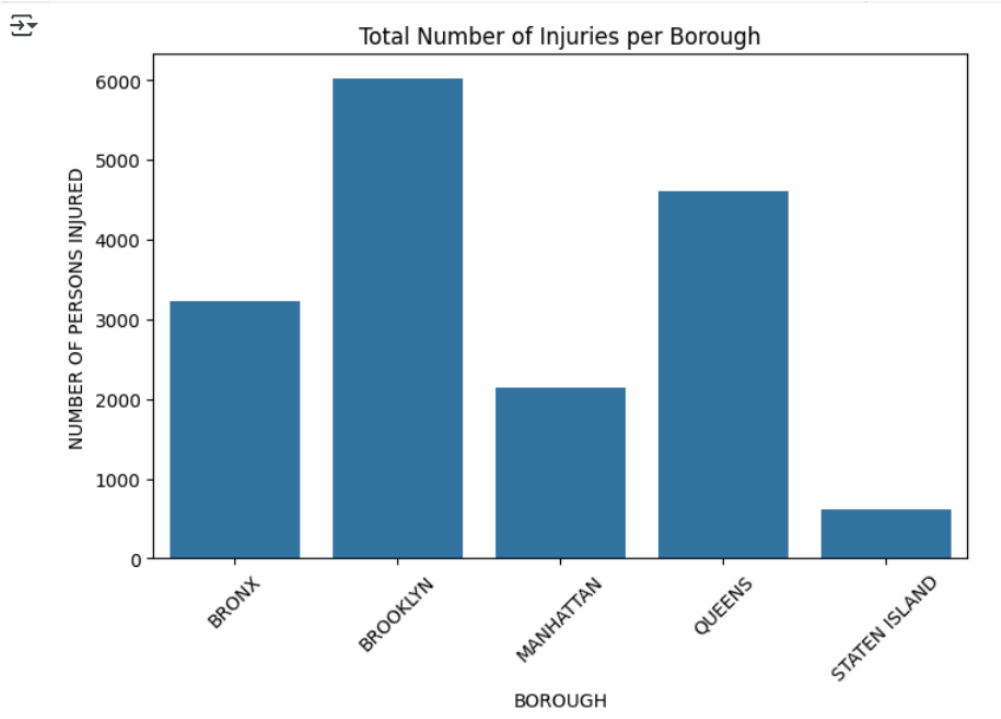
Experiment No. 2

Problem Statement: Data Visualization/ Exploratory data Analysis using Matplotlib and Seaborn.

Perform following data visualization and exploration on your selected dataset.

1. **Create bar graph, contingency table using any 2 features.**

```
import matplotlib.pyplot as plt
import seaborn as sns

df_grouped = df.groupby("BOROUGH")["NUMBER OF PERSONS
INJURED"].sum().reset_index()
plt.figure(figsize=(8,5))
sns.barplot(x="BOROUGH", y="NUMBER OF PERSONS INJURED", data=df_grouped)
plt.title("Total Number of Injuries per Borough")
plt.xticks(rotation=45)
plt.show()

# Contingency Table: Borough vs. Injuries
contingency_table = pd.crosstab(df['BOROUGH'], df['NUMBER OF PERSONS
INJURED'])
print("Contingency Table:\n", contingency_table)
```

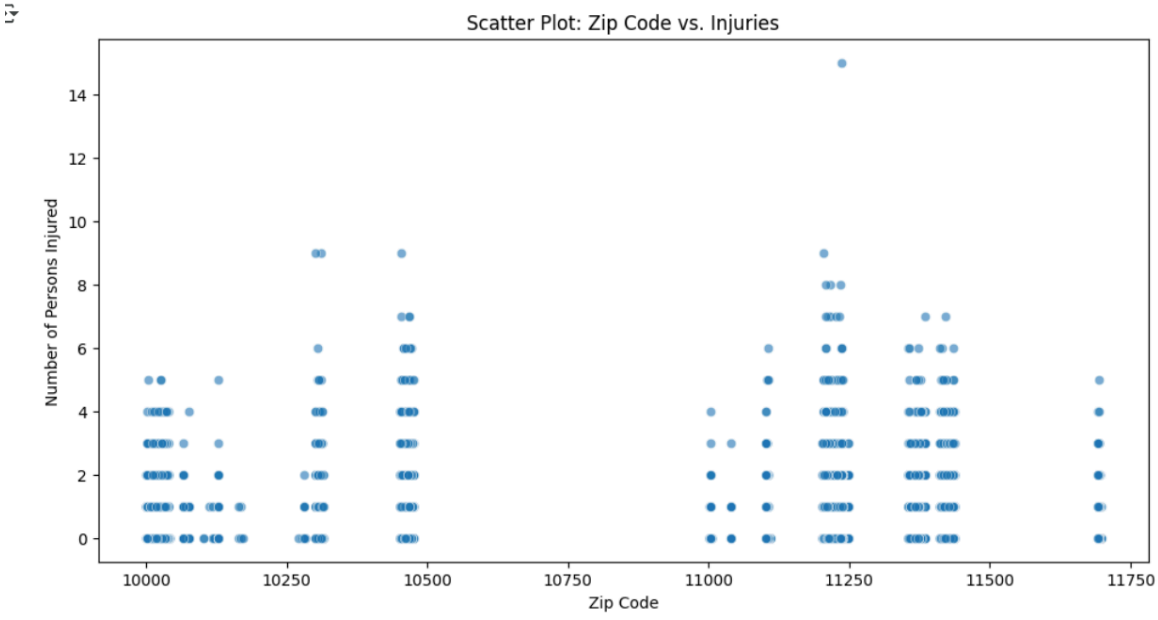


Contingency Table:

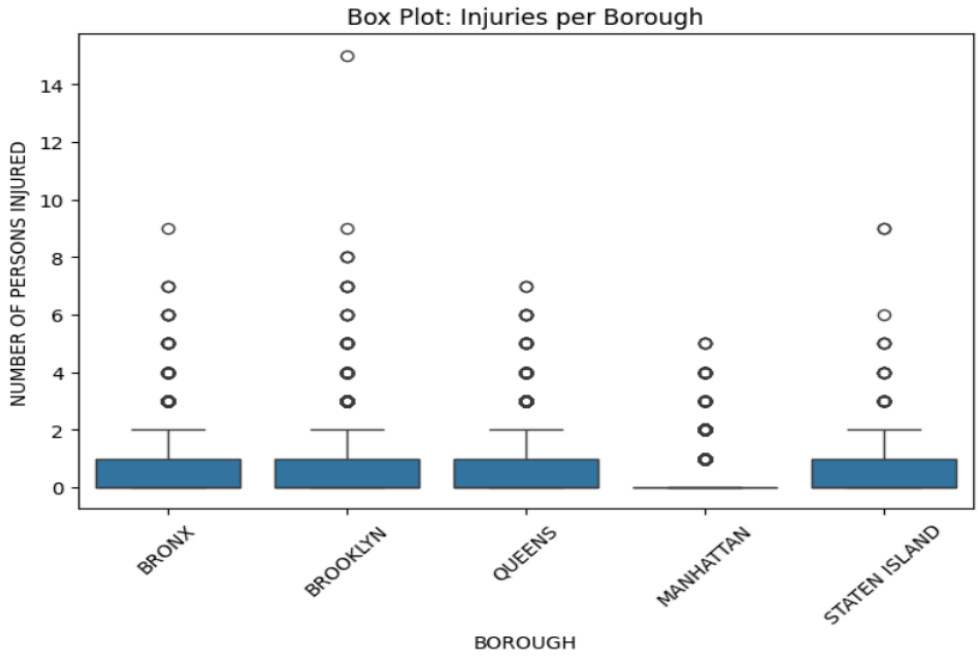
NUMBER OF PERSONS INJURED	0	1	2	3	4	5	6	7	8	9	15
BOROUGH											
BRONX	6956	1971	309	121	36	14	6	3	0	1	0
BROOKLYN	12350	3574	670	219	55	24	5	5	3	1	1
MANHATTAN	5553	1553	177	40	26	4	0	0	0	0	0
QUEENS	10483	2799	517	142	50	17	7	2	0	0	0
STATEN ISLAND	1007	334	69	22	7	4	1	0	0	2	0

2. Plot Scatter plot, box plot, Heatmap using seaborn.

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12,6))
sns.scatterplot(x=df["ZIP CODE"], y=df["NUMBER OF PERSONS INJURED"], alpha=0.6)
plt.xlabel("Zip Code")
plt.ylabel("Number of Persons Injured")
plt.title("Scatter Plot: Zip Code vs. Injuries")
plt.show()
```

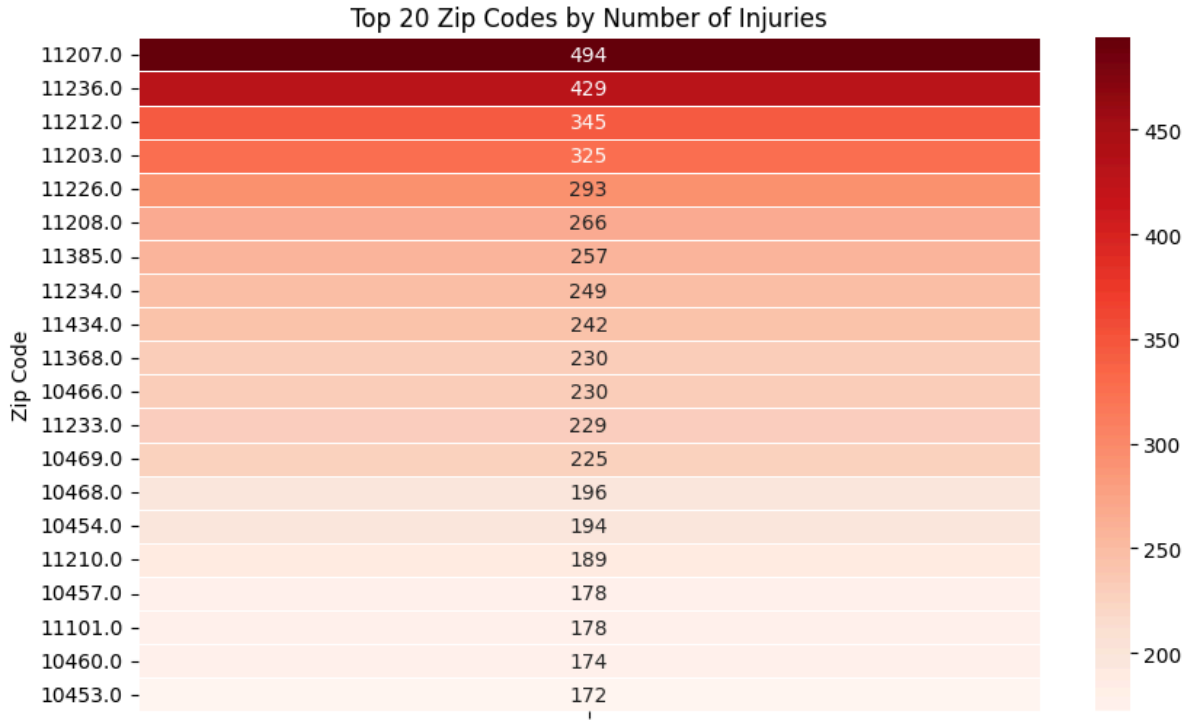


```
plt.figure(figsize=(8,5))
sns.boxplot(x="BOROUGH", y="NUMBER OF PERSONS INJURED", data=df)
plt.xticks(rotation=45)
plt.title("Box Plot: Injuries per Borough")
plt.show()
```




```
import seaborn as sns
import matplotlib.pyplot as plt

# Group by ZIP Code and sum injuries
zip_injury = df.groupby("ZIP CODE")["NUMBER OF PERSONS
INJURED"].sum().reset_index()

# Sort by highest injuries
zip_injury = zip_injury.sort_values(by="NUMBER OF PERSONS INJURED",
ascending=False).head(20)

# Set ZIP Code as index for heatmap
zip_injury_pivot = zip_injury.set_index("ZIP CODE")

# Creating an improved heatmap
```

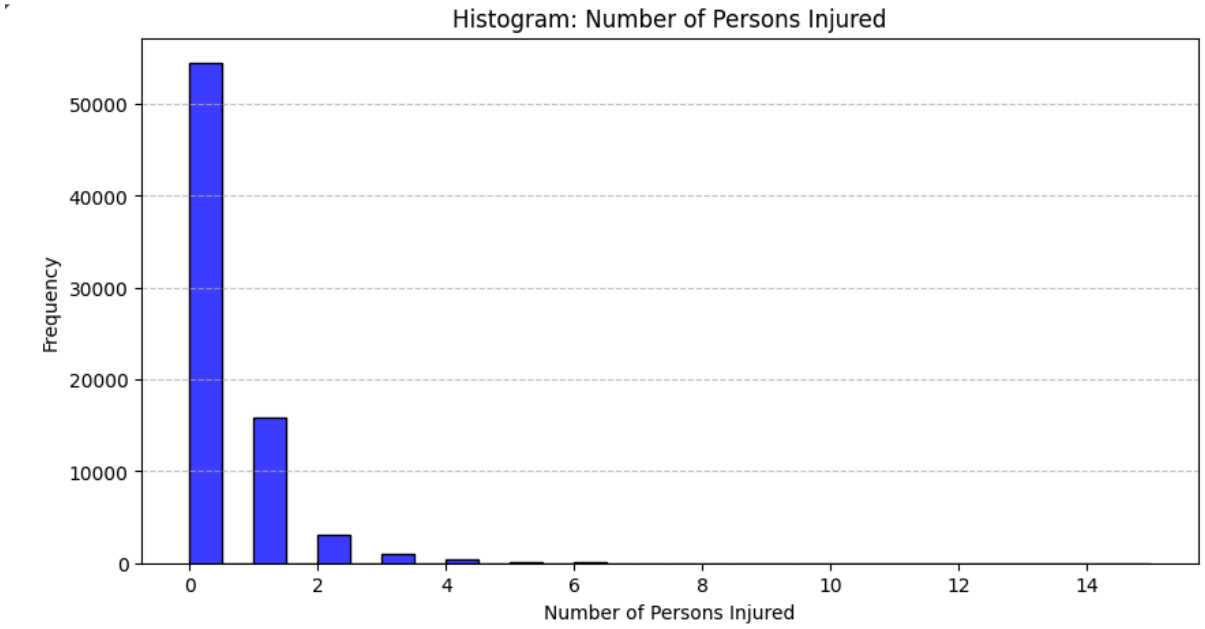
```
plt.figure(figsize=(10, 6))
sns.heatmap(zip_injury_pivot, cmap="Reds", linewidths=0.5, annot=True,
fmt=".0f")

plt.title("Top 20 Zip Codes by Number of Injuries")
plt.xlabel("Number of Persons Injured")
plt.ylabel("Zip Code")
plt.xticks(rotation=45)  # Rotate labels for better visibility
plt.show()
```

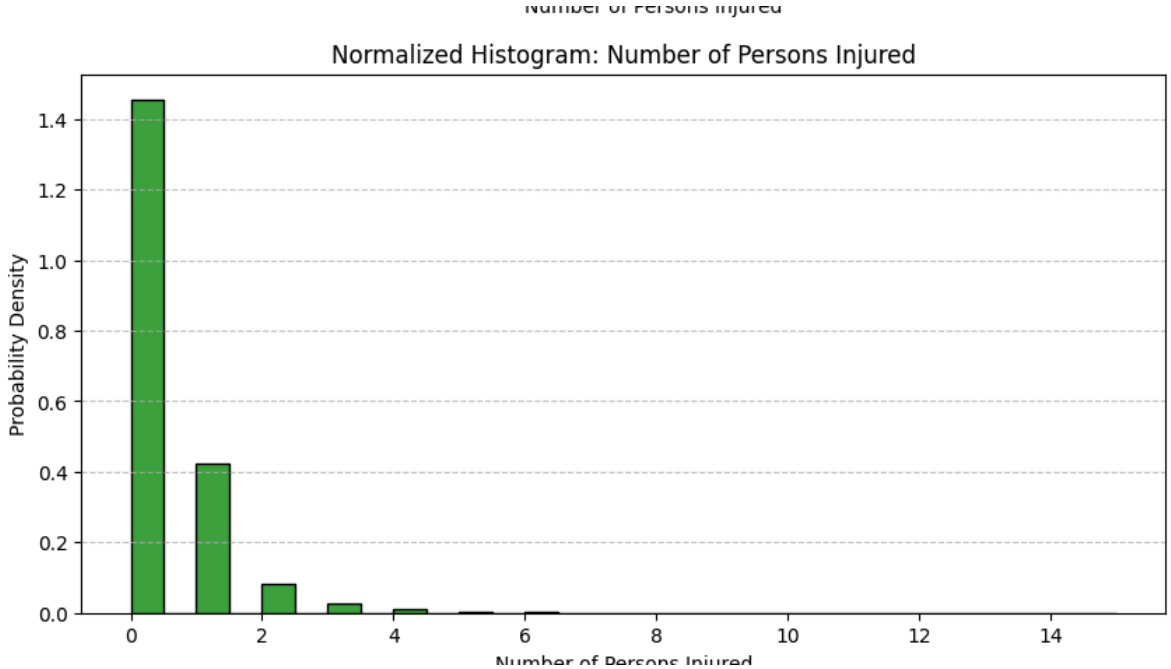


3. **Create histogram and normalized Histogram.**

```
import matplotlib.pyplot as plt
import seaborn as sns

# Extract the column of interest
injury_data = df["NUMBER OF PERSONS INJURED"].dropna()

# Creating the Regular Histogram
plt.figure(figsize=(10, 5))
```

```
sns.histplot(injury_data, bins=30, kde=False, color="blue")
plt.title("Histogram: Number of Persons Injured")
plt.xlabel("Number of Persons Injured")
plt.ylabel("Frequency")
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.show()

# Creating the Normalized Histogram
plt.figure(figsize=(10, 5))
sns.histplot(injury_data, bins=30, kde=False, color="green", stat="density")  # Normalized
plt.title("Normalized Histogram: Number of Persons Injured")
plt.xlabel("Number of Persons Injured")
plt.ylabel("Probability Density")
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.show()
```





4. **Describe what this graph and table indicates.**

Bar Graph & Contingency Table

- **Bar Graph:**
    - A bar graph visually represents the frequency distribution of categorical variables.
    - Example: A bar graph comparing different **boroughs vs. number of accidents** shows which borough has the highest incidents.
    - Interpretation: If **Manhattan** has the highest bar, it means that most accidents occur there compared to other boroughs.
- **Contingency Table:**
    - A contingency table (cross-tabulation) shows relationships between two categorical variables.
    - Example: A table comparing **Vehicle Type vs. Cause of Accident** might reveal that **SUVs are more likely involved in collisions due to driver distraction**.
    - Interpretation: Helps identify patterns in accident causes based on vehicle type.

---

2. Scatter Plot, Box Plot, Heatmap (Seaborn)

- **Scatter Plot:**
  - Displays relationships between two numerical variables.
  - Example: **Zip Code vs. Number of Persons Injured** might show clusters indicating higher injuries in specific areas.
  - Interpretation: If points form a pattern, it suggests correlation (e.g., high injury numbers in densely populated zip codes).
- **Box Plot:**
  - Shows the distribution of numerical data and outliers.
  - Example: **Number of Injuries per Borough**
  - Interpretation: If **Queens** has a long upper whisker, it suggests that some accidents there involve **significantly more injuries** than others.
- **Heatmap:**
  - Visualizes data intensity using color gradients.
  - Example: **Zip Code vs. Number of Accidents** using a heatmap will show dark areas in locations with frequent accidents.
  - Interpretation: Helps identify accident-prone areas that may need improved safety measures.

---

3. Histogram & Normalized Histogram

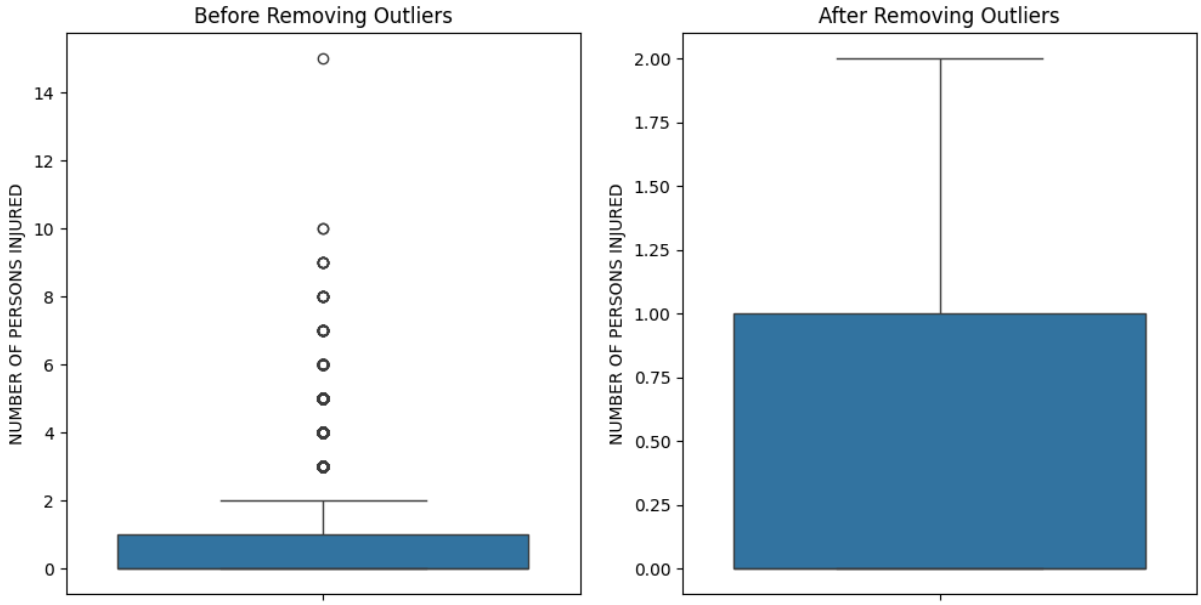
- **Histogram:**
  - Displays the frequency of values within different ranges (bins).
  - Example: **Number of Persons Injured** histogram shows that most accidents involve 0-2 injuries.
  - Interpretation: If most values are concentrated in **low injury numbers**, it suggests that severe accidents are rare.
- **Normalized Histogram:**
  - Similar to a histogram but represents probability density instead of frequency.
  - Interpretation: It helps compare distributions across different datasets without being affected by sample size.

5. Handle outlier using box plot and Inter quartile range.

```
import pandas as pd
import seaborn as sns
```

```
import matplotlib.pyplot as plt

# Load dataset (replace with your actual dataset)
df = pd.read_csv("nyc_accidents.csv")

# Choose a numerical column, e.g., 'NUMBER OF PERSONS INJURED'
col = 'NUMBER OF PERSONS INJURED'

# Calculate Q1, Q3, and IQR
Q1 = df[col].quantile(0.25)
Q3 = df[col].quantile(0.75)
IQR = Q3 - Q1

# Define lower and upper bounds
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Identify outliers
outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)]
print("Number of outliers:", len(outliers))

# Remove outliers
df_cleaned = df[(df[col] >= lower_bound) & (df[col] <= upper_bound)]

# Plot boxplot before and after removing outliers
fig, axes = plt.subplots(1, 2, figsize=(12, 6))
sns.boxplot(y=df[col], ax=axes[0])
axes[0].set_title("Before Removing Outliers")

sns.boxplot(y=df_cleaned[col], ax=axes[1])
axes[1].set_title("After Removing Outliers")

plt.show()
```

Number of outliers: 1543



Conclusion:
In this experiment, we performed exploratory data analysis (EDA) and data visualization using Matplotlib and Seaborn to uncover patterns and insights from the dataset. Through bar graphs and contingency tables, we identified boroughs with the highest accident-related injuries. Scatter plots and heatmaps highlighted spatial trends, while box plots revealed data distribution and outliers. Histograms and normalized histograms provided insights into the frequency and probability distribution of injuries. Outliers were detected and handled using the interquartile range (IQR) method, ensuring a cleaner dataset for analysis. These visualizations and statistical methods helped in understanding accident trends, identifying high-risk areas, and emphasizing the need for data-driven safety measures.