Phase-3

Data Visualization

Date	18 October 2023
Team ID	Proj-212168-Team-2
Project Name	Market Basket Insights
Maximum marks	

Data visualization is a powerful tool for gaining insights from market basket analysis, which is often used in retail to understand the relationships between products that customers purchase together. You can use libraries like Python's Matplotlib, Seaborn for creating data visualizations.

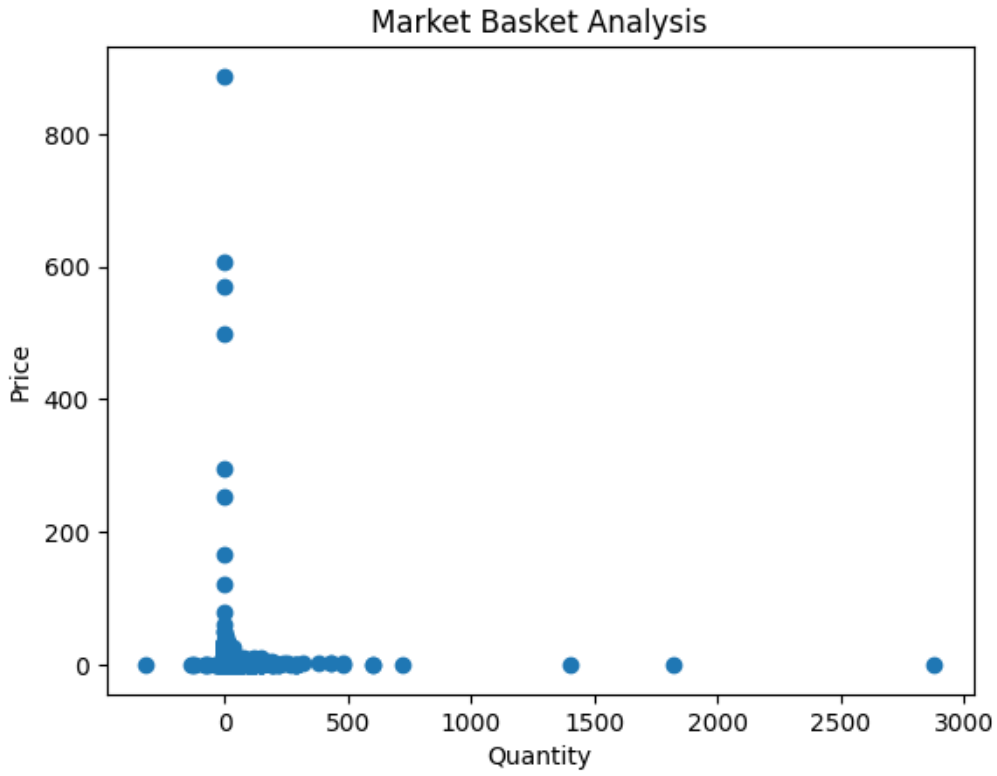
```
# Import necessary libraries

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

#import dataset

datasets=pd.read_csv('dataset.csv')

# Create a DataFrame from the data

df = pd.DataFrame(datasets)
```

Scatter plot:

Use plt.scatter() to create a scatter plot. Set a title, labels for the x and y axes, and a legend using plt.title(), plt.xlabel(), plt.ylabel(), and plt.legend().

Display the grid lines with plt.grid(True) and then use plt.show() to display the plot.

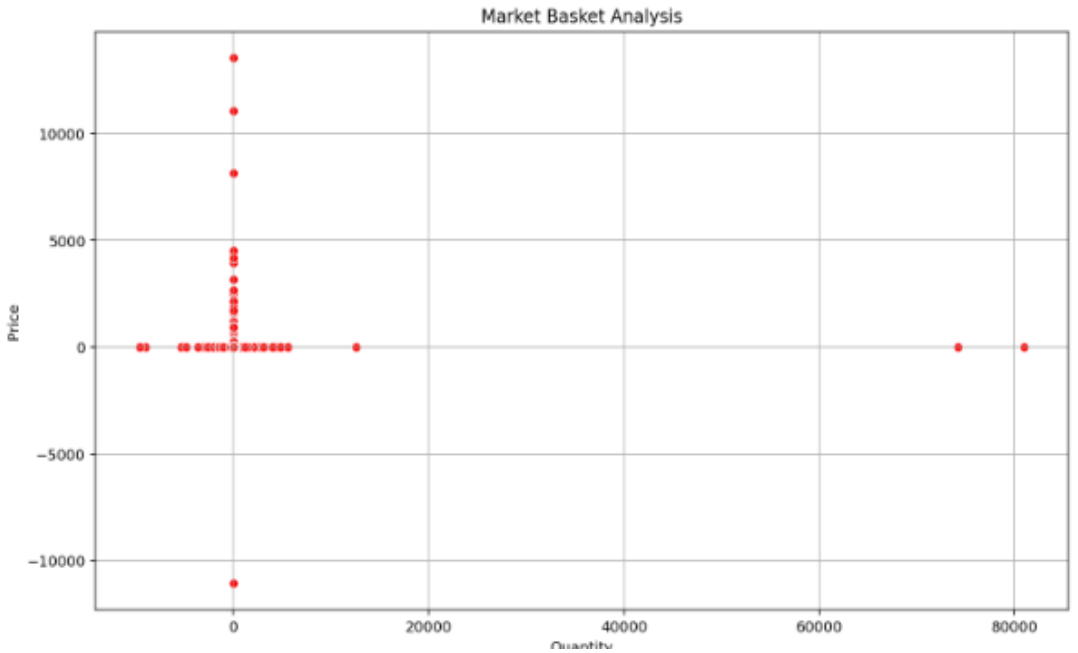
```
df1= pd.DataFrame(datasets)
x=df1['Quantity']
y=df1['Price']
plt.scatter(x,y)
plt.xlabel('Quantity')
plt.ylabel('Price')
plt.title('Market Basket Analysis')
plt.show()
```



Use sns.scatterplot() to create the scatter plot. You can specify the color using the color parameter. You don't need to manually set up grid lines with Seaborn as it provides a more visually appealing default style.

In [5]:
```
x=df1['Quantity']
y=df1['Price']
sns.scatterplot(x=x, y=y, color='red')
plt.xlabel('Quantity')
plt.ylabel('Price')
plt.title('Market Basket Analysis')
plt.grid(True)
plt.show()
```

Out[5]:

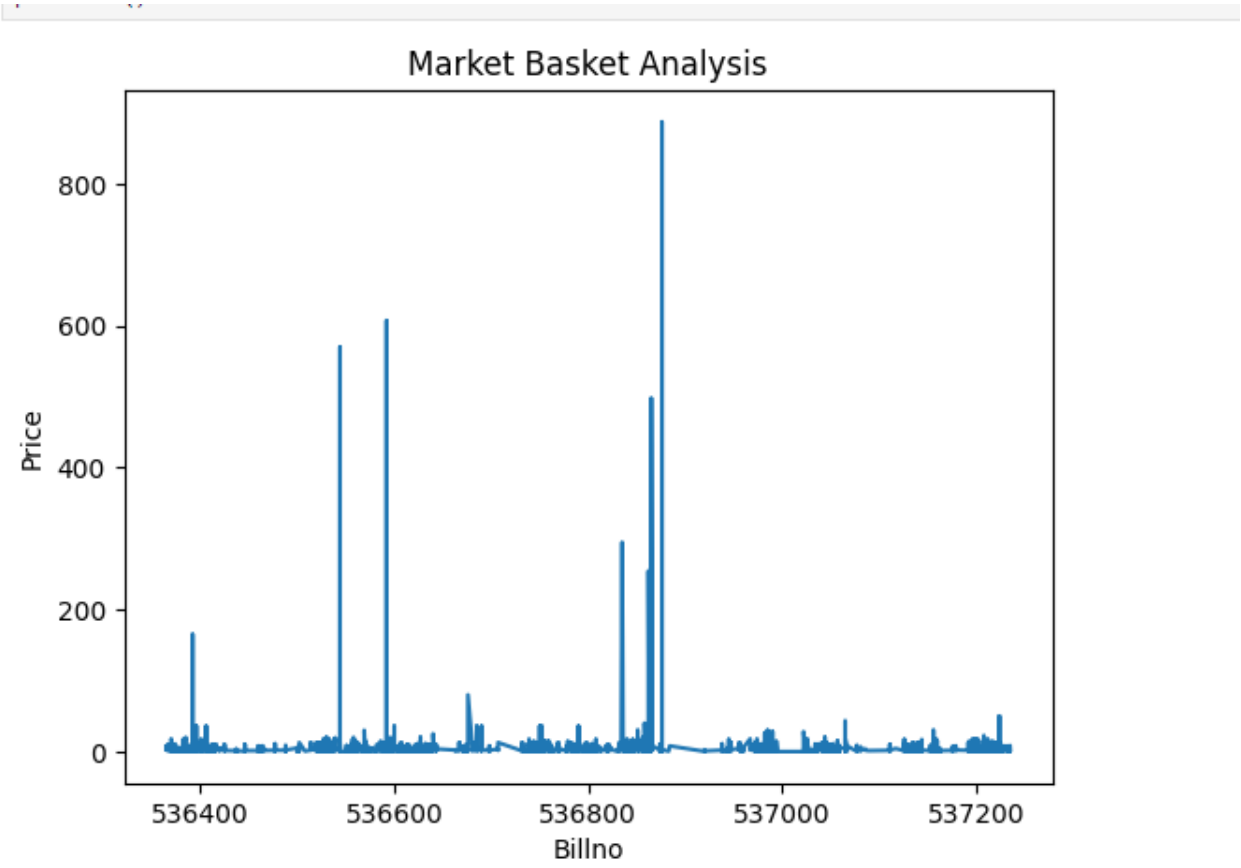


Line plot:

Use plt.plot() to create a scatter plot. Set a title, labels for the x and y axes, and a legend using plt.title(), plt.xlabel(), plt.ylabel().

Use plt.show() to display the plot.

```
df1= pd.DataFrame(datasets)
x=df1['BillNo']
y=df1['CustomerID']
plt.bar(x,y)
plt.xlabel('Billno')
plt.ylabel('CustomerID')
plt.title('Market Basket Analysis')
plt.show()
```

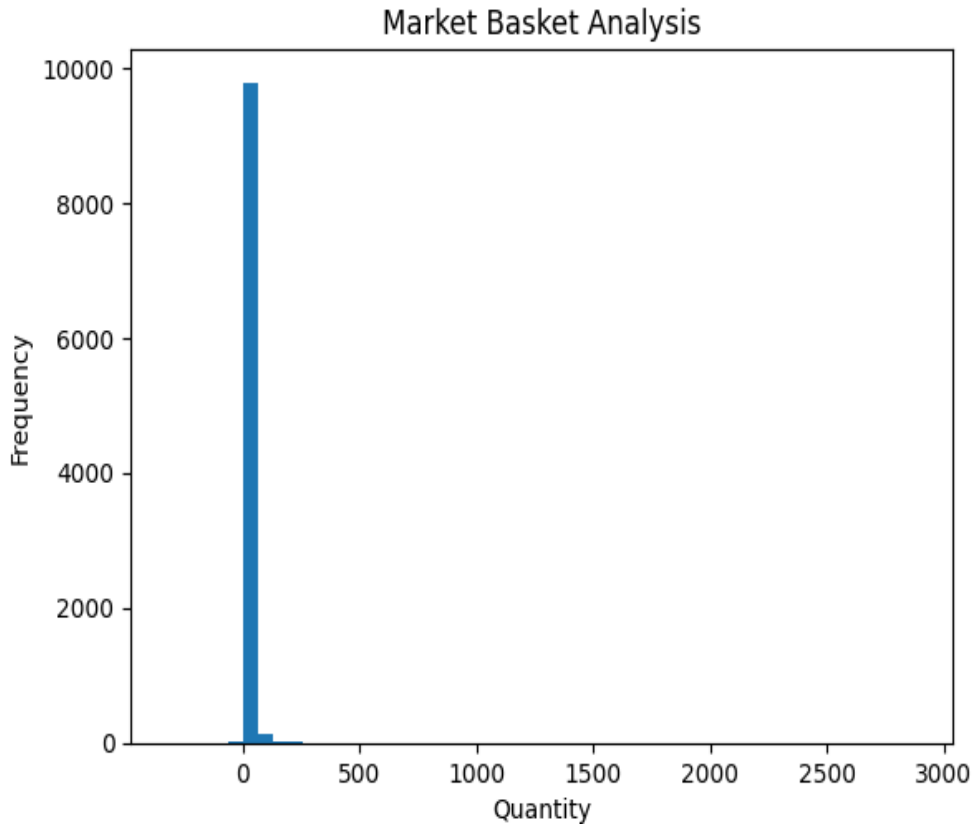


Histogram:

Histogram in Python is a common data visualization technique, and use libraries like Matplotlib and Seaborn to easily generate histograms.

Use plt.hist() to create the histogram. Specify the number of bins using the bins parameter. The edgecolor parameter sets the color of the bin edges, and the color parameter sets the color of the bars. Set a title, labels for the x and y axes, and display the plot using plt.title(), plt.xlabel(), and plt.ylabel(), followed by plt.show().

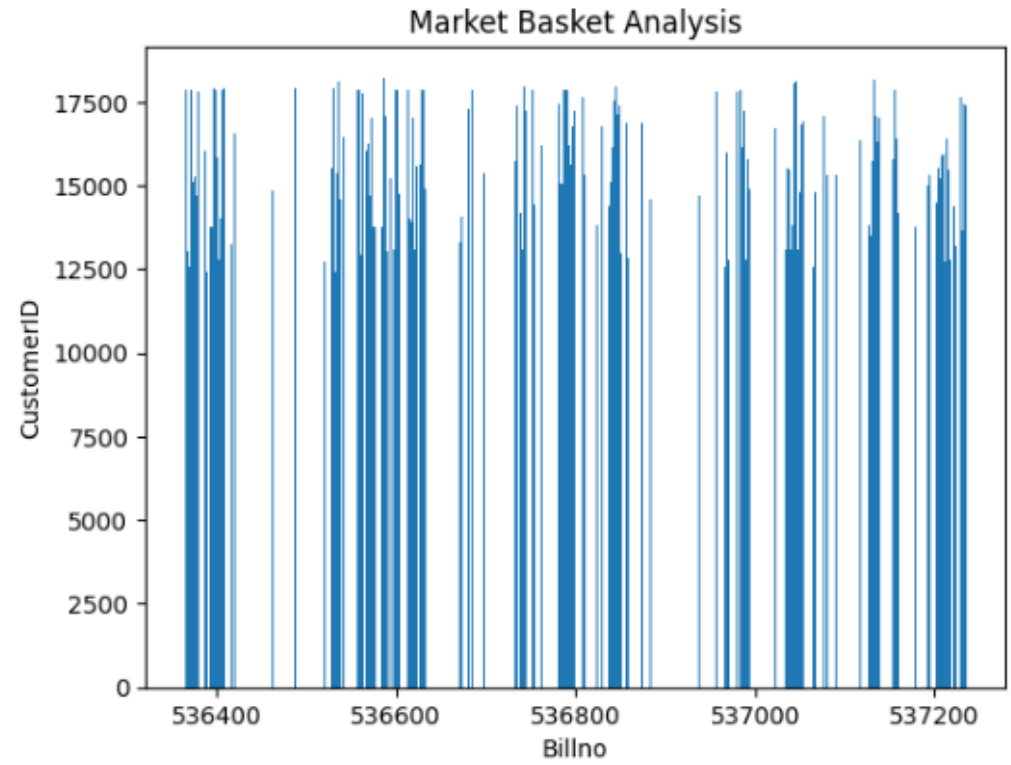
```
In [7]: x=df1['Quantity']
        plt.hist(x,bins=50,edgecolor='k', color='blue')
        plt.xlabel('Quantity')
        plt.ylabel('Frequency')
        plt.title('Market Basket Analysis')
        plt.show()
```



Barplot:

Creating a barplot in Python can be done using libraries like Matplotlib and Seaborn. Here we use a Matplotlib. Use plt.boxplot() to create the boxplot. Set a title and a label for the y-axis using plt.title() and plt.ylabel(), respectively. Display the plot using plt.show().

```
df1= pd.DataFrame(datasets)
x=df1['BillNo']
y=df1['CustomerID']
plt.bar(x,y)
plt.xlabel('Billno')
plt.ylabel('CustomerID')
plt.title('Market Basket Analysis')
plt.show()
```



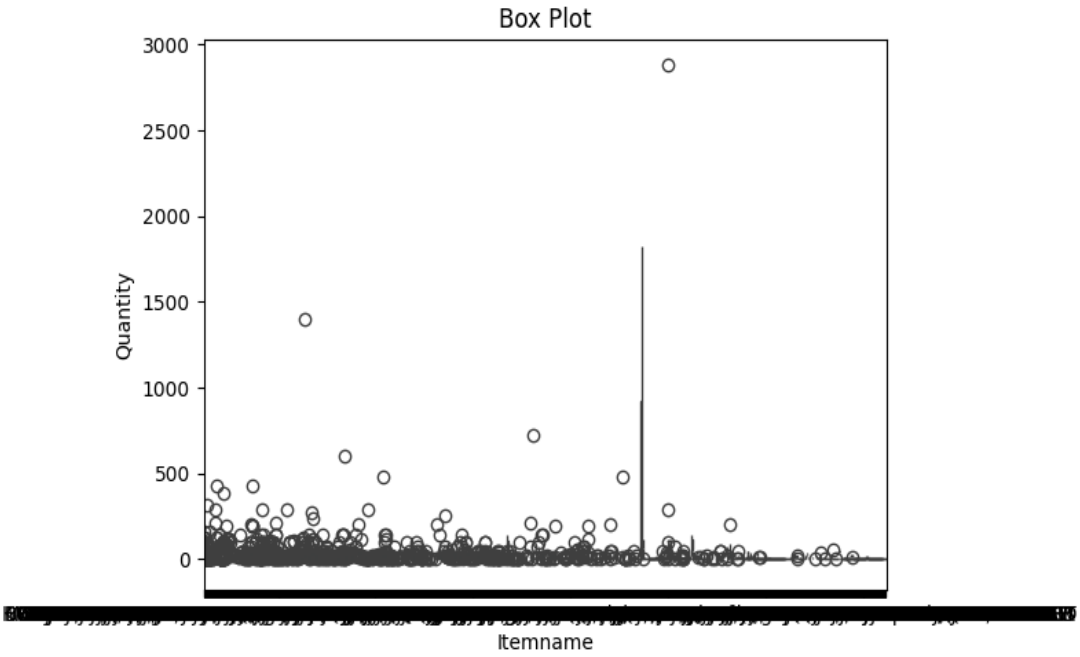
Boxplot:

Boxplots are a great way to visualize the distribution and central tendencies of a dataset. Creating a boxplot in Python can be done using libraries like Matplotlib and Seaborn.

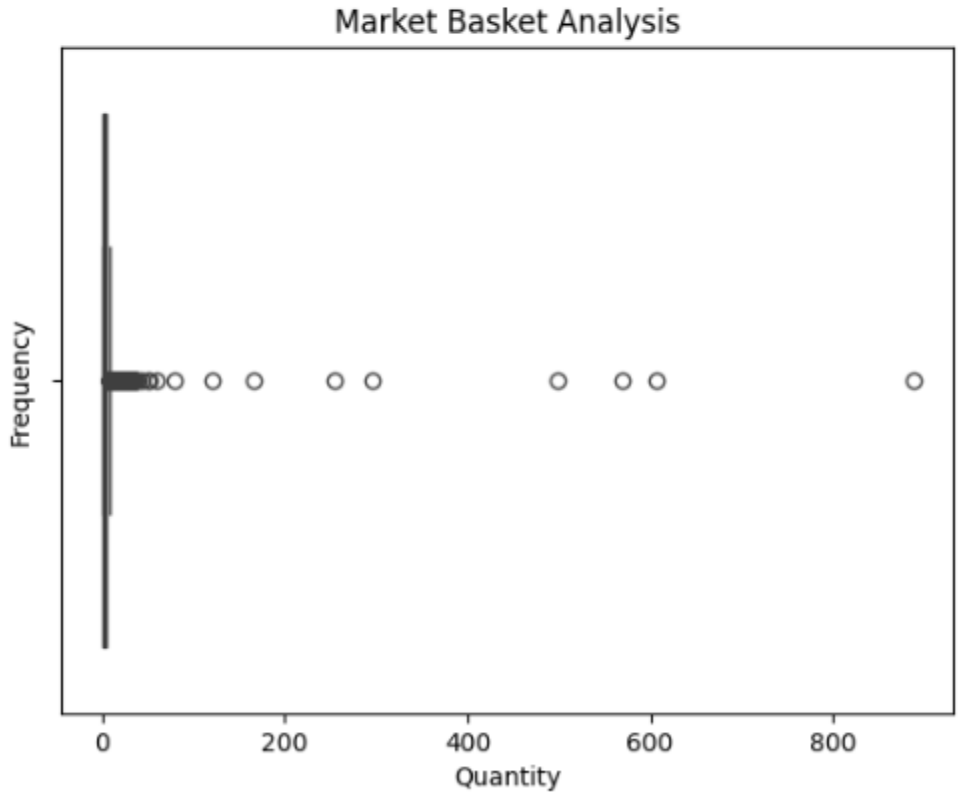
Use plt.boxplot() to create the boxplot. Set a title and a label for the y-axis using plt.title() and plt.ylabel(), respectively. Display the plot using plt.show().

Use sns.boxplot() to create the boxplot.

```
sns.boxplot(x='Itemname', y='Quantity', data=datasets)
plt.xlabel('Itemname')
plt.ylabel('Quantity')
plt.title('Box Plot')
plt.show()
```



```
]:
sns.boxplot(datasets,x='Price')
plt.xlabel('Quantity')
plt.ylabel('Frequency')
plt.title('Market Basket Analysis')
plt.show()
```

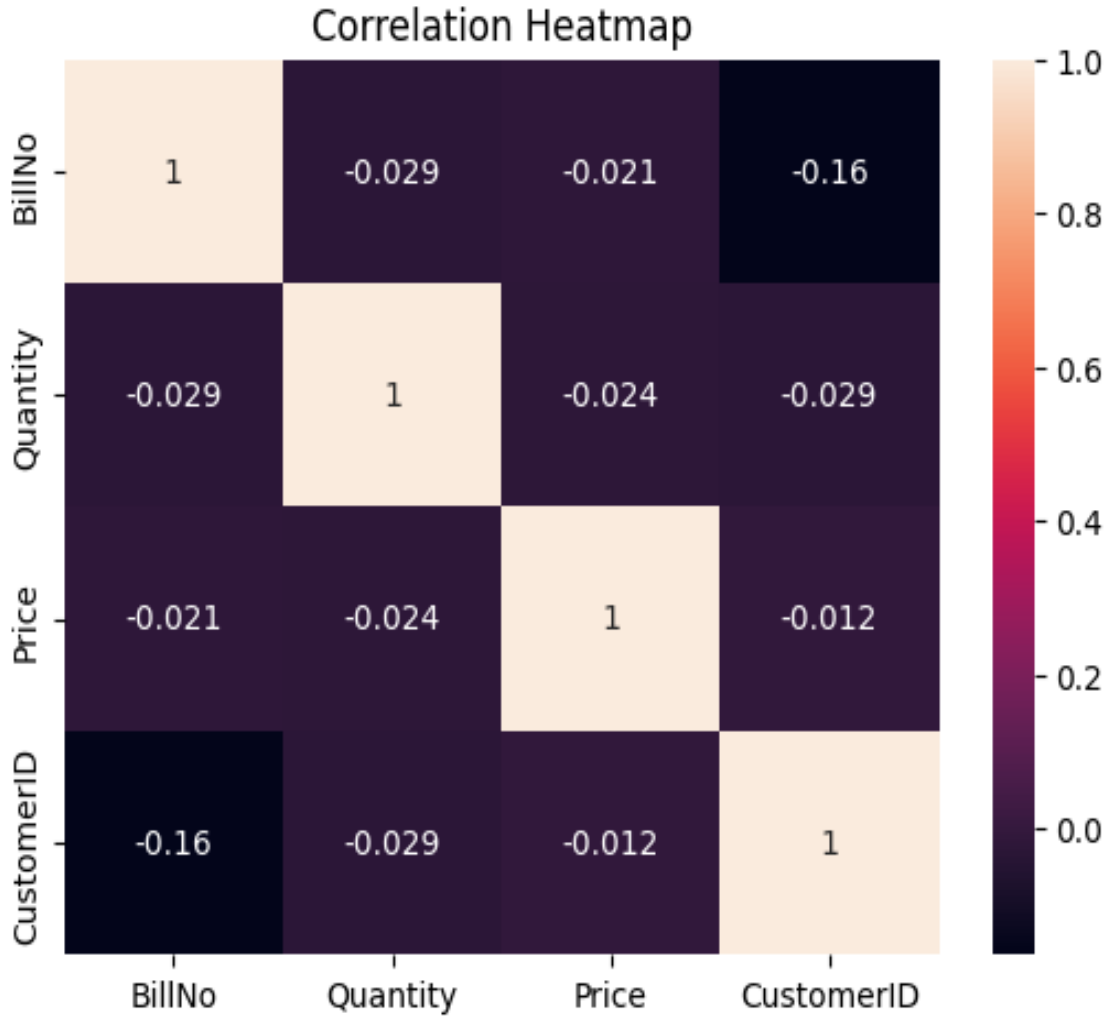


Heatmap:

A heatmap is a graphical representation of data where individual values are represented as colors. It is a way to visualize data in a matrix or grid format. Heatmaps are commonly used to depict data in various fields, including data analysis, statistics, and data visualization.

The heatmap is created using sns.heatmap(), and the annot=True parameter adds values to the cells for better visualization. Display the heatmap using plt.show().

```
correlation = datasets.corr()
sns.heatmap(correlation, annot=True)
plt.title('Correlation Heatmap')
plt.show()
```



Pairplot:

A pairplot is a type of data visualization that shows pairwise relationships between variables in a dataset.

Create a pairplot using sns.pairplot(). The hue parameter is set to "Country" to color the data points by the Country in the dataset.

Use plt.show( ) to display the plot.

```
sns.pairplot(datasets, hue='Country')
plt.title('Pair Plot')
plt.show()
```