Experiment No:2

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

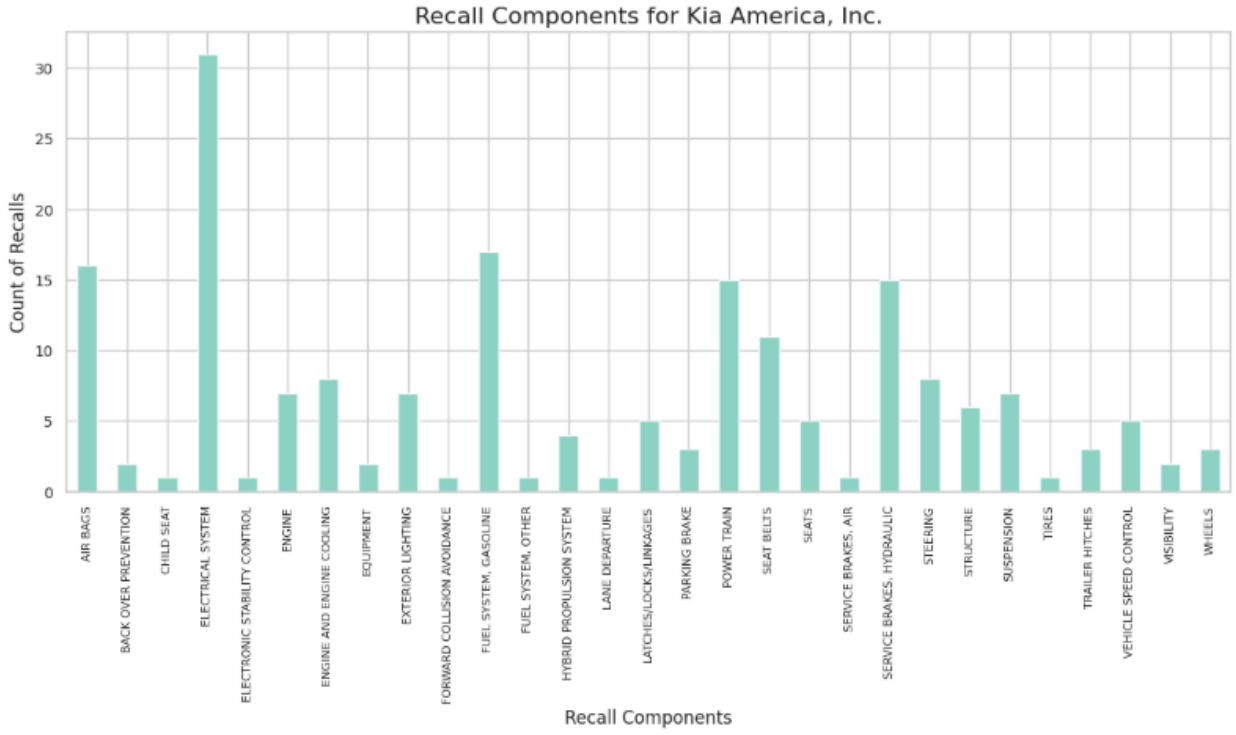
Theory:

1. Bar graph & contingency table using any two features :

1. Bar Graph:

This graph visualizes the count of recalls for different components associated with Kia America, Inc. Each bar represents a specific component and the number of recalls it has. It helps identify which components have the most recalls, aiding in targeting problem areas.

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

kia_df = df[df['Manufacturer'] == 'Kia America, Inc.']
crosstb_kia = pd.crosstab(kia_df['Component'], kia_df['Manufacturer'])
plt.figure(figsize=(12, 6))
crosstb_kia.plot.bar(figsize=(12, 6), colormap='Set3', rot=90)
plt.legend().set_visible(False)
plt.title('Recall Components for Kia America, Inc.', fontsize=16)
plt.xlabel('Recall Components', fontsize=12)
plt.ylabel('Count of Recalls', fontsize=12)
plt.xticks(fontsize=8, rotation=90)
plt.yticks(fontsize=10)
plt.tight_layout()
plt.subplots_adjust(bottom=0.2)
plt.show()
```



2.Contingency table: Displays the frequency of recalls for components against manufacturers. For Kia America, Inc., it shows the relationship between components and recall counts. Thus it helps in understanding how components vary by recall count across manufacturers.

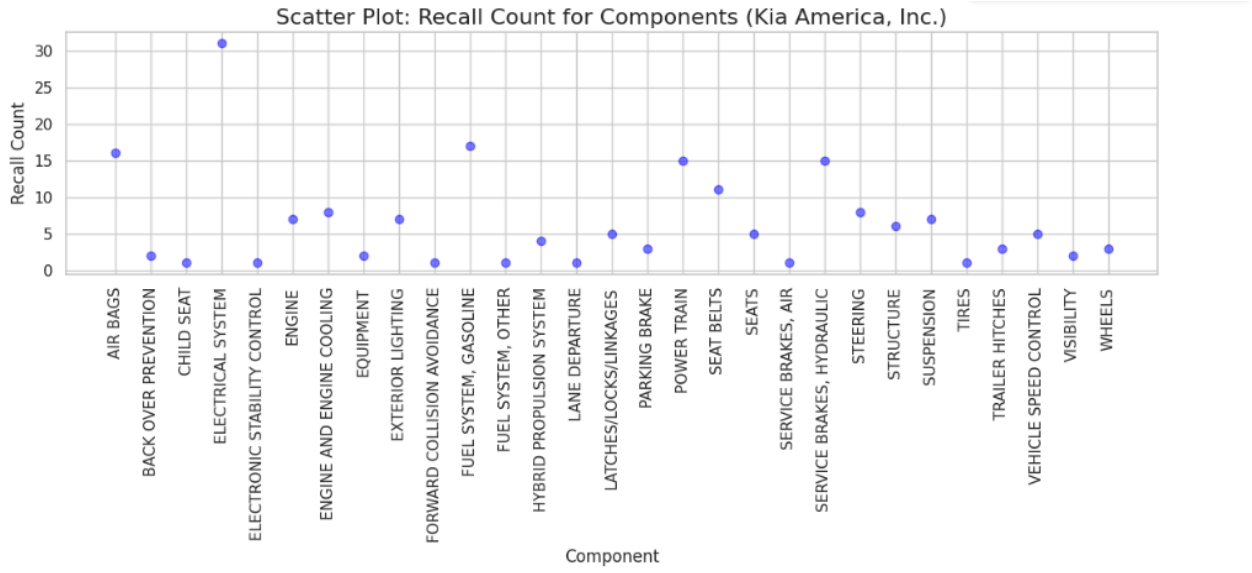
```
import pandas as pd
import matplotlib.pyplot as plt

# Assuming 'df' is your DataFrame

# Create a contingency table (crosstab) for Component vs Manufacturer
crosstb = pd.crosstab(df['Component'], df['Manufacturer'])

# Display the contingency table
print("Contingency Table:")
print(crosstb)
```

```
Contingency Table:
Manufacturer                                  1888653 Ontario Inc  \
Component
AIR BAGS                                                         0
BACK OVER PREVENTION                                             0
CHILD SEAT                                                       0
COMMUNICATION                                                    0
ELECTRICAL SYSTEM                                                0
ELECTRONIC STABILITY CONTROL                                     0
ELECTRONIC STABILITY CONTROL (ESC)                               0
ENGINE                                                           0
ENGINE AND ENGINE COOLING                                        0
EQUIPMENT                                                        1
EQUIPMENT ADAPTIVE/MOBILITY                                      0
EXTERIOR LIGHTING                                                0
FORWARD COLLISION AVOIDANCE                                      0
FUEL SYSTEM, DIESEL                                              0
FUEL SYSTEM, GASOLINE                                            0
FUEL SYSTEM, OTHER                                               0
HYBRID PROPULSION SYSTEM                                         0
INTERIOR LIGHTING                                                0
LANE DEPARTURE                                                   0
```

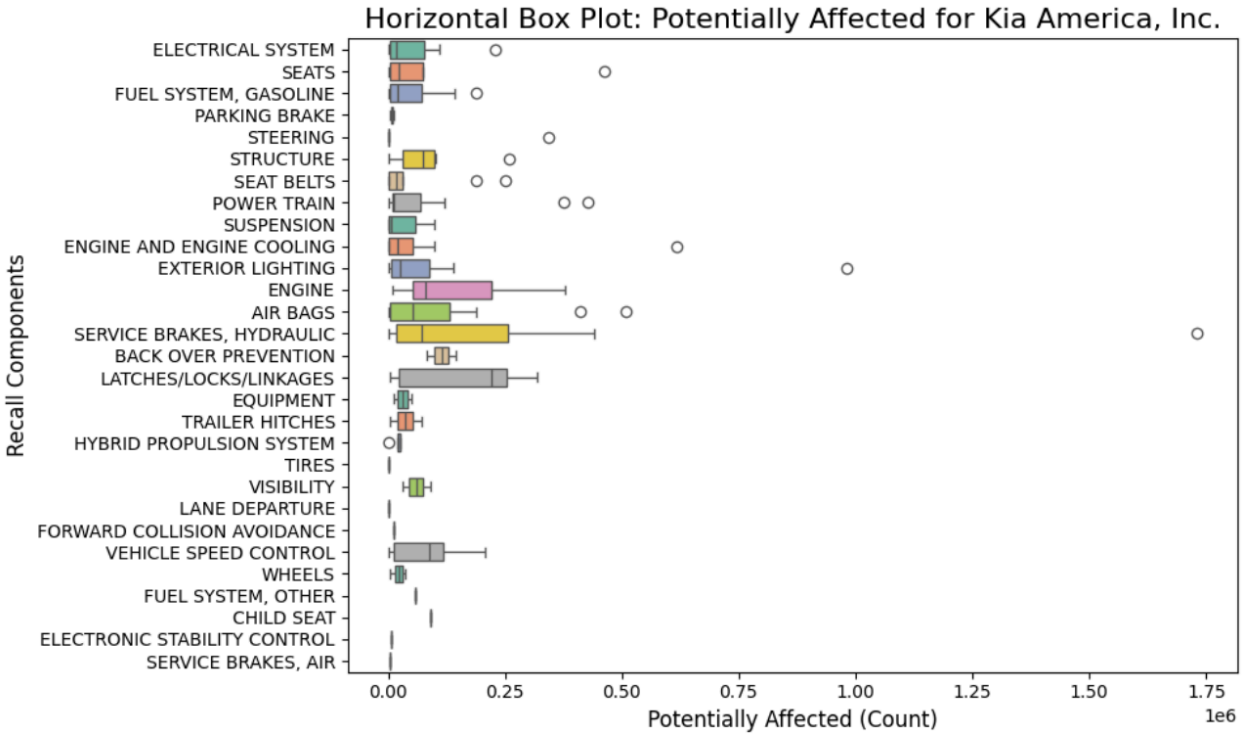
2. Plot Scatter plot, box plot, Heatmap using seaborn.

1. Scatter plot : This plot shows the recall count for various components of Kia America, Inc. Each point represents a component and its associated recall count.
Thus it highlights the distribution and patterns of recall counts, identifying outliers or components with extreme values.

```
import pandas as pd
import matplotlib.pyplot as plt
kia_df = df[df['Manufacturer'] == 'Kia America, Inc.']
crosstb_kia = pd.crosstab(kia_df['Component'], kia_df['Manufacturer'])
scatter_data = crosstb_kia.reset_index()
scatter_data = scatter_data.melt(id_vars=['Component'], value_vars=crosstb_kia.columns, var_name='Manufacturer', value_name='Recall Count')
plt.figure(figsize=(12, 6))
plt.scatter(scatter_data['Component'], scatter_data['Recall Count'], c='blue', alpha=0.5)
plt.title('Scatter Plot: Recall Count for Components (Kia America, Inc.)', fontsize=16)
plt.xlabel('Component', fontsize=12)
plt.ylabel('Recall Count', fontsize=12)
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



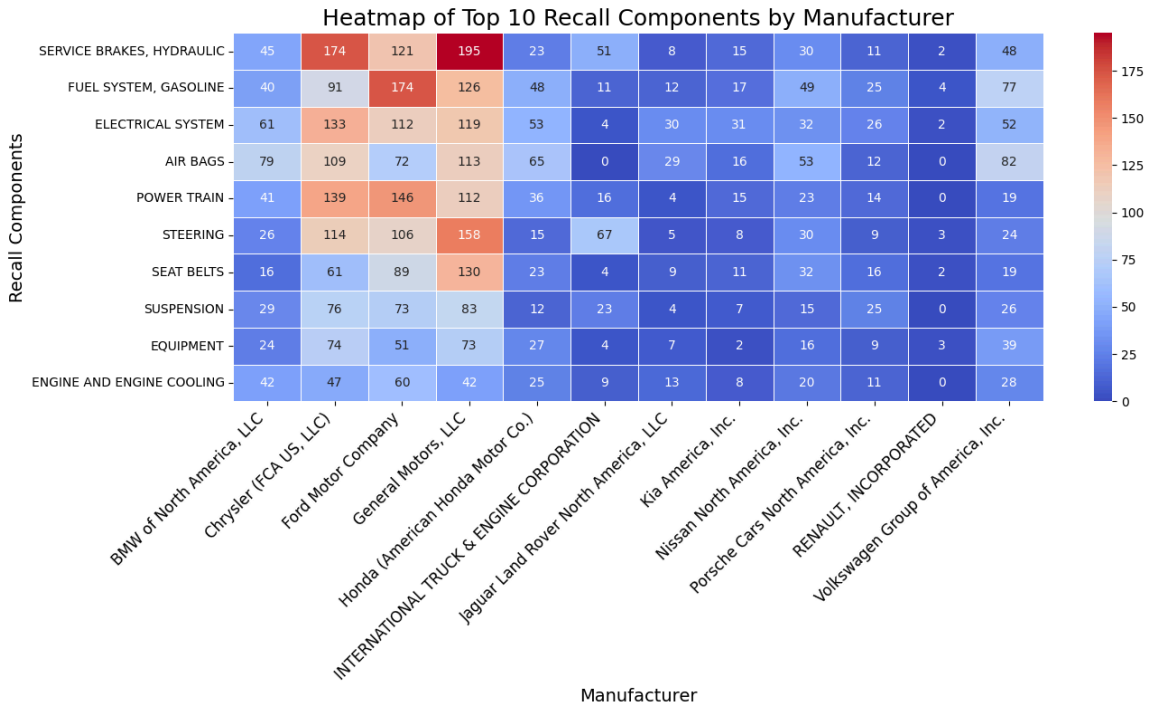
2. Box plot: The box plot shows the distribution of the number of potentially affected vehicles for each recalled component. The horizontal layout makes it easier to compare components.It identifies the spread, central tendency, and outliers in the data, with whiskers representing the data range and dots as potential outliers.




```python
#box plot
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

kia_df = df[df['Manufacturer'] == 'Kia America, Inc.']
plt.figure(figsize=(10, 6))
sns.boxplot(x='Potentially Affected', y='Component', data=kia_df, palette='Set2')
plt.title('Horizontal Box Plot: Potentially Affected for Kia America, Inc.', fontsize=16)
plt.xlabel('Potentially Affected (Count)', fontsize=12)
plt.ylabel('Recall Components', fontsize=12)
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)

plt.tight_layout()
plt.show()
```

3.Heat map: The heatmap visualizes the contingency table data, where the intensity of color represents the recall count for each component.Thus helps in identifying which components have a higher recall count at a glance, making it easy to spot patterns or correlations.

```
# Load data (ensure this is defined)
df = pd.read_csv('Recalls_Data.csv')  # Update with the correct file path

# Filter data for selected manufacturers
selected_manufacturers = [
    'Kia America, Inc.', 'Ford Motor Company', "Nissan North America, Inc.",'Mercedes-Benz
]

filtered_df = df[df['Manufacturer'].isin(selected_manufacturers)]

# Create crosstab for heatmap
heatmap_data = pd.crosstab(filtered_df['Component'], filtered_df['Manufacturer'])

# Fill NaN values with 0
heatmap_data = heatmap_data.fillna(0)

# Sort by total recalls and select only the top 10 components
top_10_components = heatmap_data.sum(axis=1).nlargest(10).index
heatmap_data = heatmap_data.loc[top_10_components]

# Plot heatmap
plt.figure(figsize=(14, 8))
sns.heatmap(heatmap_data, annot=True, cmap='coolwarm', fmt='d', linewidths=0.5)

# Labels and title
plt.title('Heatmap of Top 10 Recall Components by Manufacturer', fontsize=18)
plt.xlabel('Manufacturer', fontsize=14)
plt.ylabel('Recall Components', fontsize=14)
plt.xticks(fontsize=12, rotation=45, ha="right")
plt.yticks(fontsize=10)

plt.tight_layout()
plt.show()
```

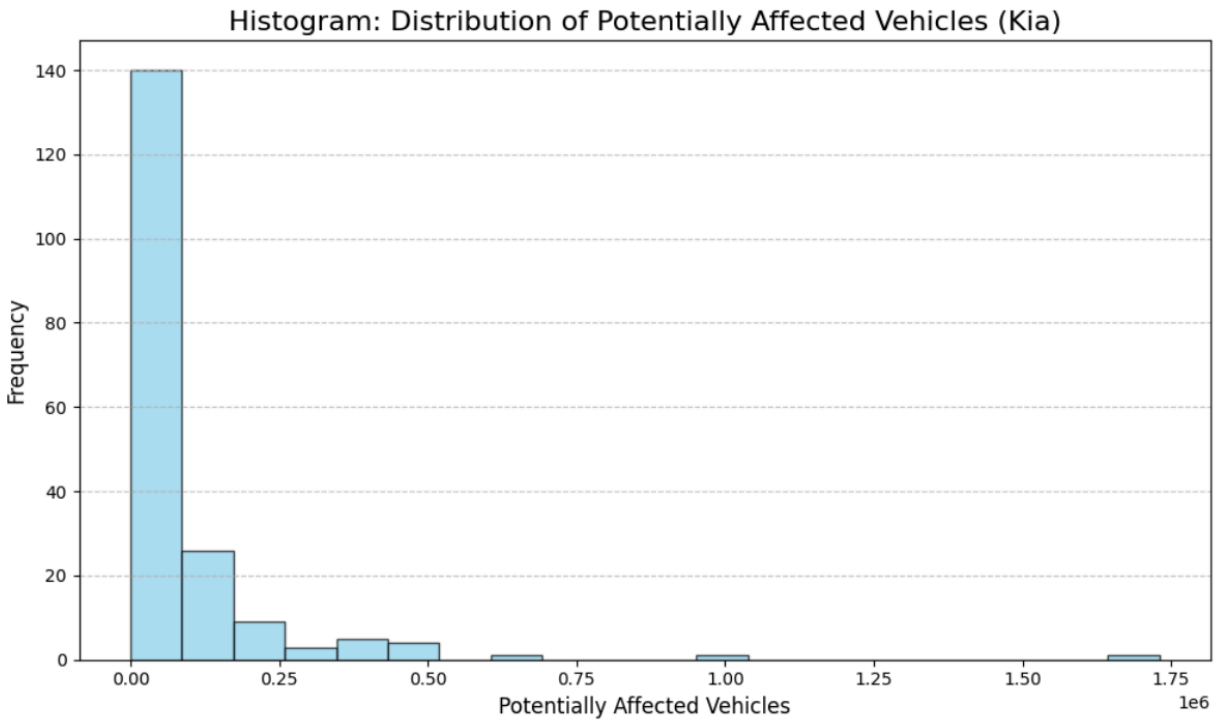


3. Create histogram and normalized Histogram.

The histogram shows the frequency distribution of the "Potentially Affected" vehicles.

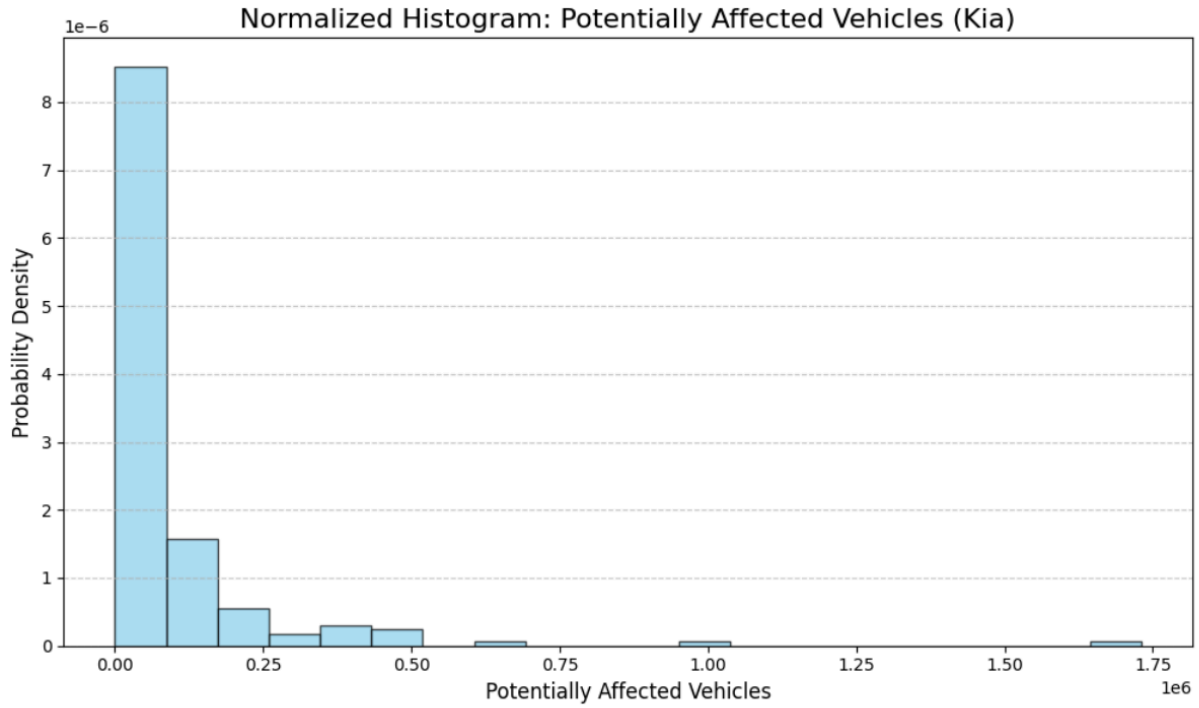
Regular Histogram: It displays counts in bins.

```
import pandas as pd, matplotlib.pyplot as plt
kia_df = df[df['Manufacturer'] == 'Kia America, Inc.']
plt.figure(figsize=(10,6))
plt.hist(kia_df['Potentially Affected'], bins=20, color='skyblue', edgecolor='black', alpha=0.7)
plt.title('Histogram: Distribution of Potentially Affected Vehicles (Kia)', fontsize=16)
plt.xlabel('Potentially Affected Vehicles', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



Normalized Histogram: It displays probability density.

```
import pandas as pd, matplotlib.pyplot as plt
kia_df = df[df['Manufacturer'] == 'Kia America, Inc.']
plt.figure(figsize=(10,6))
plt.hist(kia_df['Potentially Affected'], bins=20, color='skyblue', edgecolor='black', alpha=0.7, density=True)
plt.title('Normalized Histogram: Potentially Affected Vehicles (Kia)', fontsize=16)
plt.xlabel('Potentially Affected Vehicles', fontsize=12)
plt.ylabel('Probability Density', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



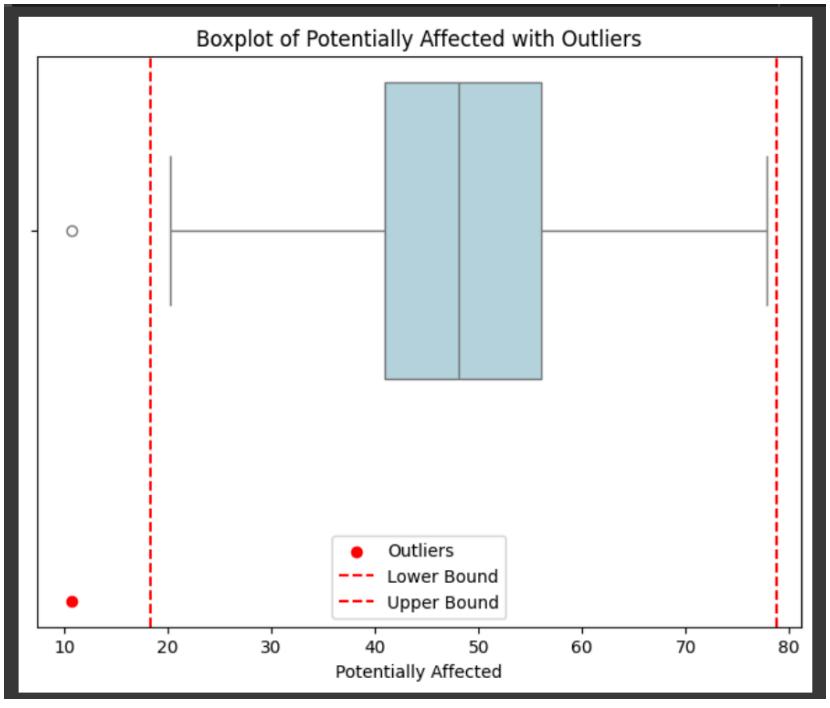
This highlights the data's skewness, central tendency, and spread. The normalized version shows the proportion of each bin relative to the total.

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

Outliers are values that deviate significantly from the rest of the data. Using the IQR method, data points beyond Q1−1.5×IQR or Q3+1.5×IQR are considered outliers.

Process: Outliers can be removed or replaced to reduce skewness and improve data accuracy.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
np.random.seed(42)
df = pd.DataFrame({'Potentially Affected': np.random.normal(50, 15, 100)})
col = 'Potentially Affected'
Q1 = df[col].quantile(0.25)
Q3 = df[col].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)]
plt.figure(figsize=(8, 6))
sns.boxplot(x=df[col], color='lightblue')
plt.scatter(outliers[col], [1] * len(outliers), color='red', label='Outliers', zorder=2)
plt.axvline(lower_bound, color='red', linestyle='dashed', label='Lower Bound')
plt.axvline(upper_bound, color='red', linestyle='dashed', label='Upper Bound')
plt.title(f'Boxplot of {col} with Outliers')
plt.legend()
plt.show()
```



Conclusion: Bar Graph and Scatter Plot are suitable for identifying trends and comparisons.Heatmap is excellent for understanding correlations between variables. Box Plot provides insights into distributions and outliers.Normalized Histogram helps understand probabilities and densities.Outlier Handling improves the accuracy of the results.