**How to Perform Exploratory Data Analysis (EDA): A Comprehensive Tutorial with Examples**

**Introduction**

Exploratory Data Analysis (EDA) is a crucial step in any data science project. It allows us to gain valuable insights, identify patterns, and understand the underlying structure of our data. In this tutorial, we'll walk through the process of conducting EDA using various data visualization techniques and statistical tools. We'll be working with a dataset containing information about online retail transactions, exploring and visualizing the data to extract meaningful insights.

Dataset Description

For this tutorial, we'll use a dataset called "OnlineRetail.csv," which contains details of transactions made by customers in an online retail store. The dataset consists of the following columns:

* InvoiceNo: Unique identifier for each transaction.
* StockCode: Unique identifier for each product.
* Description: Description of the product.
* Quantity: The quantity of products purchased.
* UnitPrice: The unit price of each product.
* CustomerID: Unique identifier for each customer.
* Country: The country where the customer resides.

Before we dive into the code, ensure you have the necessary libraries installed. We'll be using Pandas, Matplotlib, Seaborn, and NumPy for this tutorial.

Step 1: Importing the necessary libraries and loading the dataset

***import*** *pandas* ***as*** *pd*

***import*** *matplotlib.pyplot* ***as*** *plt*

***import*** *seaborn* ***as*** *sns*

***import*** *numpy* ***as*** *np*

*# Load the dataset*

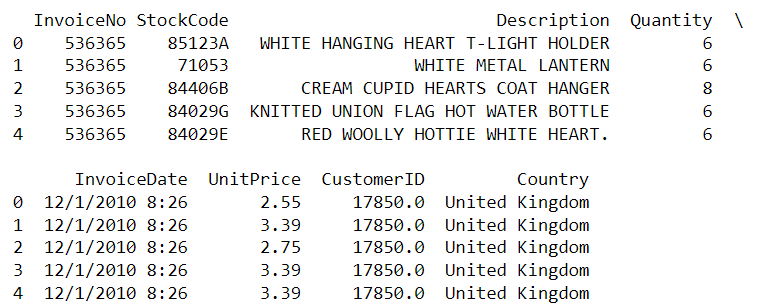
*df* ***=*** *pd.read\_csv("OnlineRetail.csv")*

Step 2: Understanding the data

Let's start by getting an overview of our dataset to understand its structure and check for missing values.

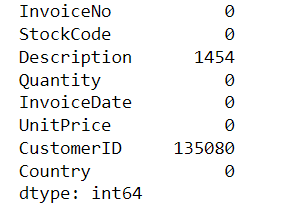
*# View the first few rows of the dataset*

print(df.head())



*# Check for missing values*

print(df.isnull().sum())

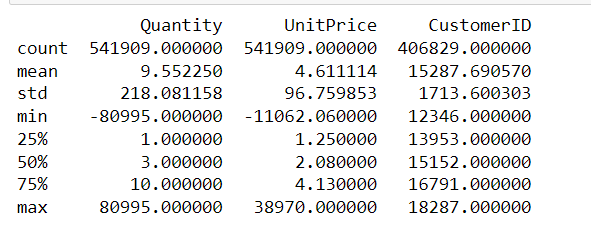


Step 3: Summary statistics

Obtain summary statistics of the numerical columns in the dataset.

*# Summary statistics*

print(df.describe())



Step 4: Data Cleaning

Before proceeding with EDA, we need to handle missing values and outliers. For simplicity, let's remove rows with missing values and filter out negative quantities and unit prices.

*# Drop rows with missing values*

df.dropna(inplace**=True**)

*# Filter out negative quantities and unit prices*

df **=** df[(df["Quantity"] **>** 0) **&** (df["UnitPrice"] **>** 0)]

Step 5: Data Visualization

Now, let's visualize the data to gain insights and identify patterns.

5.1. Sales by Country

*# Visualize sales by country using a bar plot*

plt.figure(figsize**=**(12, 6))

country\_sales **=** df.groupby("Country")["Quantity"].sum().sort\_values(ascending**=False**)

sns.barplot(x**=**country\_sales.index, y**=**country\_sales.values)

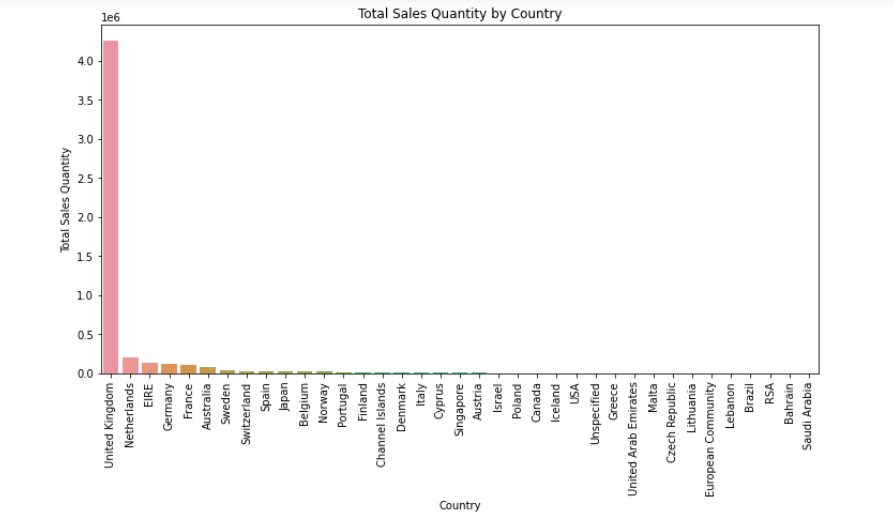
plt.xticks(rotation**=**90)

plt.xlabel("Country")

plt.ylabel("Total Sales Quantity")

plt.title("Total Sales Quantity by Country")

plt.show()



Using Numpy to take the natural log of the data to make it sparser.

*country\_sales\_df = pd.DataFrame(country\_sales)*

*country\_sales\_df['Quantity'] = np.log(country\_sales\_df['Quantity'])*

*# Visualize sales by country using a bar plot*

*plt.figure(figsize=(12, 6))*

*country\_sales = df.groupby("Country")["Quantity"].sum().sort\_values(ascending=False)*

*sns.barplot(x=country\_sales\_df.index, y=country\_sales\_df.Quantity)*

*plt.xticks(rotation=90)*

*plt.xlabel("Country")*

*plt.ylabel("Total Sales Quantity")*

*plt.title("Total Sales Quantity by Country")*

*plt.show()*



5.2. Distribution of Quantity and Unit Price

*# Distribution plot for Quantity*

plt.figure(figsize**=**(10, 6))

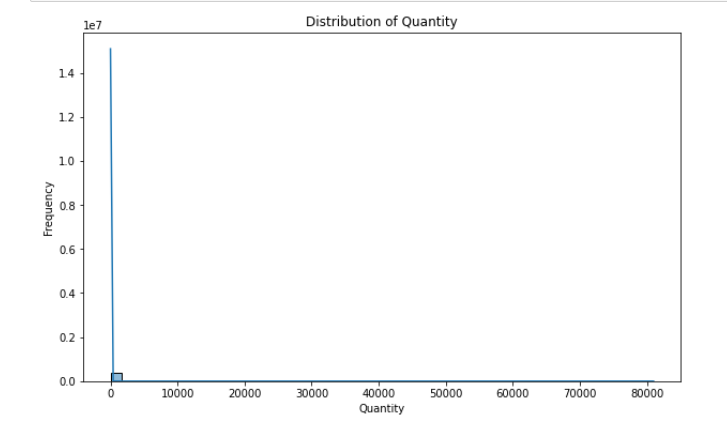
sns.histplot(df["Quantity"], bins**=**50, kde**=True**)

plt.xlabel("Quantity")

plt.ylabel("Frequency")

plt.title("Distribution of Quantity")

plt.show()



Using Numpy to take the natural log of the data to make it sparser.

df['Quantity'] = np.log(df['Quantity'])

# Distribution plot for Quantity

plt.figure(figsize=(10, 6))

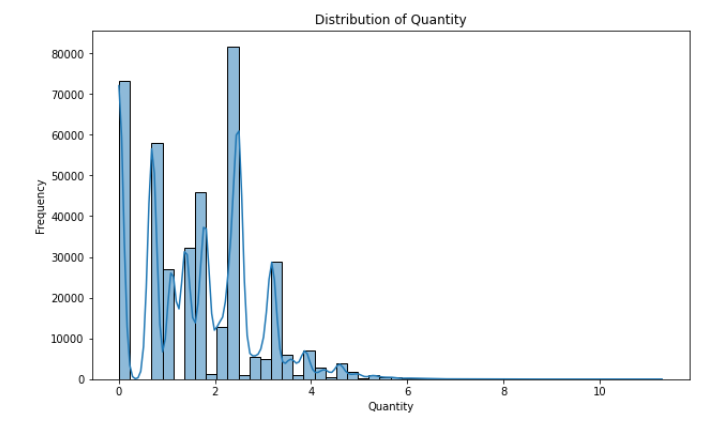
sns.histplot(df["Quantity"], bins=50, kde=True)

plt.xlabel("Quantity")

plt.ylabel("Frequency")

plt.title("Distribution of Quantity")

plt.show()



*# Distribution plot for UnitPrice*

plt.figure(figsize**=**(10, 6))

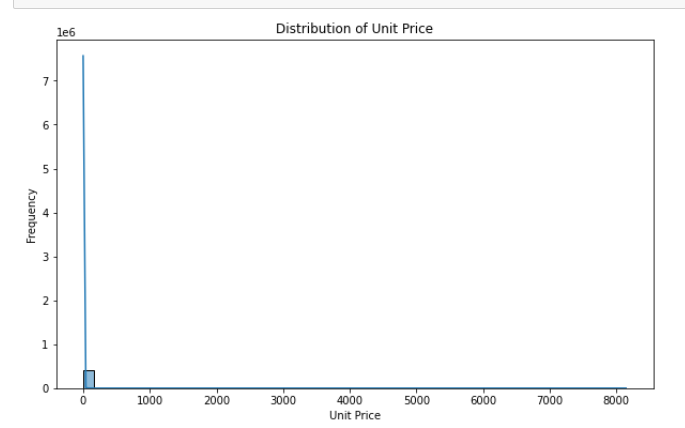
sns.histplot(df["UnitPrice"], bins**=**50, kde**=True**)

plt.xlabel("Unit Price")

plt.ylabel("Frequency")

plt.title("Distribution of Unit Price")

plt.show()



Using Numpy to take the natural log of the data to make it sparser.

df['UnitPrice'] = np.log(df['UnitPrice'])

# Distribution plot for UnitPrice

plt.figure(figsize=(10, 6))

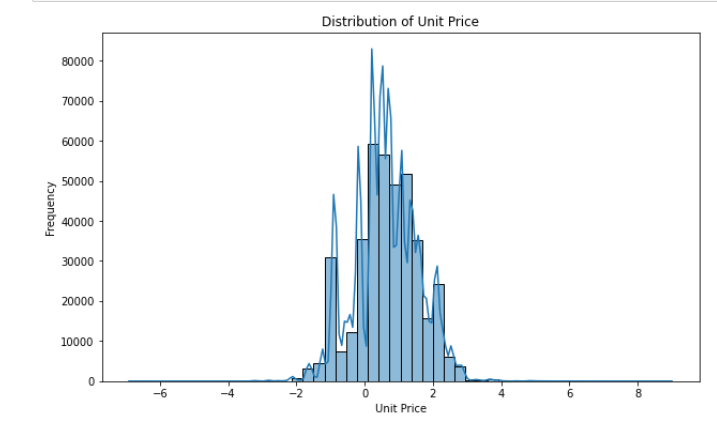
sns.histplot(df["UnitPrice"], bins=50, kde=True)

plt.xlabel("Unit Price")

plt.ylabel("Frequency")

plt.title("Distribution of Unit Price")

plt.show()



5.3. Monthly Sales Trend

*# Extracting the month from the InvoiceDate column*

df["InvoiceDate"] **=** pd.to\_datetime (df["InvoiceDate"])

df["Month"] **=** df["InvoiceDate"].dt.to\_period("M")

*# Visualize monthly sales trend using a line plot*

monthly\_sales **=** df.groupby("Month")["Quantity"].sum()

plt.figure(figsize**=**(12, 6))

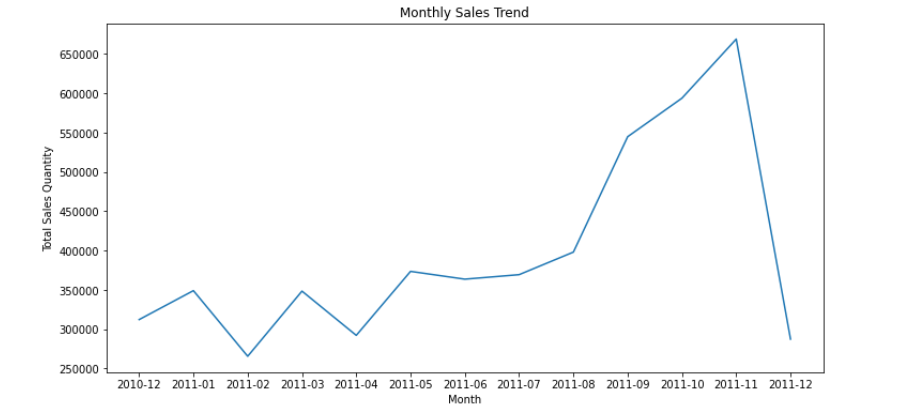
sns.lineplot(x**=**monthly\_sales.index, y**=**monthly\_sales.values)

plt.xlabel("Month")

plt.ylabel("Total Sales Quantity")

plt.title("Monthly Sales Trend")

plt.show()



Conclusion

In this tutorial, we explored the world of Exploratory Data Analysis (EDA) using Python and various data visualization techniques. We started by loading and understanding the dataset, then performed data cleaning to ensure the data's quality. Finally, we visualized the data to gain insights into sales by country, the distribution of quantity and unit price, and the monthly sales trend.

EDA is a crucial step in the data science workflow, as it helps us identify patterns, outliers, and relationships within the data, which lays the foundation for making informed decisions and building predictive models.

Remember that EDA is an iterative process, and there are countless ways to explore and visualize data. The techniques and examples covered here are just the beginning of what you can do with EDA. As you dive deeper into data science, you'll encounter more complex datasets and discover even more powerful ways to gain insights from them. Happy exploring!