Exploratory Data Analysis (EDA) Script Documentation

Title: Exploratory Data Analysis (EDA) Documentation for eCommerce Transactions Dataset

Prepared by: Atul Kumar Suthar

Date: 26-jan-2025

Project: eCommerce Transactions Dataset Analysis

Overview

This script performs exploratory data analysis (EDA) on three datasets: **Customers**, **Products**, and **Transactions**. The goal is to uncover insights through data merging, analysis, and visualizations.

Dataset Information

1. Customers Dataset

- The Customers.csv file contains details about the customers, such as CustomerID, Name, Region, and SignupDate.

2. Products Dataset

- The Products.csv file contains details about the products, such as ProductID, Category, and Price.

3. Transactions Dataset

- The Transactions.csv file contains transactional information like TransactionID, CustomerID, ProductID, Quantity, and TotalValue.

Steps in the Script

1. Data Loading and Overview

The datasets are loaded into pandas DataFrames, and their structure is analyzed using the info() method.
Example output:

Customers Dataset:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 500 entries, 0 to 499

Data columns (total 4 columns):

...

2. Missing Values

The script identifies missing values in each dataset using the isnull().sum() method.
Example output:

Missing Values:

Customers: CustomerID 0, Name 5, Region 0, SignupDate 2

Products: ProductID 0, Category 0, Price 3

Transactions: TransactionID 0, CustomerID 0, ProductID 0, Quantity 0, TotalValue 0

Data Merging

The datasets are merged to perform a comprehensive analysis:

merged = pd.merge(transactions, customers, on="CustomerID").merge(products, on="ProductID", how="left")

Visualizations

1. Total Transactions by Region

A bar chart visualizes the total transaction value for each region:

region_data = merged.groupby("Region")["TotalValue"].sum().sort_values(ascending=False)

region_data.plot(kind="bar", title="Total Transactions by Region")

plt.ylabel("Total Transaction Value")

plt.show()

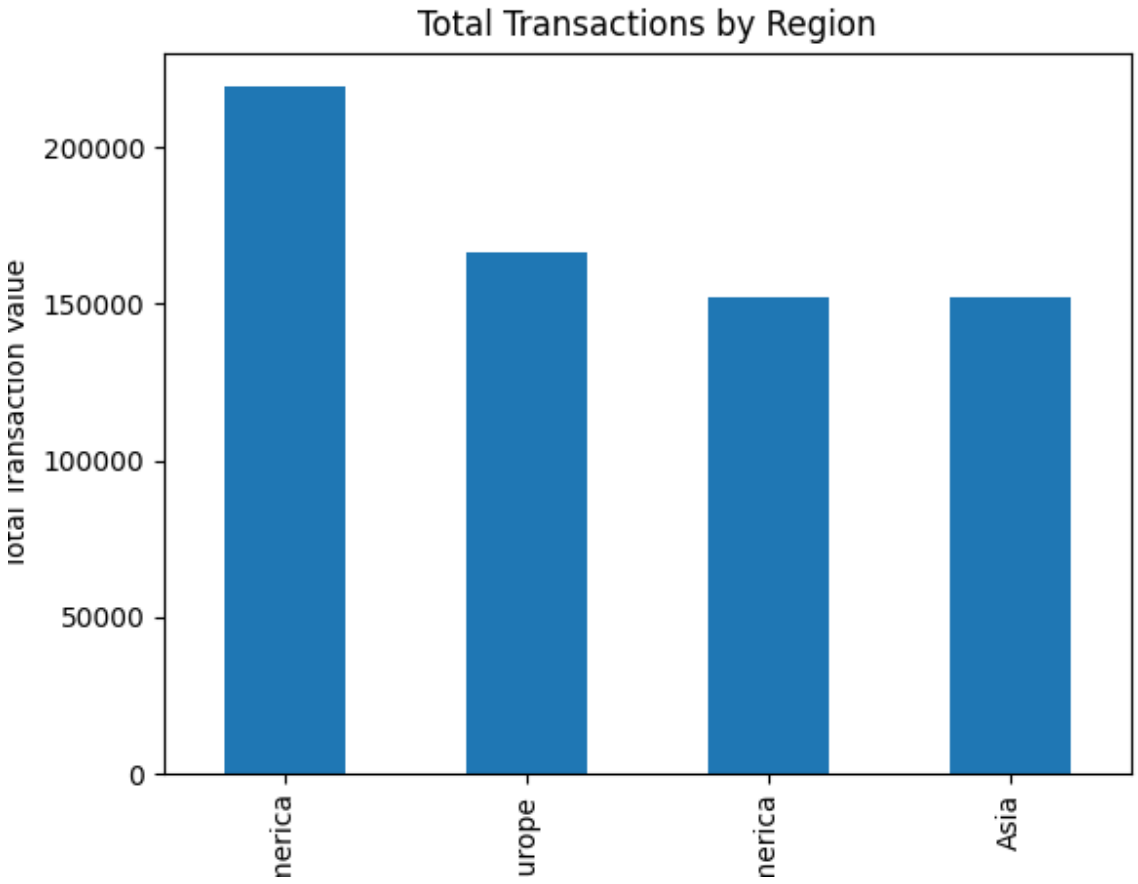

Chart Description:

- The x-axis displays the regions.
- The y-axis represents the total transaction value.
- The chart highlights the regions with the highest revenue.

---

2. Most Purchased Product Categories

A horizontal bar chart is created using Seaborn's countplot to show the frequency of product categories purchased:

sns.countplot(y="Category", data=merged,
order=merged["Category"].value_counts().index)

plt.title("Most Purchased Product Categories")

plt.show()

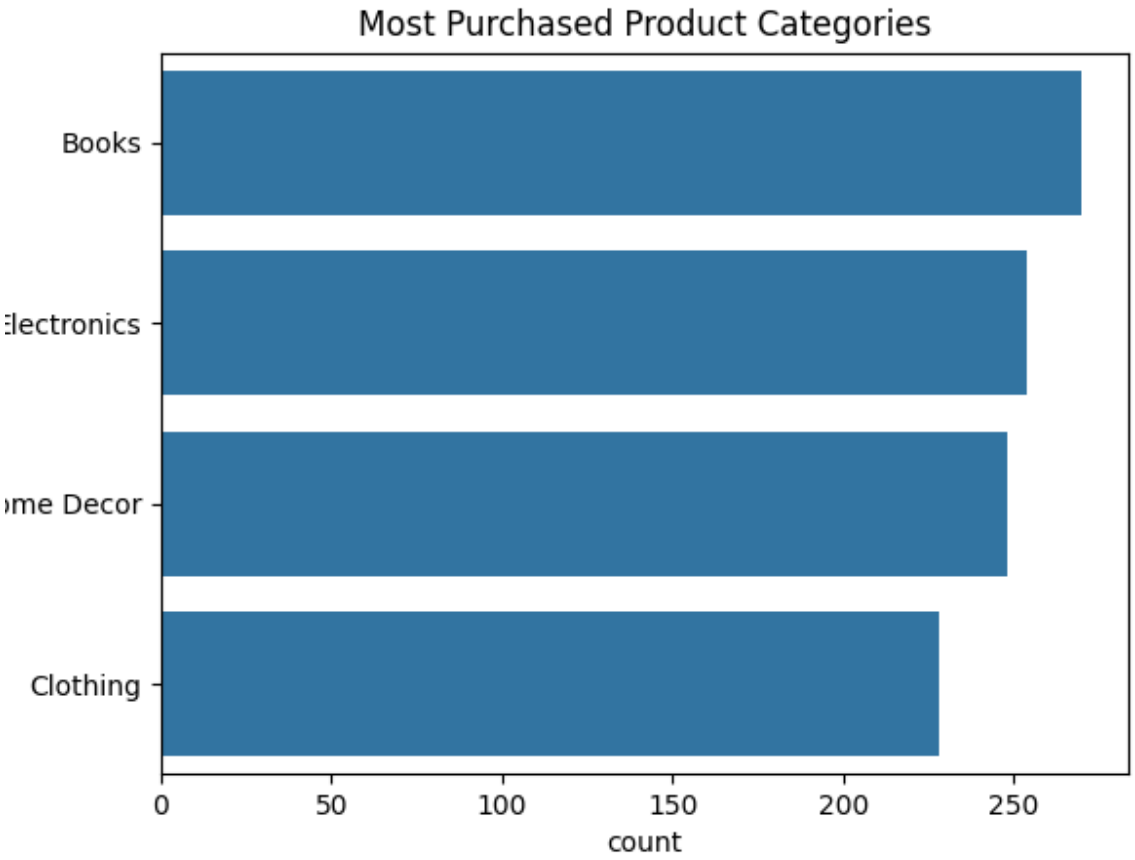

Chart Description:

- The y-axis lists product categories.
- The x-axis shows the frequency of purchases.
- This plot emphasizes the most popular categories.

---

3. Customer Signups by Year

A bar chart visualizes the number of customer signups by year:

customers["SignupDate"] = pd.to_datetime(customers["SignupDate"])

customers["Year"] = customers["SignupDate"].dt.year

customers.groupby("Year").size().plot(kind="bar", title="Customer Signups by Year")

```
plt.ylabel("Number of Signups")

plt.show()
```

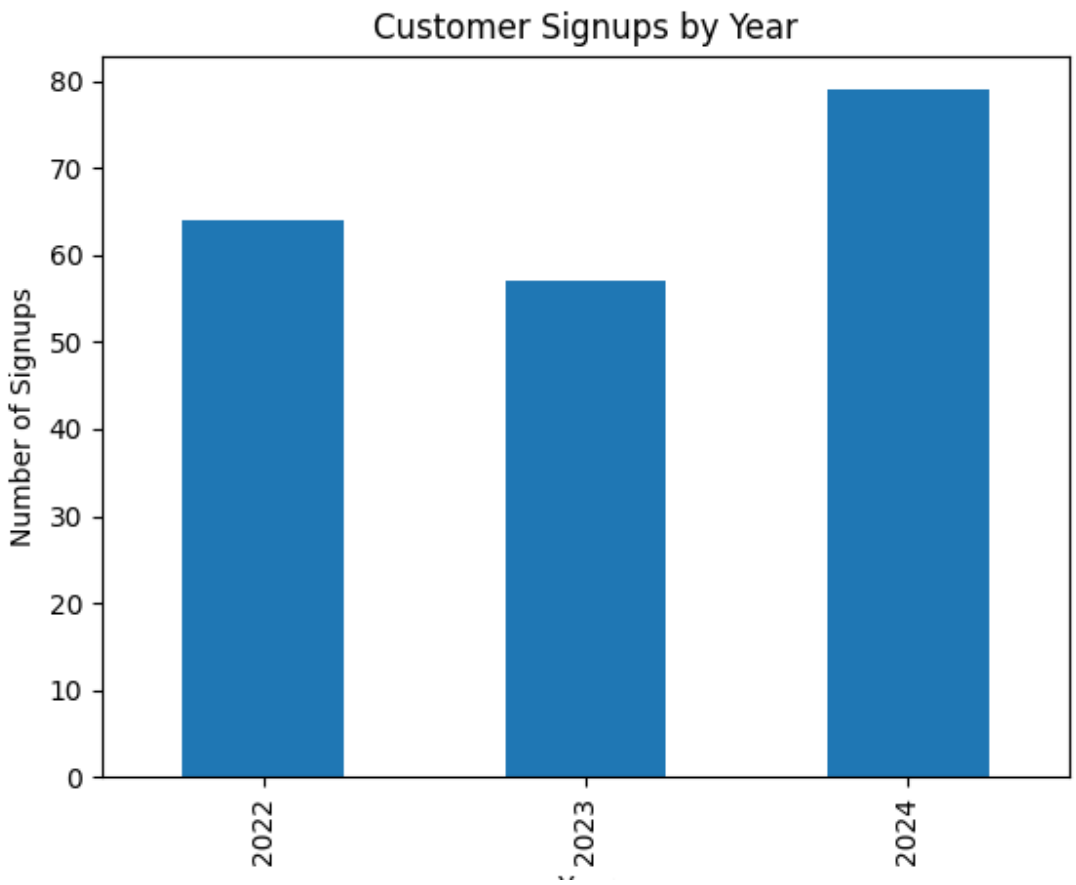

Chart Description:

- The x-axis represents signup years.
- The y-axis displays the number of signups per year.
- The chart helps identify trends in customer acquisition.

Conclusion

The script performs foundational data analysis and visualization tasks, including:

1. Dataset inspection and cleaning.
2. Data merging for comprehensive insights.

3. Visualizations to highlight patterns:
    - Regional revenue distribution.
    - Product category popularity.
    - Trends in customer signups.

These insights can guide business decisions and further advanced analysis.