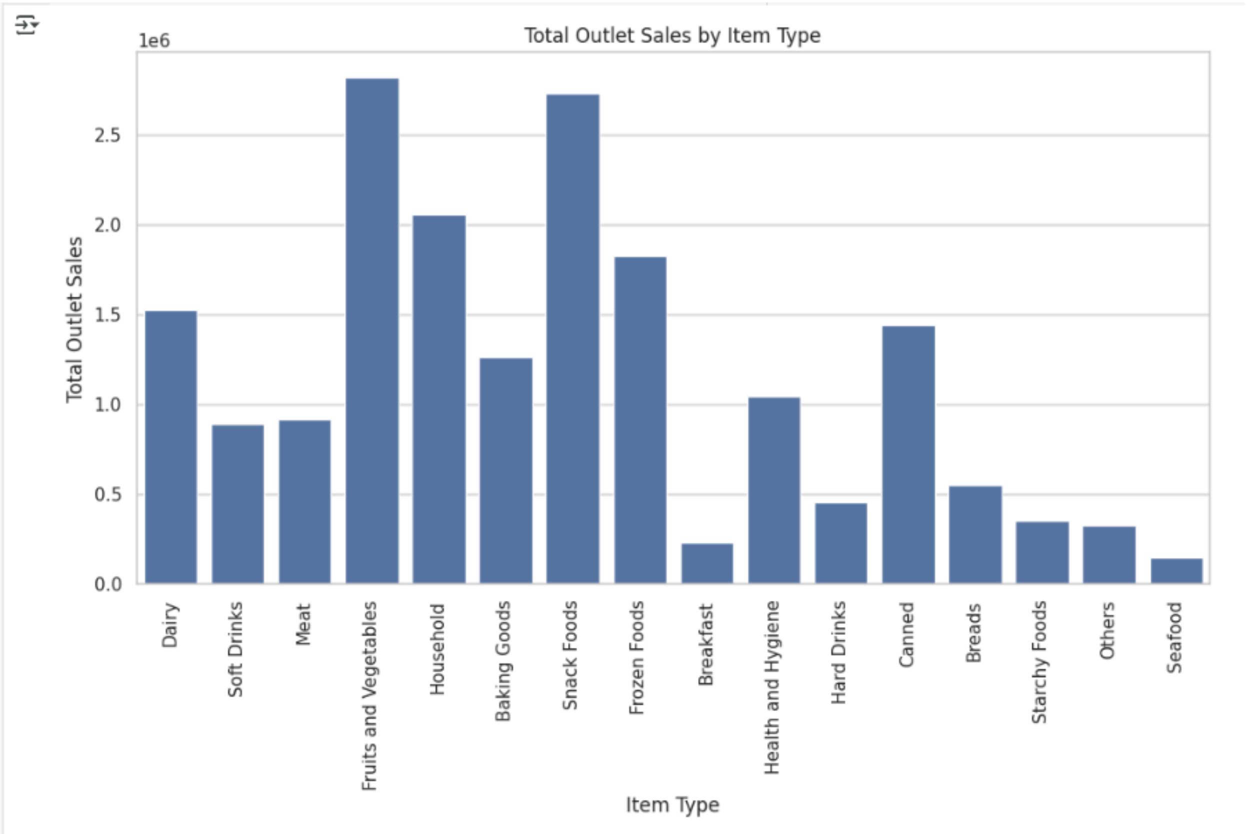
**Aim**: Data Visualization/ Exploratory data Analysis using Matplotlib and Seaborn.

**Theory**:
Data visualization and exploratory data analysis (EDA) using Matplotlib and Seaborn help uncover patterns, trends, and relationships within data. Matplotlib is a flexible, low-level library for creating static, animated, and interactive plots, while Seaborn is built on top of Matplotlib and provides a high-level interface for visually appealing statistical graphics. EDA involves techniques like histograms, scatter plots, box plots, and heatmaps to understand data distributions, detect outliers, and identify correlations

**Matplotlib :**
It is a powerful and versatile Python library for data visualization, built around a hierarchical structure of Figures and Axes. It provides a comprehensive set of plotting functions, including line plots, scatter plots, bar charts, and histograms. Users can extensively customize plots by adjusting colors, linestyles, markers, and grid lines, as well as incorporating LaTeX-style expressions for labels and titles.

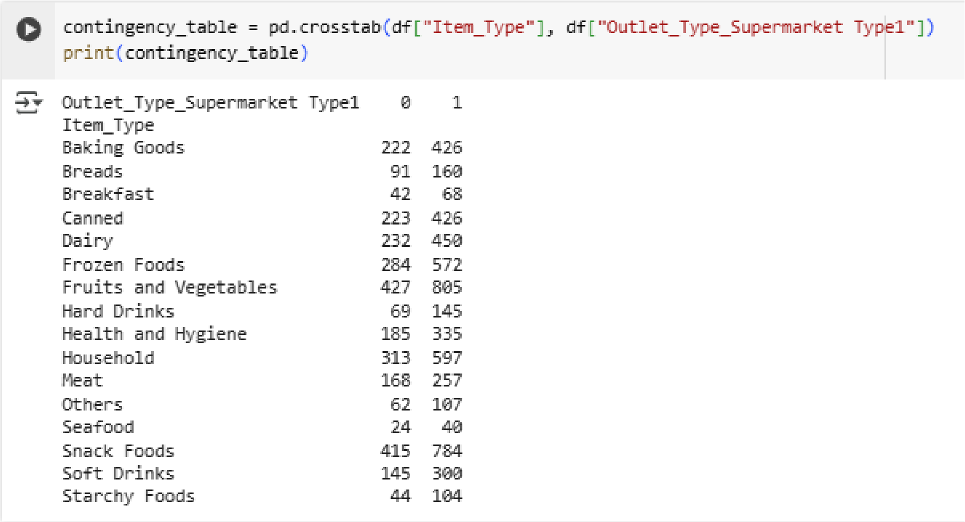
**Seaborn** :
It is built on top of Matplotlib, is a specialized statistical visualization library that enhances both aesthetics and usability. It provides built-in themes, color palettes, and high-level abstractions for statistical plots like box plots, violin plots, and scatter plots with regression lines.
Seaborn excels in creating complex visualizations such as pair plots, heatmaps, and cluster maps, making it particularly useful for data analysis and pattern recognition. While Matplotlib offers low-level control for general-purpose plotting, Seaborn simplifies statistical data visualization, making it an essential tool for analysts and researchers aiming to extract insights from data efficiently.

1. **Bar graph and contingency table using any two features:**
   a. Distribution of item type.
      A bar graph is a visual representation of categorical data where each category is represented by a bar. You can use the plt.bar function in matplotlib or sns.barplot in seaborn to create a bar graph.



The bar graph shows that **Fruits and Vegetables** and **Snack Foods** generate the highest total sales, while **Seafood** has the least.

b. Contingency table

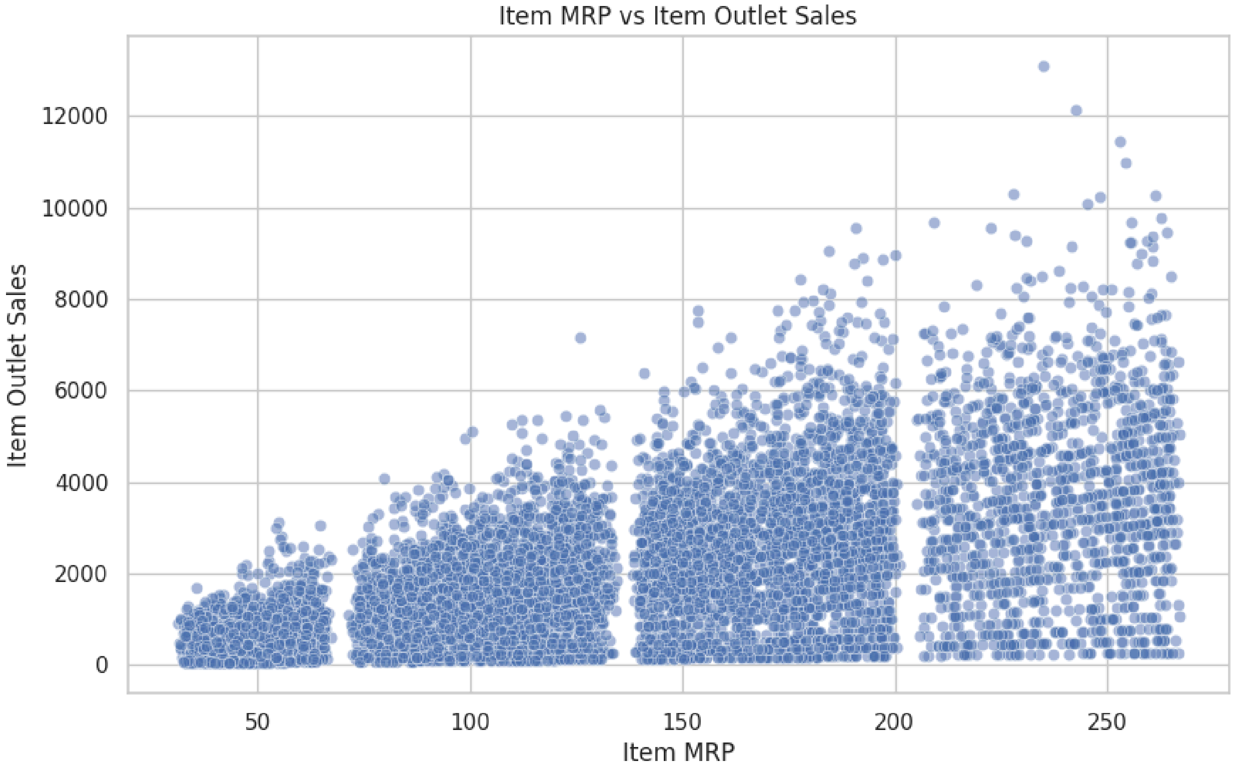


```
contingency_table = pd.crosstab(df["Item_Type"], df["Outlet_Type_Supermarket Type1"])
print(contingency_table)
```

```
Outlet_Type_Supermarket Type1    0    1
Item_Type
Baking Goods                   222  426
Breads                          91  160
Breakfast                       42   68
Canned                         223  426
Dairy                          232  450
Frozen Foods                   284  572
Fruits and Vegetables          427  805
Hard Drinks                     69  145
Health and Hygiene             185  335
Household                      313  597
Meat                           168  257
Others                          62  107
Seafood                         24   40
Snack Foods                    415  784
Soft Drinks                    145  300
Starchy Foods                   44  104
```

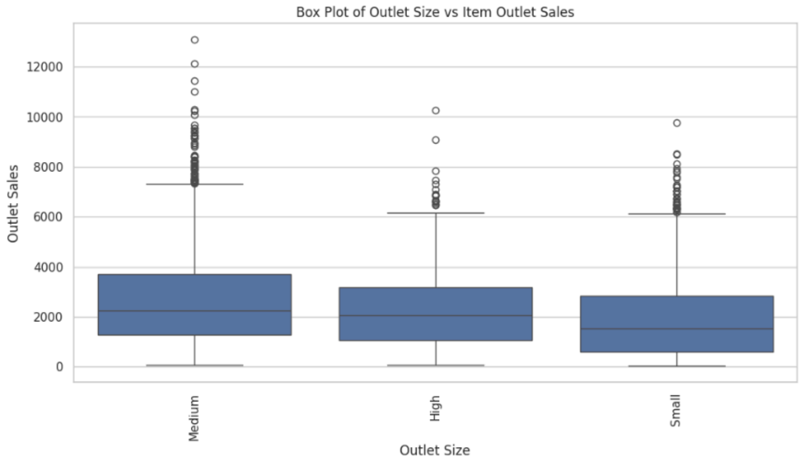
Supermarket Type 1 serves as a **one-stop shop for daily essentials**, whereas the other outlet type likely caters to a **more limited or specialized consumer base**, possibly smaller grocery stores

2. **Scatter plot**



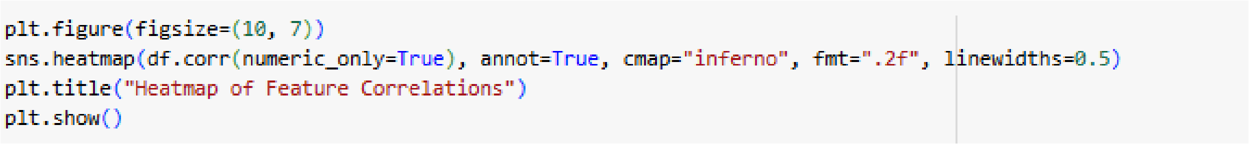
The scatter plot of Item MRP vs. Item Outlet Sales shows a positive trend, indicating that as the Item MRP increases, the Item Outlet Sales also tend to increase, although with some variability.Additionally, the spread of sales increases with MRP, meaning higher-priced items have a wider range of sales figures, possibly due to variations in demand across different products.

3. **Box plot**

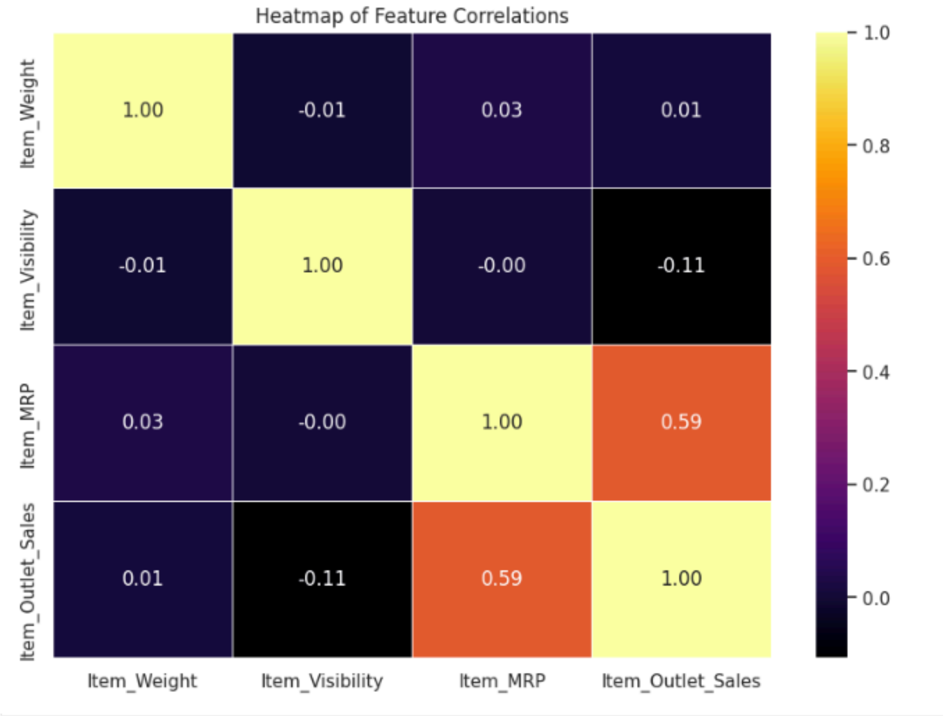


This plot shows that median sales are fairly similar across different outlet sizes, with Medium outlets having slightly higher median sales. The spread of sales is also comparable across all outlet sizes, with many outliers indicating high sales in certain cases. We can conclude that outlet size alone may not be a strong determinant of sales, and other factors like location, promotions, and product variety may play a more significant role

4. **Heat map**



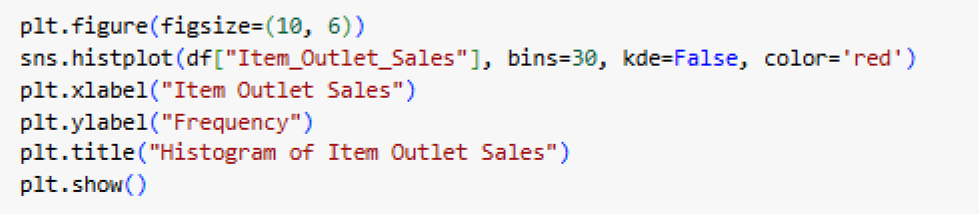
```
plt.figure(figsize=(10, 7))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap="inferno", fmt=".2f", linewidths=0.5)
plt.title("Heatmap of Feature Correlations")
plt.show()
```



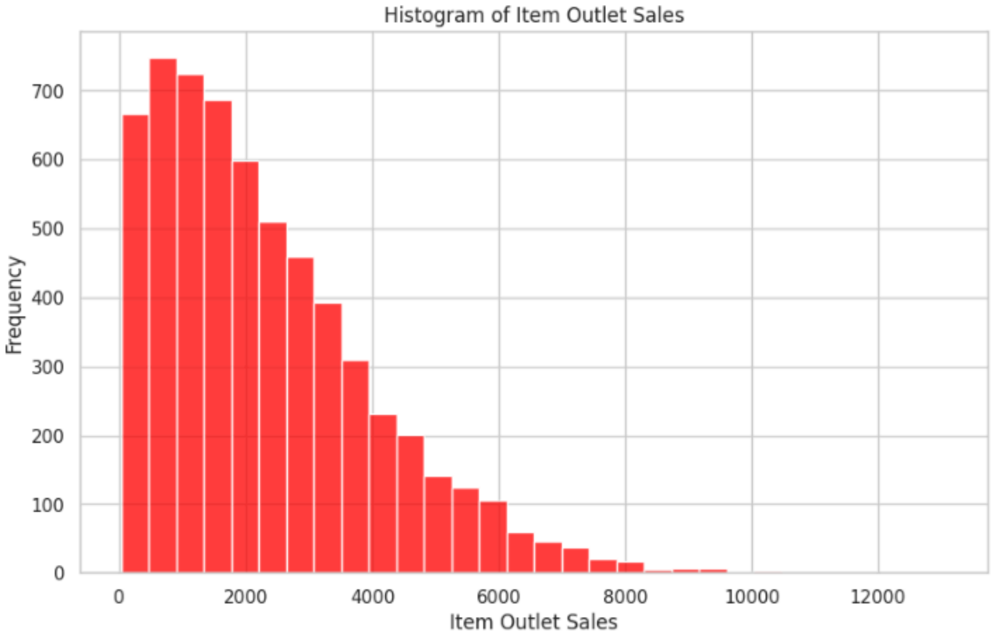
Strong correlation between **Item MRP and Item Outlet Sales**, meaning expensive items often have higher sales. This suggests that higher-priced items contribute significantly to total revenue, possibly due to factors such as better quality, brand reputation, or consumer preference for premium products

5. a. **Histogram**

A histogram is a graphical representation of the distribution of a continuous variable. It divides the range of values into bins and shows the frequency or count of observations within each bin. Histograms provide insights into the shape and spread of the data distribution.

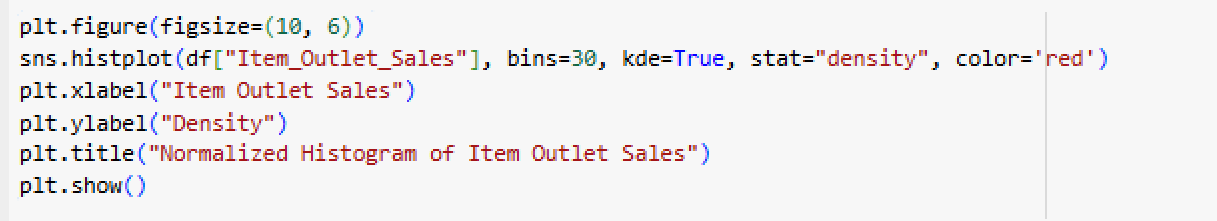


```
plt.figure(figsize=(10, 6))
sns.histplot(df["Item_Outlet_Sales"], bins=30, kde=False, color='red')
plt.xlabel("Item Outlet Sales")
plt.ylabel("Frequency")
plt.title("Histogram of Item Outlet Sales")
plt.show()
```

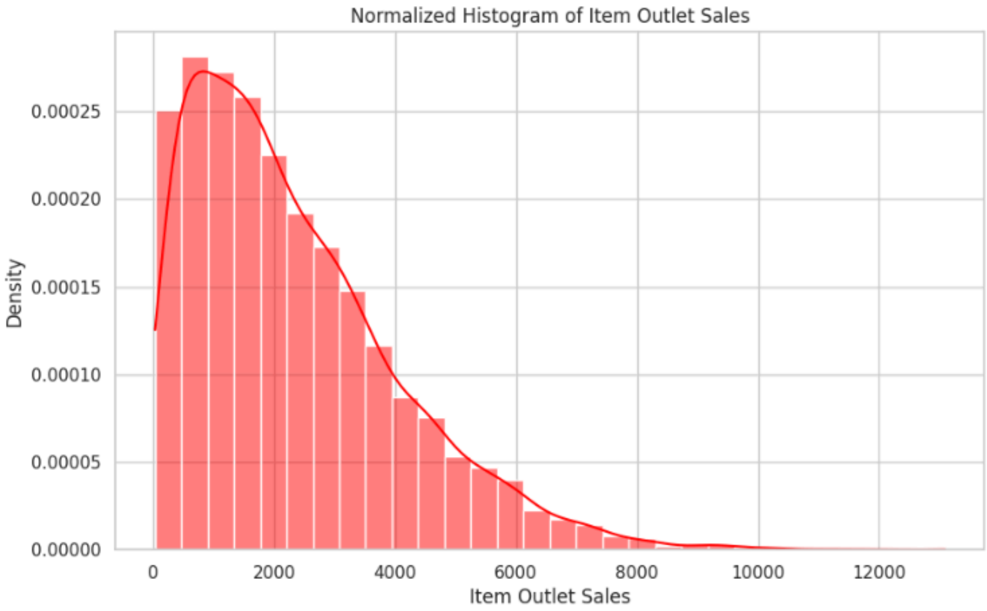


b. **Normalized Histogram**

A normalized histogram represents the relative frequencies or proportions of observations in each bin, providing a normalized view of the distribution. It is useful for comparing distributions of variables with different scales or sample sizes.

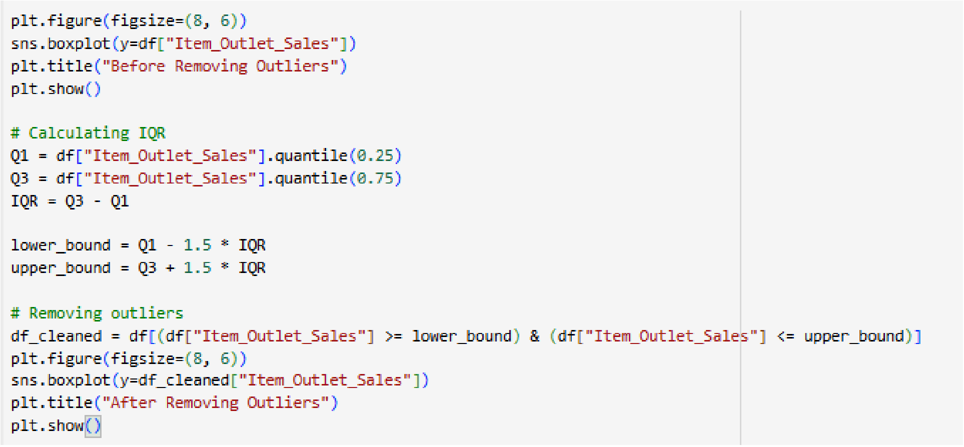


```
plt.figure(figsize=(10, 6))
sns.histplot(df["Item_Outlet_Sales"], bins=30, kde=True, stat="density", color='red')
plt.xlabel("Item Outlet Sales")
plt.ylabel("Density")
plt.title("Normalized Histogram of Item Outlet Sales")
plt.show()
```



Key observations:
1. The **histogram** shows that sales are heavily skewed to the right, meaning most sales values are low, with fewer items generating very high sales.
2. The **normalized histogram** (with KDE) confirms this trend, showing a high density at lower sales values.

### 6. **Handle outlier using box plot and Inter quartile range**

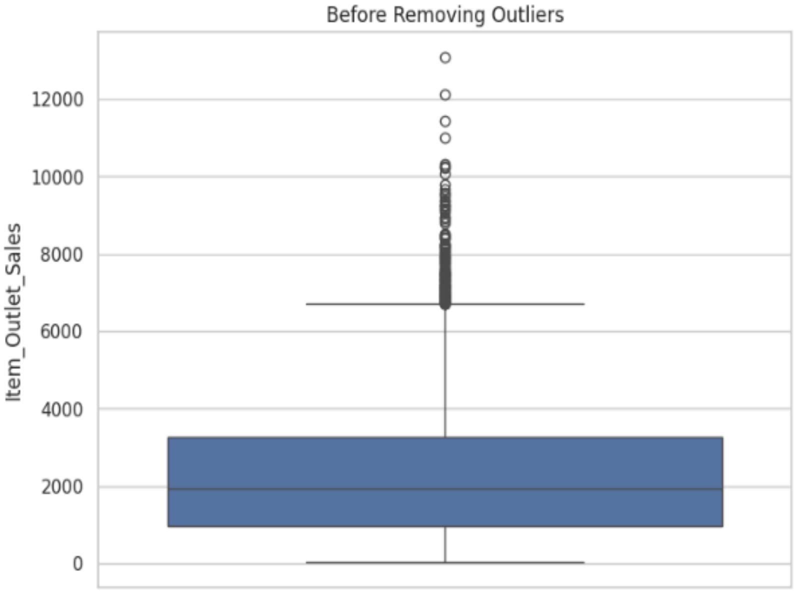


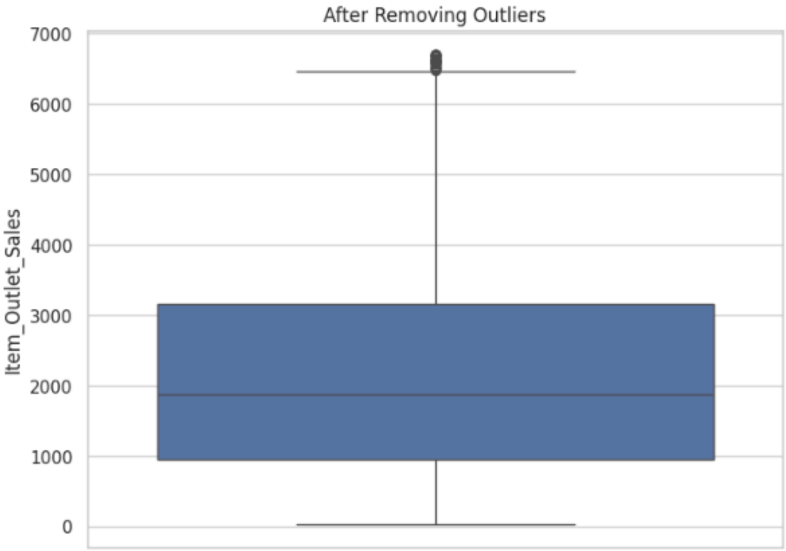
```
plt.figure(figsize=(8, 6))
sns.boxplot(y=df["Item_Outlet_Sales"])
plt.title("Before Removing Outliers")
plt.show()

# Calculating IQR
Q1 = df["Item_Outlet_Sales"].quantile(0.25)
Q3 = df["Item_Outlet_Sales"].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Removing outliers
df_cleaned = df[(df["Item_Outlet_Sales"] >= lower_bound) & (df["Item_Outlet_Sales"] <= upper_bound)]
plt.figure(figsize=(8, 6))
sns.boxplot(y=df_cleaned["Item_Outlet_Sales"])
plt.title("After Removing Outliers")
plt.show()
```





In the first box plot, a large number of extreme outliers are observed, particularly above **6000** in Item Outlet Sales. These extreme values suggest that while most sales remain within a typical range, certain items experience significantly higher sales, possibly due to promotions, high demand, or premium pricing. **After applying the Interquartile Range (IQR) method**, these extreme outliers are removed, resulting in a more balanced dataset.

**Conclusion**:
In the end, gained valuable insights into sales trends, pricing impact, and outlet characteristics. The bar graph highlighted that Fruits and Vegetables, along with Snack Foods, contribute the most to total sales, while Seafood sees the least. The contingency table revealed that Supermarket Type 1 serves as a primary shopping destination, while other outlets likely cater to

niche markets.Overall, this analysis provided a clearer understanding of sales distribution, pricing effects, and outlet performance