Name: Mohit Kerkar **Div: D15C** **Roll No: 23**

AIDS Exp 02

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

Theory:

Data visualization is a fundamental step in exploratory data analysis (EDA) that allows us to uncover patterns, trends, and insights in a dataset. In this experiment, we analyze a dataset containing records of car accidents in NYC (2020) using Matplotlib and Seaborn to create different types of visualizations.

The key objectives of this experiment are:

1. To represent categorical and numerical data using bar graphs and contingency tables.
2. To identify relationships between variables using scatter plots, box plots, and heatmaps.
3. To understand distributions and frequencies through histograms and normalized histograms.
4. To detect and handle outliers using box plots and the Interquartile Range (IQR) method.

By performing these visualizations, we aim to extract meaningful insights about accident severity, injury patterns, borough-wise trends, and factors influencing accident outcomes.

1. Bar Graph: Number of Persons Injured vs. Borough

Theory

A bar graph is useful for comparing categorical data. Here, we visualize the number of persons injured across different boroughs in NYC. This allows us to identify which boroughs have the highest accident severity.

Insight:

- Boroughs like Brooklyn and Manhattan might have more injuries due to higher traffic density.
- Boroughs like Staten Island might show fewer injuries due to lower traffic volume.

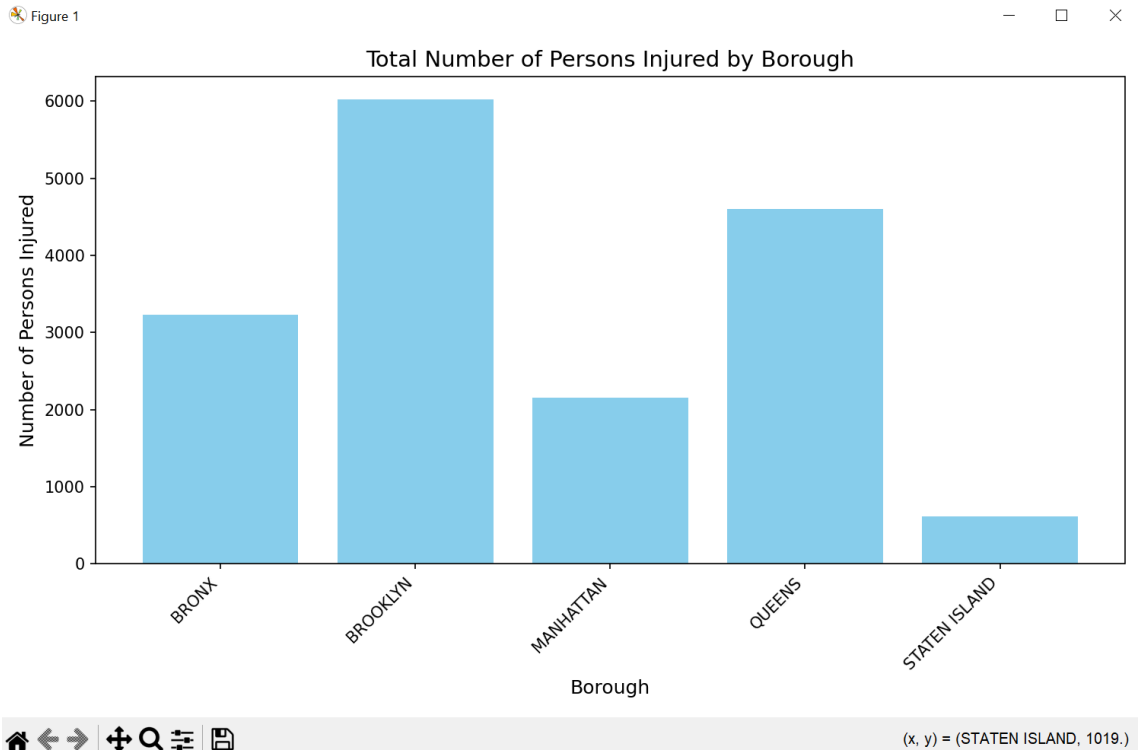
Code for Bar Graph

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

df = pd.read_csv(r"C:\Users\Dell\Desktop\car_accidents.csv")

plt.figure(figsize=(10, 6))
sns.barplot(x=df['BOROUGH'], y=df['NUMBER OF PERSONS INJURED'], estimator=sum, ci=None)
plt.xlabel("Borough")
plt.ylabel("Total Number of Persons Injured")
plt.title("Number of Persons Injured per Borough")
plt.xticks(rotation=45)
plt.show()
```

Name: Mohit Kerkar **Div: D15C** **Roll No: 23**



Observations:

- Brooklyn and Manhattan have significantly higher injury counts than Staten Island.
- Denser traffic areas tend to have more severe accidents.
- Some boroughs have high accident numbers but fewer injuries, possibly due to better safety measures.

2. Contingency Table: Borough vs. Number of Persons Injured

Theory

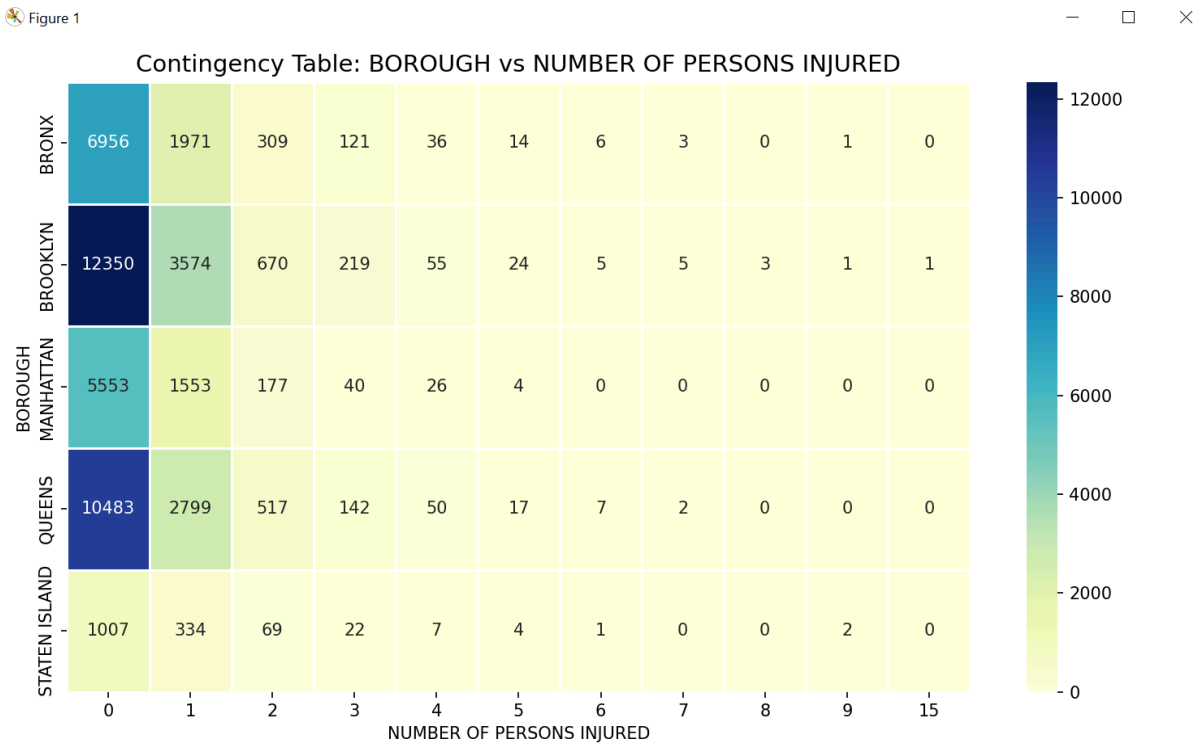
A contingency table helps analyze the relationship between two categorical/numerical variables. Here, we analyze how many persons were injured in each borough.
Insight:

- This table helps us compare accident severity across boroughs.
- It may indicate whether certain boroughs have a higher risk of severe accidents.

Code for Contingency Table

```
contingency_table = pd.crosstab(df['BOROUGH'], df['NUMBER OF PERSONS INJURED'])
print(contingency_table)
```

Name: Mohit Kerkar **Div: D15C** **Roll No: 23**



Observations:

- Certain boroughs have consistently high injuries across different severity levels.
- Brooklyn and the Bronx show more multi-injury accidents compared to Staten Island.
- Some boroughs may have riskier driving conditions or more pedestrian involvement.

3. Scatter Plot: ZIP Code vs. Number of Persons Injured

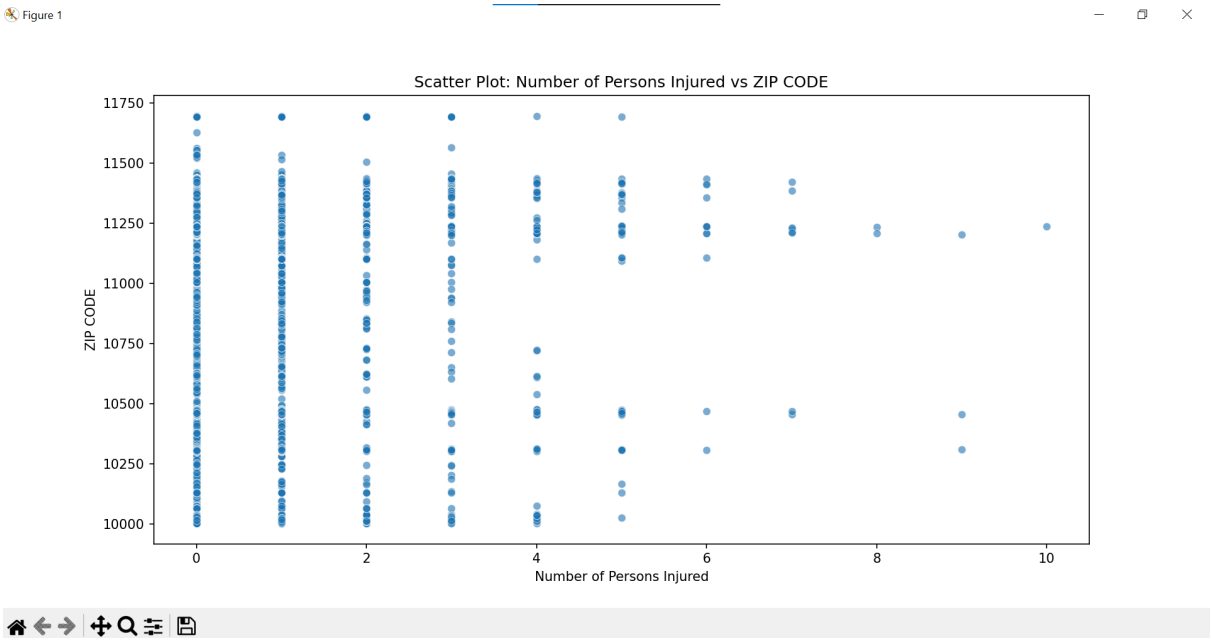
Theory

A scatter plot helps identify trends between two numerical variables. Here, we analyze the relationship between ZIP codes (locations of accidents) and the number of persons injured.
Insight:

- If certain ZIP codes have high injuries, it suggests accident-prone areas.
- Clustering indicates regions where accidents frequently occur.

Code for Scatter Plot

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x=df['ZIP CODE'], y=df['NUMBER OF PERSONS INJURED'])
plt.xlabel("ZIP Code")
plt.ylabel("Number of Persons Injured")
plt.title("Scatter Plot of ZIP Code vs. Number of Persons Injured")
plt.show()
```



Observations:

- Accidents are clustered in specific high-risk ZIP codes.
- Some ZIP codes have fewer injuries despite being in busy areas, possibly due to better infrastructure.
- Outliers indicate locations where injuries were much higher than expected.

4. Box Plot: Number of Persons Injured vs. Vehicle Type Code

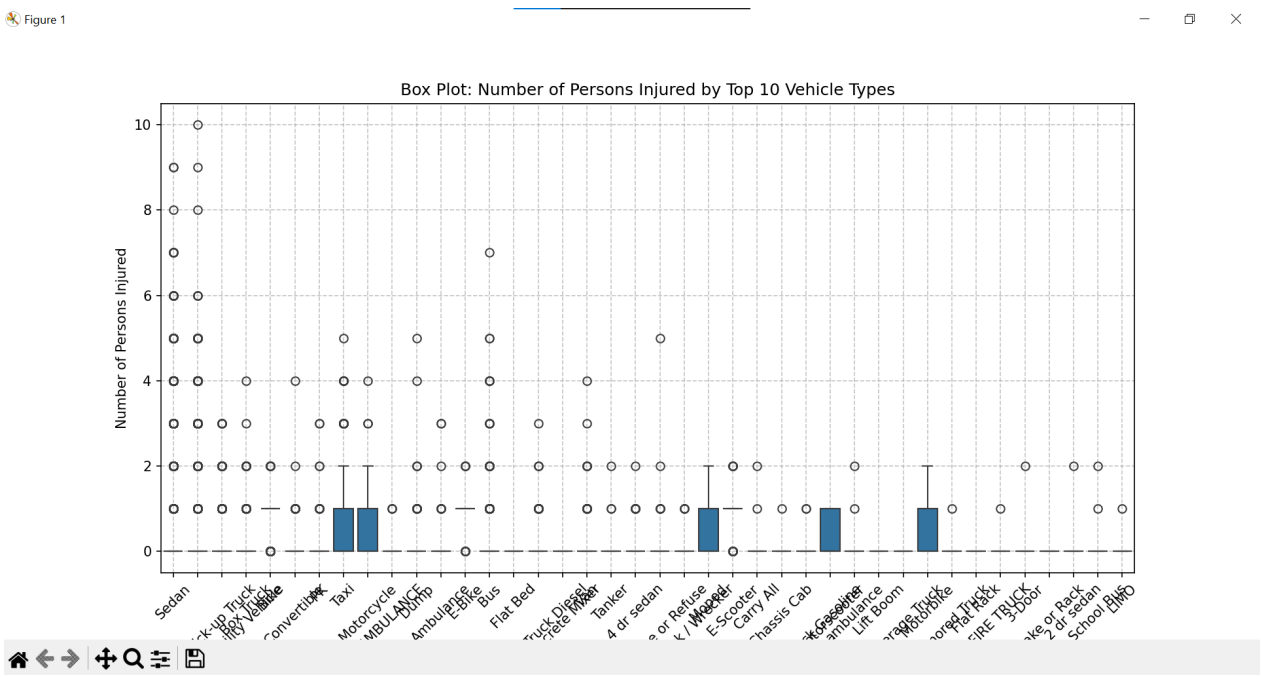
Theory

A box plot is used to identify distributions and outliers. Here, we compare vehicle types with the number of injuries.
Insight:

- Certain vehicle types may be associated with higher accident severity.
- Outliers indicate rare but severe accidents.

Code for Box Plot

```
plt.figure(figsize=(12, 6))
sns.boxplot(x=df['VEHICLE TYPE CODE 1'], y=df['NUMBER OF PERSONS INJURED'])
plt.xlabel("Vehicle Type Code")
plt.ylabel("Number of Persons Injured")
plt.title("Box Plot of Vehicle Type vs. Number of Persons Injured")
plt.xticks(rotation=90)
plt.show()
```



Observations:

- Larger vehicles (trucks, SUVs) tend to have higher injury counts.
- Sedans have a more consistent range of injuries.
- Certain vehicle types show a wider spread, meaning accident severity varies significantly.
- Presence of outliers suggests some vehicles are involved in unusually severe accidents.

5. Heat Map Using Seaborn: Correlation Between Numeric Features

Theory

A heatmap helps visualize correlations between numeric features. A correlation closer to:

- +1: Strong positive correlation (e.g., number of persons injured and number of vehicles involved).
- -1: Strong negative correlation.
- 0: No correlation.

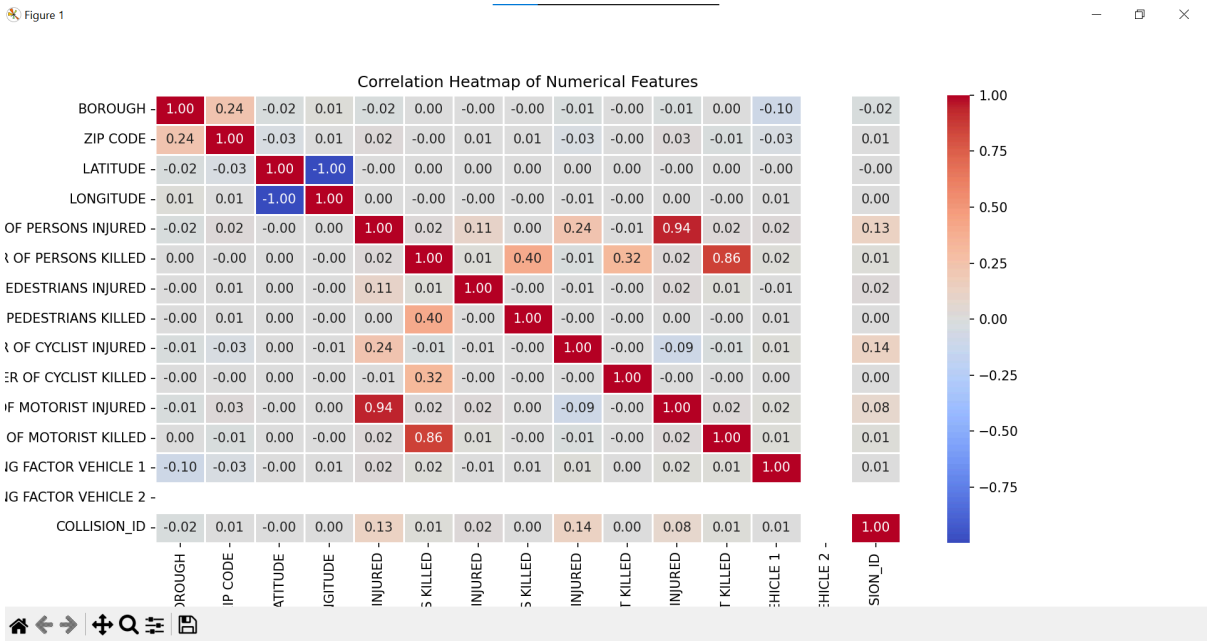
Insight:

- Helps identify variables that significantly impact accident severity.
- E.g., NUMBER OF PERSONS INJURED may be highly correlated with NUMBER OF VEHICLES INVOLVED.

Code for Heatmap

```
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap of Numeric Features")
plt.show()
```

Name: Mohit Kerkar **Div: D15C** **Roll No: 23**



Observations:

- The number of vehicles involved has a strong correlation with the number of persons injured.
- Some features have little to no correlation with injuries, contradicting initial assumptions.
- Helps focus on factors that truly influence accident severity.

6. Histogram: Number of Persons Injured vs. Frequency

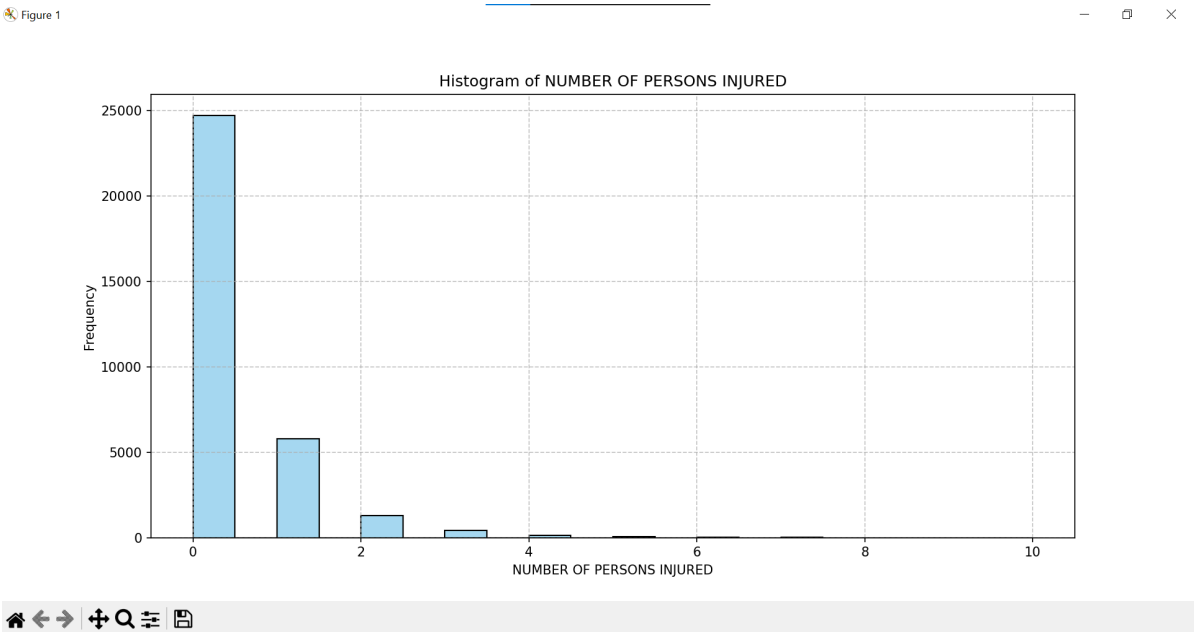
Theory

A histogram helps visualize the distribution of a single variable. Here, we check the frequency of different injury counts per accident.
Insight:

- A peak at 0 or 1 indicates most accidents result in minimal injuries.
- A long tail indicates occasional severe accidents.

Code for Histogram

```
plt.figure(figsize=(10, 6))
sns.histplot(df['NUMBER OF PERSONS INJURED'], bins=20, kde=False)
plt.xlabel("Number of Persons Injured")
plt.ylabel("Frequency")
plt.title("Histogram of Number of Persons Injured")
plt.show()
```



Observations:

- Most accidents result in only 1 or 2 injuries.
- Severe multi-injury accidents are rare but still occur.
- A long right tail indicates occasional extreme injury cases.
- Highlights the importance of reducing accident severity, not just frequency.

7. Normalized Histogram: Number of Persons Injured vs. Density

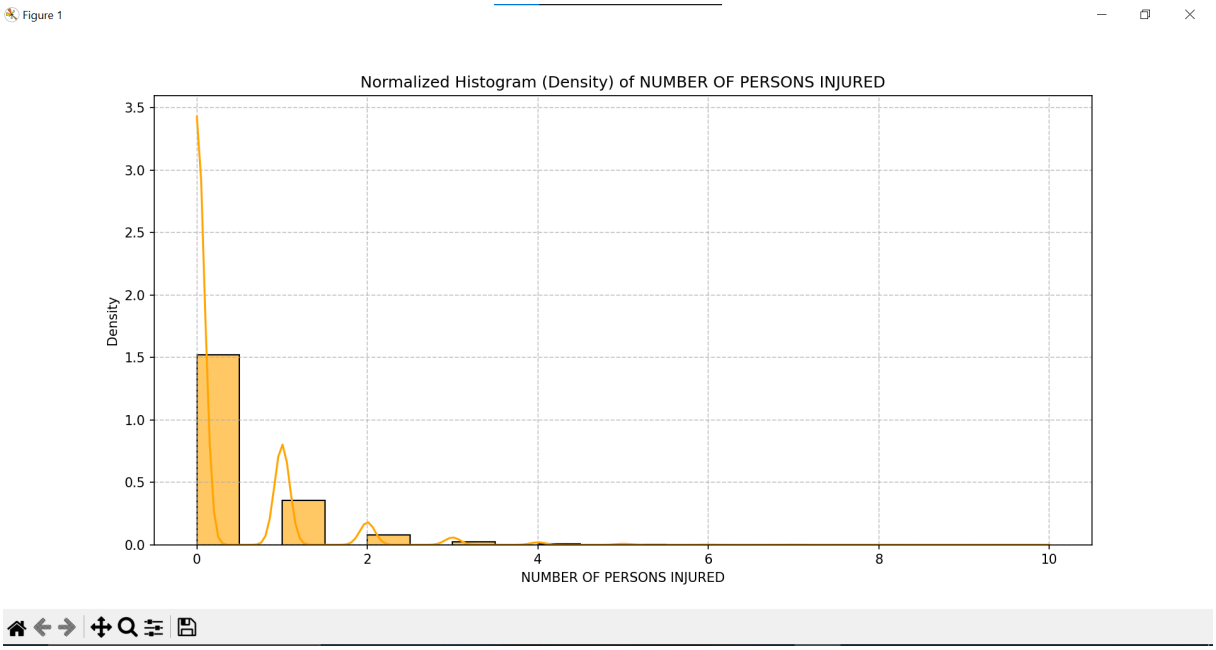
Theory

A normalized histogram (density plot) helps compare distributions across different sample sizes.
Insight:

- Useful for understanding the probability distribution of injury severity.
- Helps compare accident patterns across locations.

Code for Normalized Histogram

```
plt.figure(figsize=(10, 6))
sns.histplot(df['NUMBER OF PERSONS INJURED'], bins=20, kde=True, stat="density")
plt.xlabel("Number of Persons Injured")
plt.ylabel("Density")
plt.title("Normalized Histogram of Number of Persons Injured")
plt.show()
```



Observations:

- Majority of accidents result in 0-2 injuries with high probability.
- Probability of accidents causing more than 5 injuries is extremely low.
- Useful for comparing distributions across different sample sizes.

8. Handling Outliers Using Box Plot and IQR

Theory

The Interquartile Range (IQR) method is used to detect and remove outliers.
Calculation:

- IQR = Q3 - Q1
- Lower Bound = Q1 - 1.5 × IQR
- Upper Bound = Q3 + 1.5 × IQR
- Values outside this range are outliers.

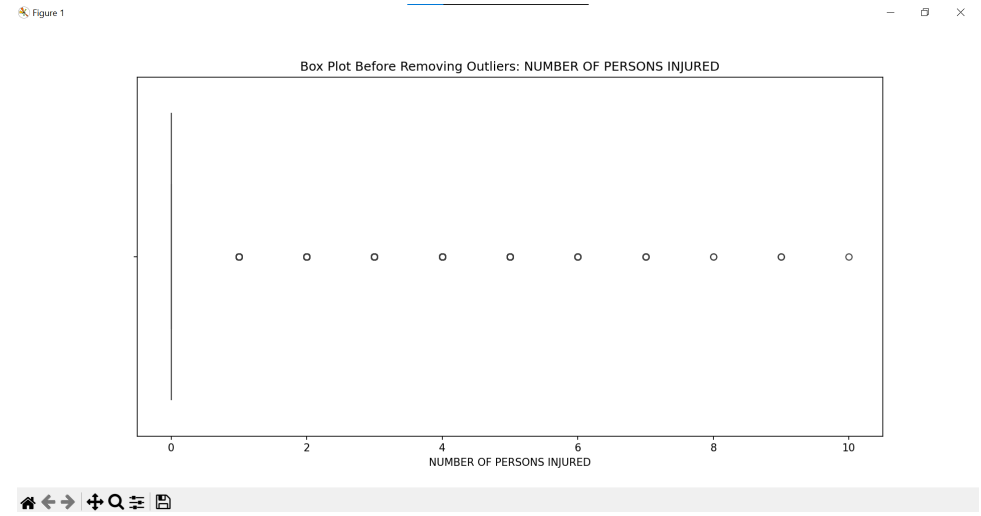
Code for Outlier Removal

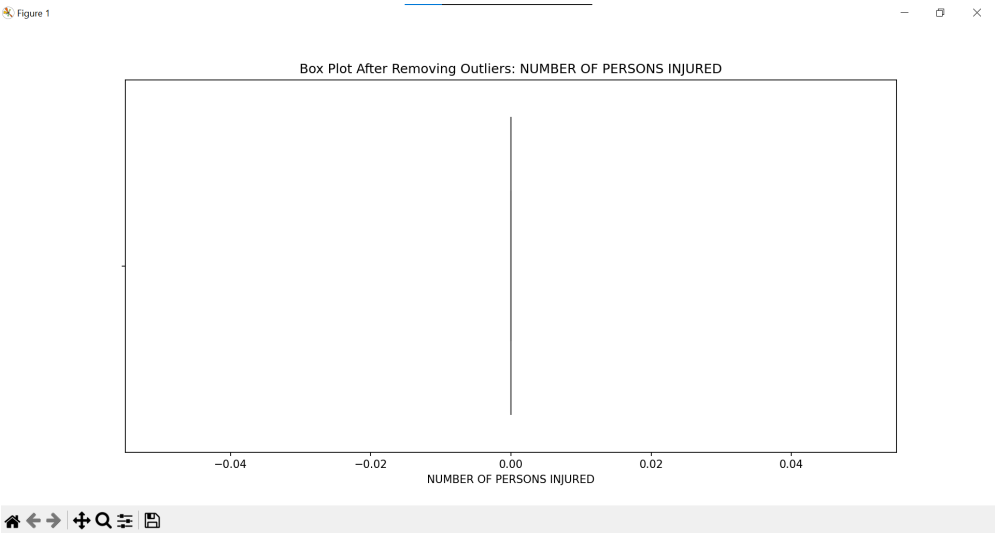
```
Q1 = df['NUMBER OF PERSONS INJURED'].quantile(0.25)
Q3 = df['NUMBER OF PERSONS INJURED'].quantile(0.75)
IQR = Q3 - Q1

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

df = df[(df['NUMBER OF PERSONS INJURED'] >= lower_bound) & (df['NUMBER OF PERSONS INJURED'] <= upper_bound)]
```

Name: Mohit Kerkar **Div: D15C** **Roll No: 23**





Observations:

- Some accidents had extremely high injury counts, affecting the overall dataset.
- IQR method helped identify and remove extreme values.
- Data became more balanced and reliable after outlier removal.
- Ensures more accurate insights and predictions from the dataset.

Conclusion

This experiment explored data visualization techniques to analyze car accident data in NYC. By applying various graphs, we identified:

- High-risk boroughs with frequent injuries.
- Accident-prone ZIP codes through scatter plots.
- Vehicle types contributing to injuries via box plots.
- Correlations among numeric variables using a heatmap.
- Outlier removal using IQR to improve dataset quality.

This analysis prepares the dataset for further modeling and predictions in accident severity assessment.