(figsize=(10, 6))
sns.histplot(merged_data['TotalValue'], kde=True, bins=30,
color="skyblue")
plt.title("Distribution of Total Transaction Values")
plt.xlabel("Total Transaction Value")
plt.ylabel("Frequency")
plt.show()
```



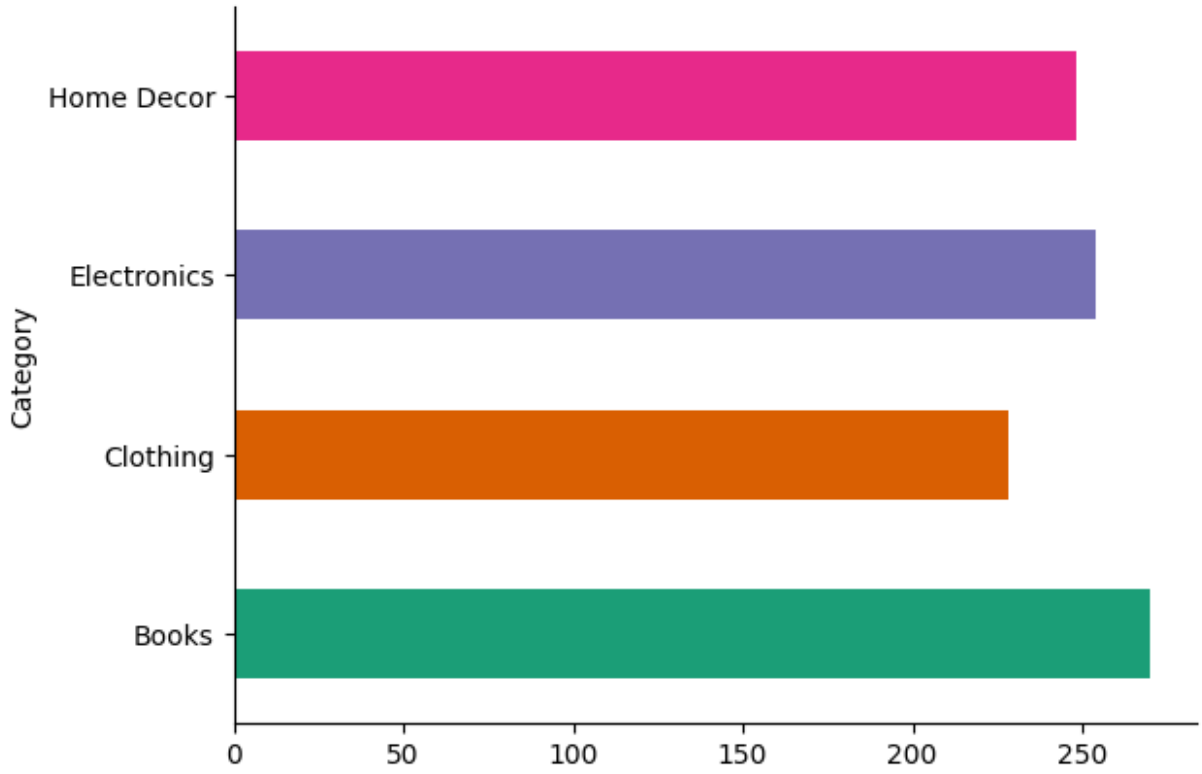

```
revenue_by_category = merged_data.groupby('Category')
['TotalValue'].sum().sort_values(ascending=False)
plt.figure(figsize=(10, 6))
revenue_by_category.plot(kind='bar', color="teal")
plt.title("Total Revenue by Product Category")
plt.xlabel("Product Category")
plt.ylabel("Total Revenue")
plt.xticks(rotation=45)
plt.show()
```



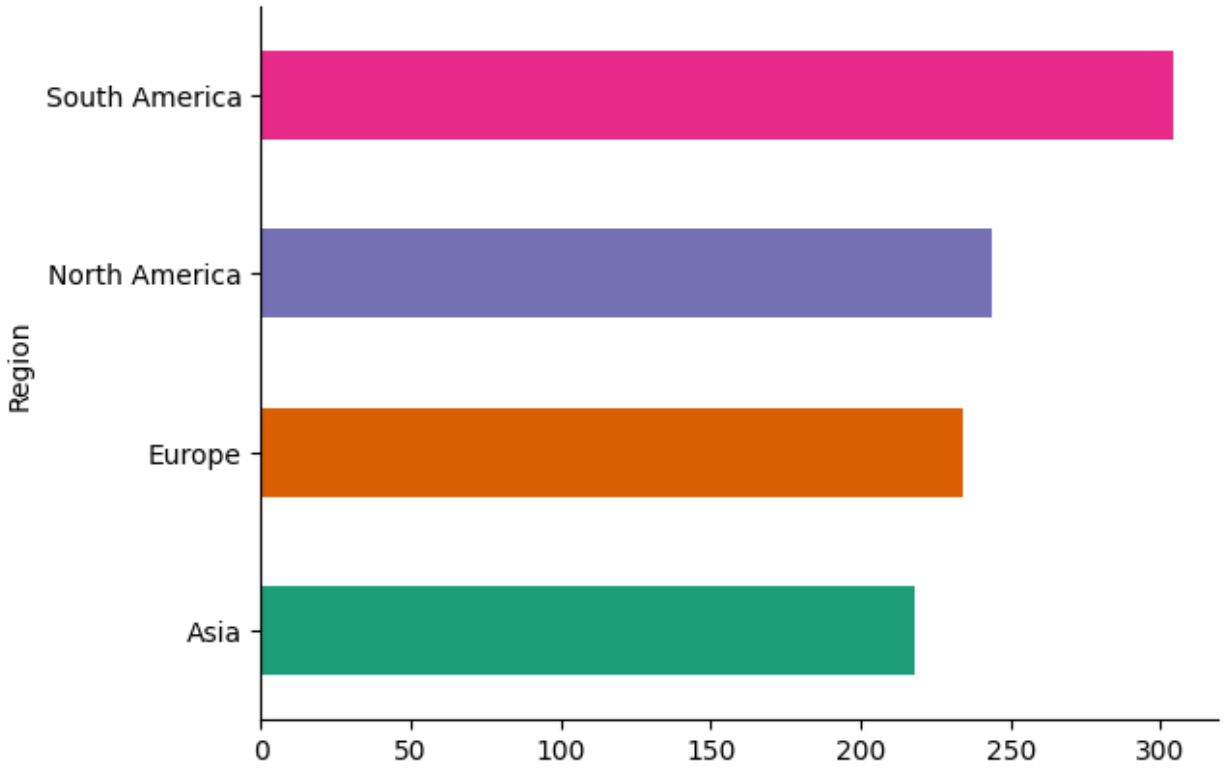

```
# @title Quantity vs TotalValue

from matplotlib import pyplot as plt
merged_data.plot(kind='scatter', x='Quantity', y='TotalValue', s=32,
alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```



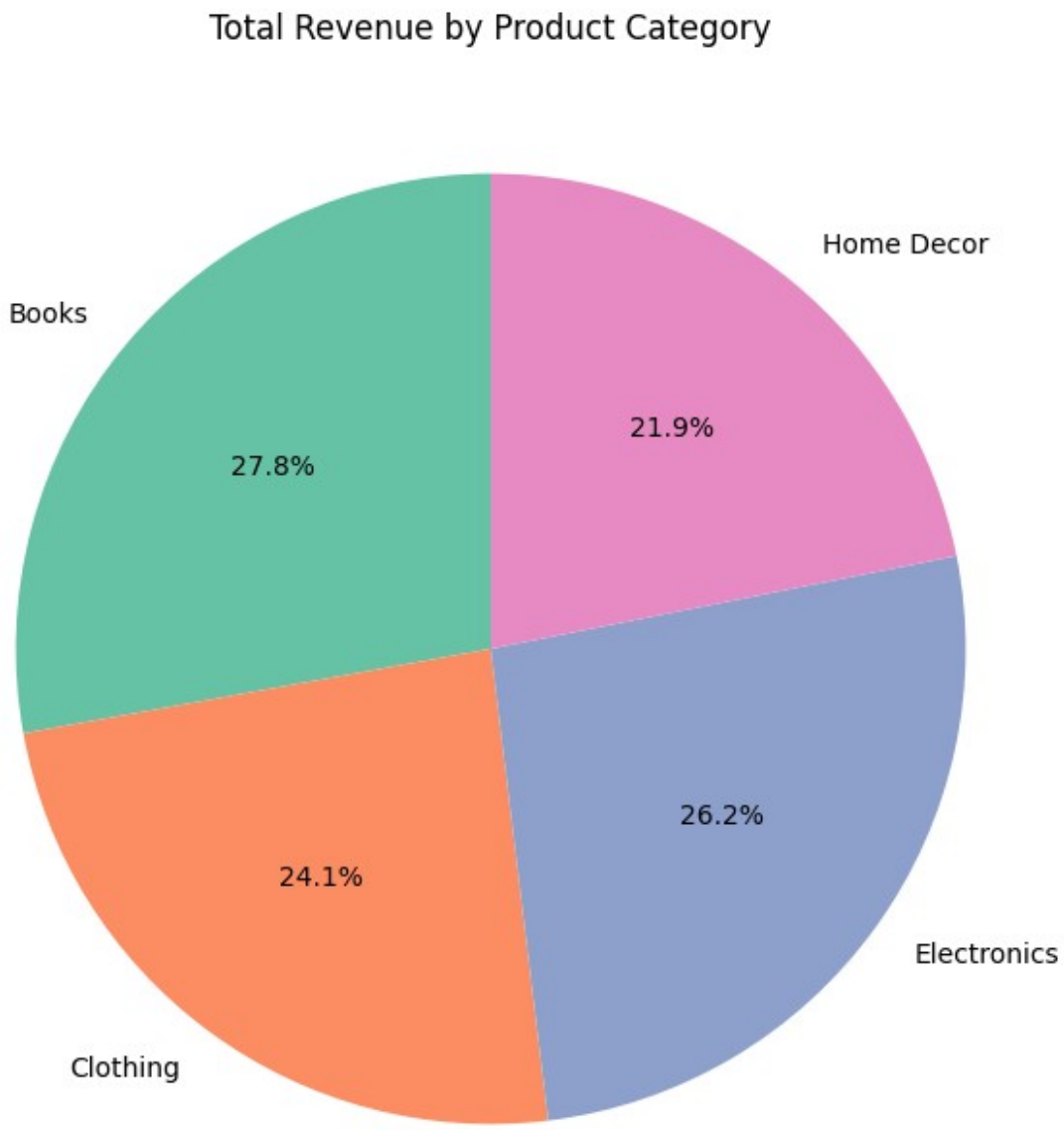

```
from matplotlib import pyplot as plt
import seaborn as sns
merged_data.groupby('Category').size().plot(kind='barh',
color=sns.palettes.mpl_palette('Dark2'))
plt.gca().spines[['top', 'right',]].set_visible(False)
```




```
from matplotlib import pyplot as plt
import seaborn as sns
merged_data.groupby('Region').size().plot(kind='barh',
color=sns.palettes.mpl_palette('Dark2'))
plt.gca().spines[['top', 'right',]].set_visible(False)
```




```
revenue_by_category = merged_data.groupby('Category')
['TotalValue'].sum()

# Plot total revenue by category in a pie chart
plt.figure(figsize=(8, 8))
revenue_by_category.plot(kind='pie', autopct='%1.1f%%', startangle=90,
colors=sns.color_palette('Set2'))
plt.title("Total Revenue by Product Category")
plt.ylabel("")  # Remove default ylabel
plt.show()
```



Insights We can Obtain From the Above EDA Process

- More Number of Transactions are in the range of 300-350.
- The Books sales provides more Revenue than other Category.
- Mostly the South American's is recorded in the given Dataset.
- Books and Clothing took more than 50% of total revenue.