

Aim: 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.
2. Plot Scatter plot, box plot, Heatmap using seaborn.
3. Create histogram and normalized Histogram.
4. Describe what this graph and table indicates.
5. Handle outlier using box plot and Inter quartile range.

Steps:

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

Bar Graph: A bar graph will be plotted to visualize the relationship between a categorical feature (Class) and a numerical feature (Data_Value). The mean of Data_Value for each category will be represented.

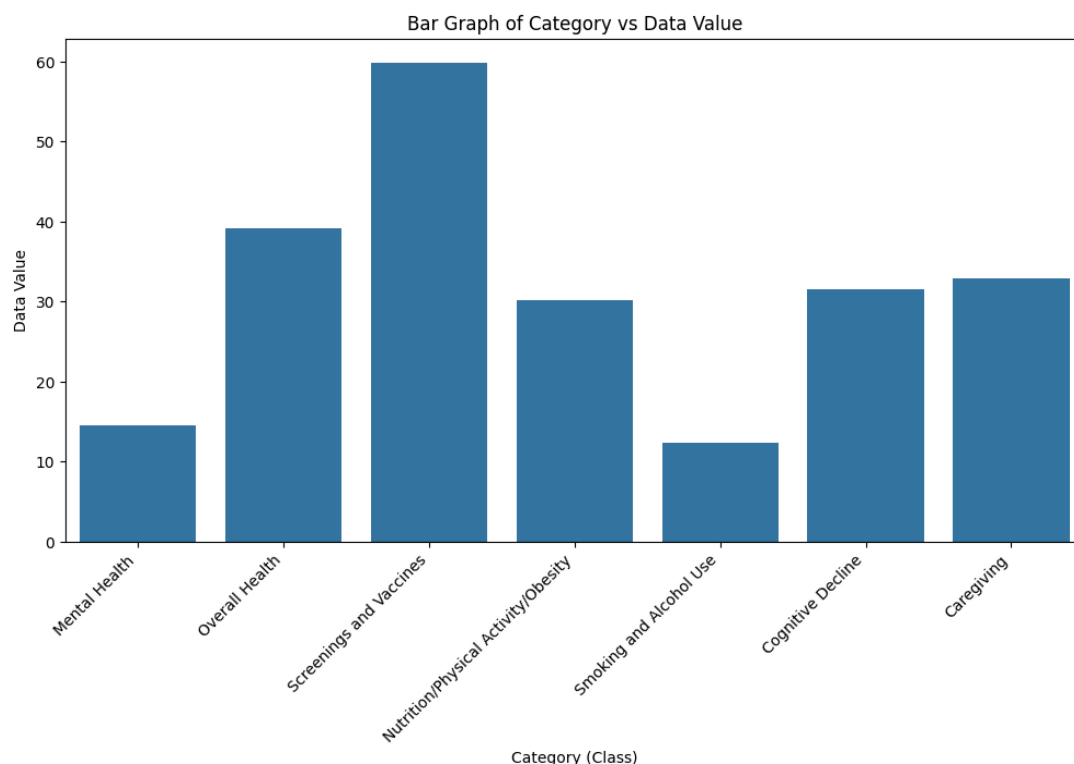
Contingency Table: A contingency table (cross-tabulation) will be generated using two categorical features (Region and Class), followed by a heatmap representation.

1) Bar graph:

Code:

```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(12,6))
sns.barplot(x=df["Class"], y=df["Data_Value"], estimator='mean', ci=None)
plt.xticks(rotation=45, ha='right')
plt.xlabel("Category (Class)")
plt.ylabel("Data Value")
plt.title("Bar Graph of Category vs Data Value")
plt.show()
```

Result:



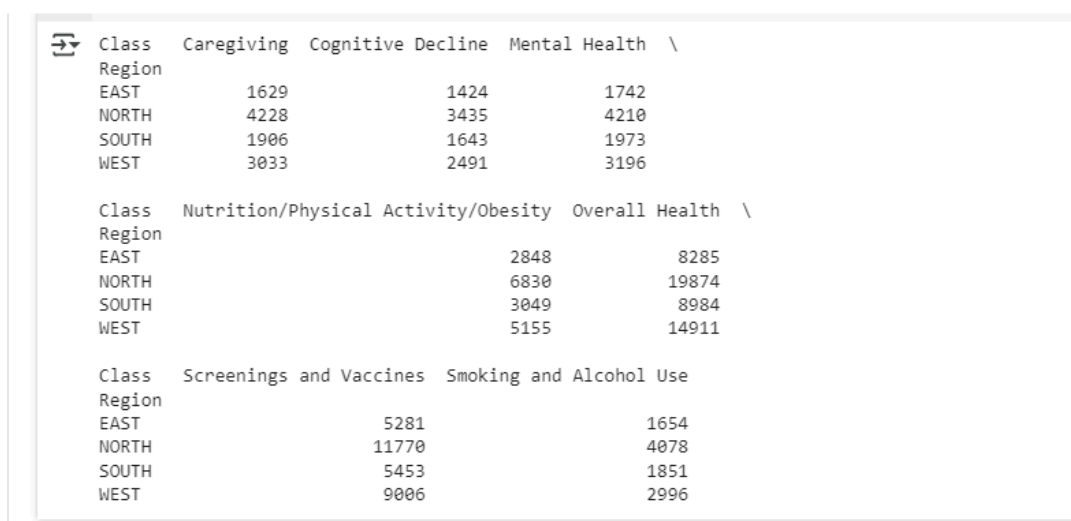
Inference: The bar graph compares different classes based on their Data_Value. Taller bars mean higher values for that class, while shorter bars mean lower values. This makes it easy to see which classes have higher or lower data points.

2) Contingency table:

Code:

```
contingency_table = df.groupby(["Region", "Class"]).size().unstack()
print(contingency_table)
```

Result:



Class	Caregiving	Cognitive Decline	Mental Health	\
Region				
EAST	1629	1424	1742	
NORTH	4228	3435	4210	
SOUTH	1906	1643	1973	
WEST	3033	2491	3196	

Class	Nutrition/Physical Activity/Obesity	Overall Health	\
Region			
EAST	2848	8285	
NORTH	6830	19874	
SOUTH	3049	8984	
WEST	5155	14911	

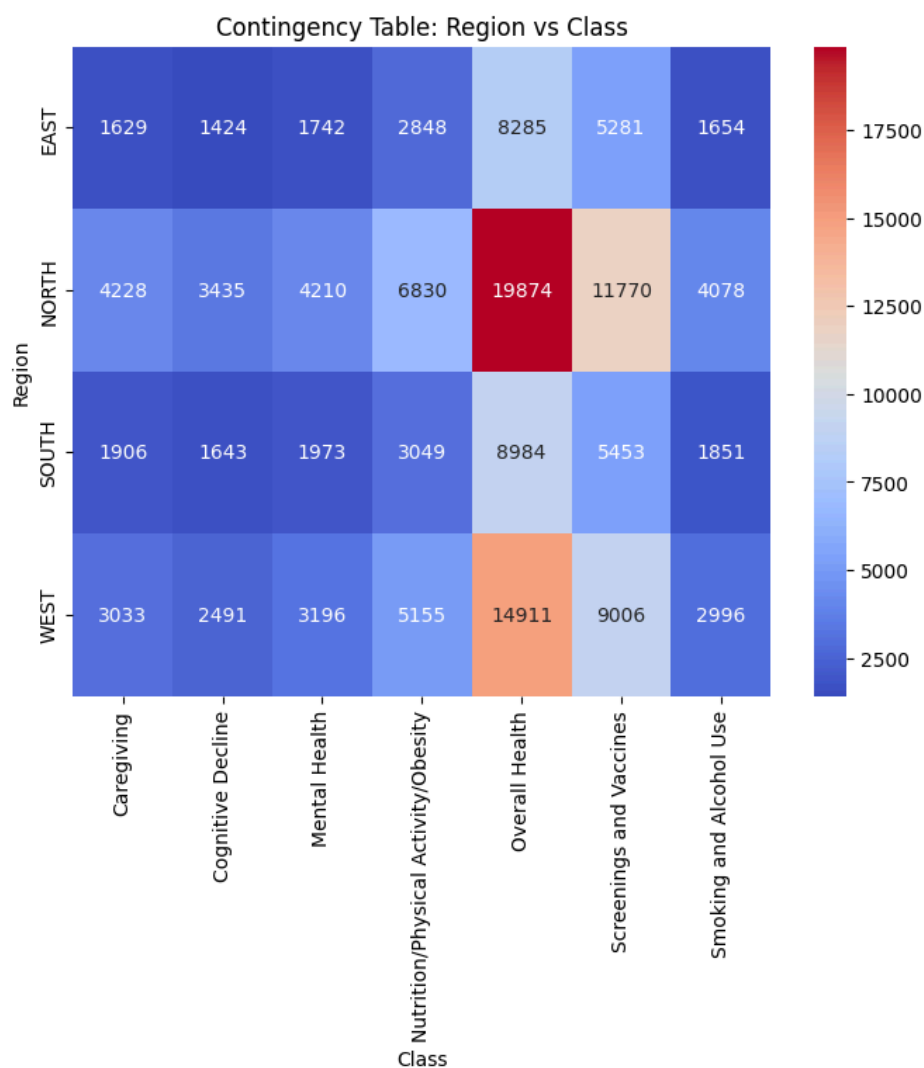
Class	Screenings and Vaccines	Smoking and Alcohol Use	\
Region			
EAST	5281	1654	
NORTH	11770	4078	
SOUTH	5453	1851	
WEST	9006	2996	

Code:

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

plt.figure(figsize=(8,6))
sns.heatmap(contingency_table, annot=True, cmap="coolwarm", fmt='d')
plt.title("Contingency Table: Region vs Class")
plt.xlabel("Class")
plt.ylabel("Region")
plt.show()
```

Result:



Inference: The contingency table helps compare two categories, Region and Class, by showing how often they appear together. It helps find patterns, like whether certain regions have more cases of a particular class. This makes it useful for understanding relationships between different groups.

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

Scatter Plot: A scatter plot will be used to show the relationship between two numerical features.

Box Plot: The distribution of Data_Value will be analyzed using a box plot, which also helps in identifying outliers.

Heatmap: A heatmap will be created to visualize correlations between numerical features in the dataset.

Box Plot:

Code:

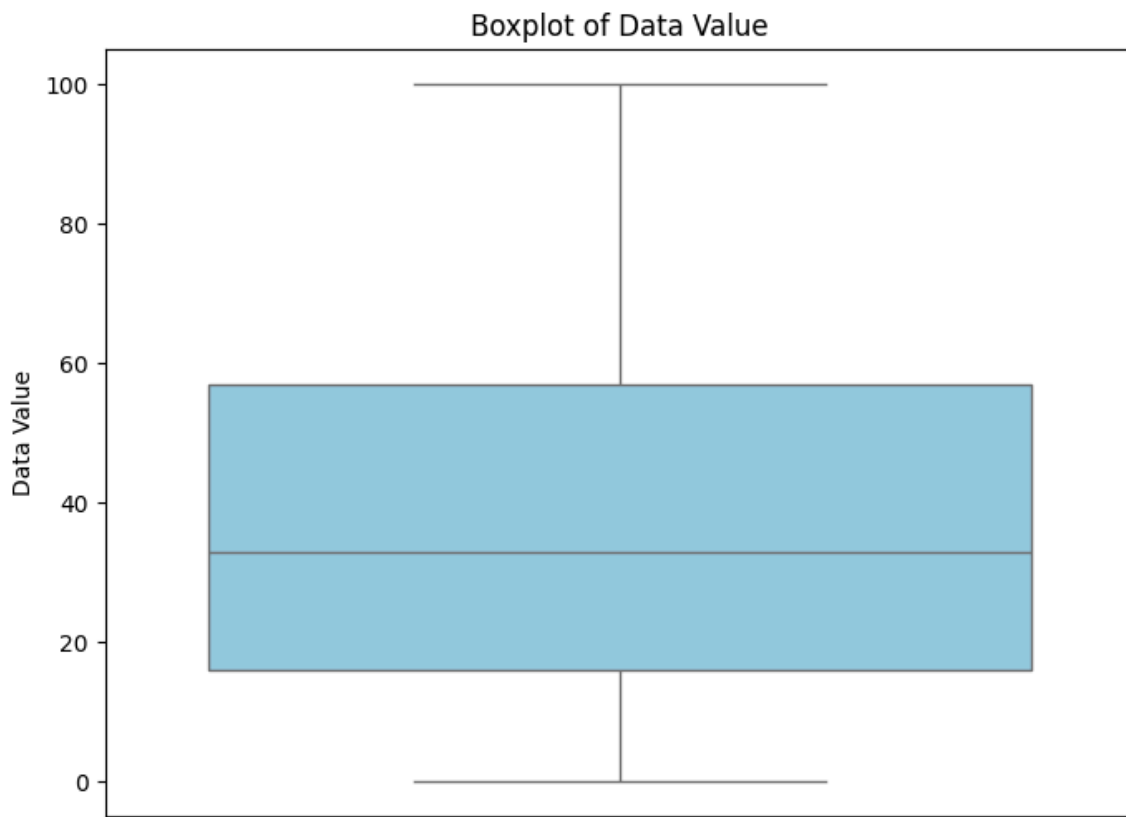
```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
plt.figure(figsize=(8,6))
sns.boxplot(y=df["Data_Value"], color="skyblue")

plt.ylabel("Data Value")
plt.title("Boxplot of Data Value")

plt.show()
```

Result:



Inference: The boxplot gives a summary of how Data_Value is spread and helps identify extreme values. The box represents the middle 50% of the data, while the whiskers extend to the minimum and maximum within a range. Any points outside this range are considered outliers, showing if the data has any unusual values.

Step 3: Create histogram and normalized Histogram.

Histogram: A histogram will be plotted to analyze the frequency distribution of Data_Value.

Normalized Histogram: A probability density histogram will be created to normalize the distribution and verify its total area.

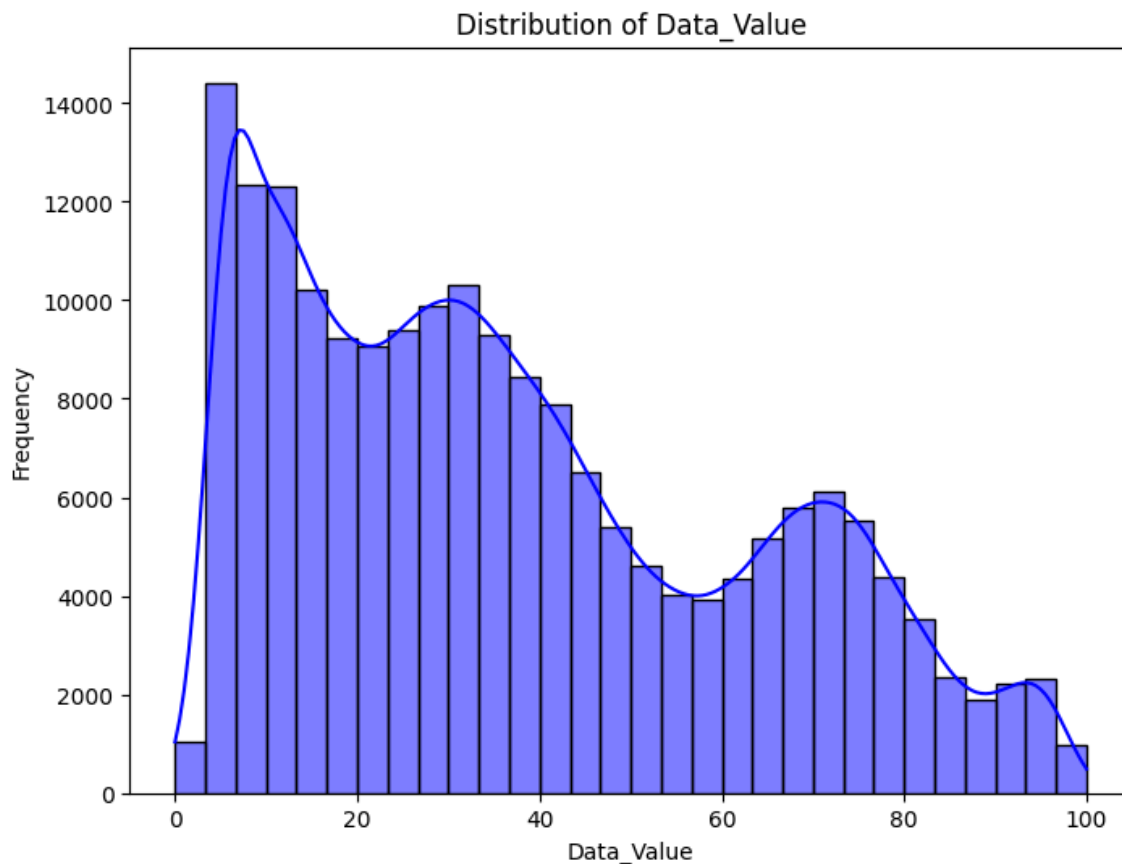
1)Histogram:

Code:

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

```
plt.figure(figsize=(8,6))
sns.histplot(df["Data_Value"], bins=30, kde=True, color="blue") # KDE for smooth
curve
plt.title("Distribution of Data_Value")
plt.xlabel("Data_Value")
plt.ylabel("Frequency")
plt.show()
```

Result:



Inference: The frequency distribution graph shows how often different Data_Value ranges appear in the dataset. If certain ranges have higher bars, it means more data points fall within them. This helps in spotting patterns, such as clusters of values or unusual gaps.

2) Normalized Histogram:

Code:

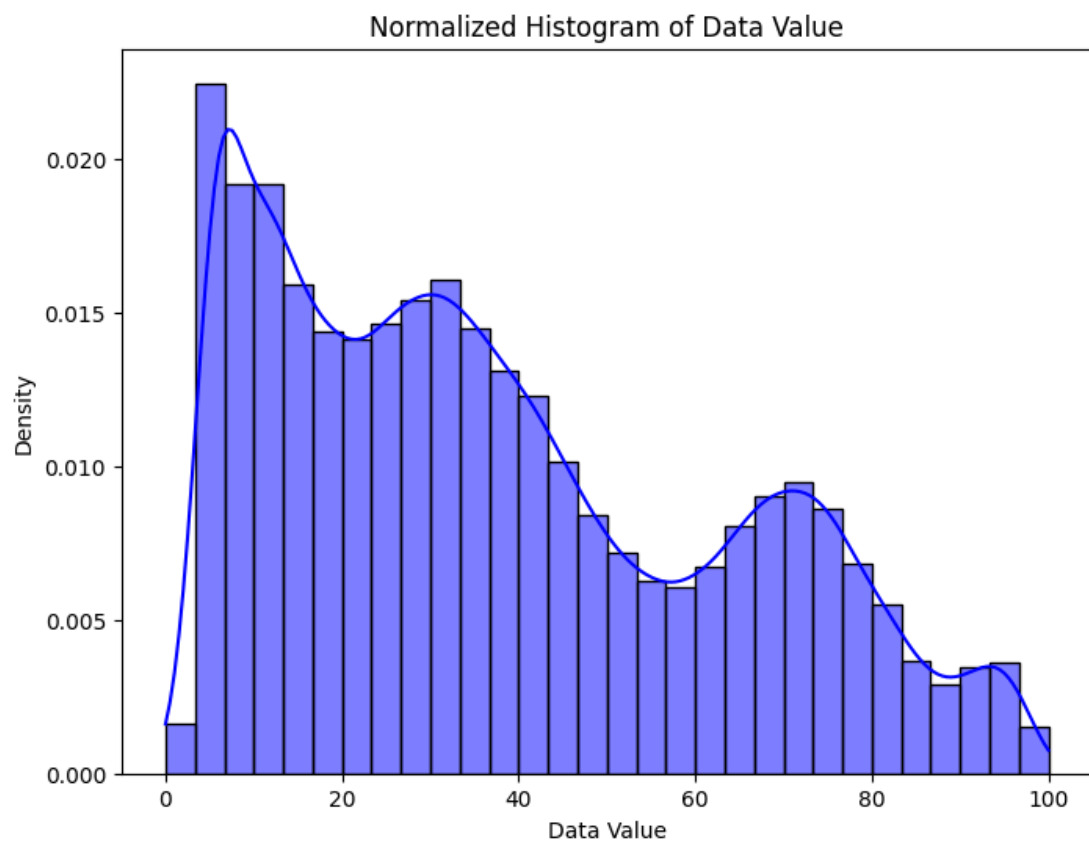
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Plot histogram and store bin heights and edges
plt.figure(figsize=(8,6))
hist_data = sns.histplot(df["Data_Value"], bins=30, kde=True, stat='density',
color="blue")
```

```
# Extract heights of bars
heights = [patch.get_height() for patch in hist_data.patches]

# Extract bin widths
bin_width = hist_data.patches[1].get_x() - hist_data.patches[0].get_x()
# Compute total area
area = sum(heights) * bin_width
print("Total Area of Normalized Histogram:", area)
plt.xlabel("Data Value")
plt.ylabel("Density")
plt.title("Normalized Histogram of Data Value")
plt.show()
```

Result:



Inference: The normalized histogram shows how Data_Value is distributed across different ranges. The x-axis represents the values, while the y-axis shows their probability, making sure the total area equals one. This helps in understanding whether the data follows a normal pattern or is unevenly spread.

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

Box Plot Analysis: Outliers will be detected using box plots.

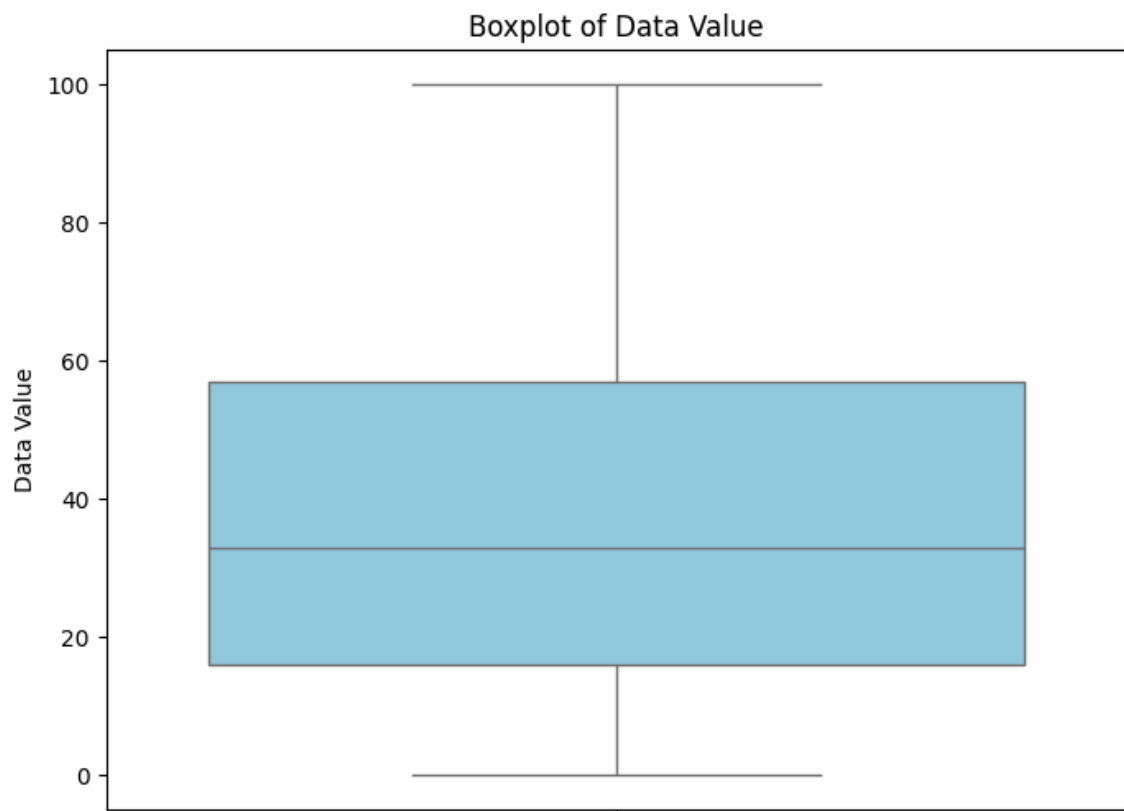
Interquartile Range (IQR) Method: The IQR method will be applied to determine the lower and upper bounds for identifying and handling outliers.

1) Box Plot:

Code:

```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(8,6))
sns.boxplot(y=df["Data_Value"], color="skyblue")
plt.ylabel("Data Value")
plt.title("Boxplot of Data Value")
plt.show()
```

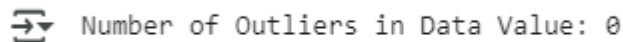
Result:



Inference: The boxplot gives a summary of how Data_Value is spread and helps identify extreme values. The box represents the middle 50% of the data, while the whiskers extend to the minimum and maximum within a range. Any points outside this range are considered outliers, showing if the data has any unusual values.

2) Interquartile Range:**Code:**

```
Q1 = df['Data_Value'].quantile(0.25)
Q3 = df['Data_Value'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
outliers = df[(df['Data_Value'] < lower_bound) | (df['Data_Value'] >
upper_bound)]
print("Number of Outliers in Data Value:", len(outliers))
```

Result:The image shows a terminal window with a light blue border. Inside, there is a green icon of a terminal with a cursor, followed by the text "Number of Outliers in Data Value: 0".

```
➤ Number of Outliers in Data Value: 0
```

Inference:The inference from the Interquartile Range (IQR) analysis is that the dataset does not contain any significant outliers. This suggests that the data values are well-distributed within the expected range and do not have extreme deviations. Since outliers can impact statistical analysis and model performance, the absence of outliers indicates that the dataset is stable and reliable for further analysis without requiring additional preprocessing to handle extreme values.

Conclusion: This analysis helps in understanding the distribution, relationships, and anomalies in the dataset. The findings will provide meaningful insights for further decision-making and model building. The exploratory data analysis (EDA) revealed significant insights into the dataset's distribution, relationships, and anomalies. The bar graph and contingency table highlighted categorical trends, while scatter plots and heatmaps uncovered correlations between numerical features. Box plots and histograms provided a deeper understanding of data distribution, identifying potential outliers. Using the Interquartile Range (IQR) method, outliers were detected and handled to improve data quality. Overall, this analysis helped in feature selection, pattern recognition, and data preprocessing, ensuring a more refined dataset for future modeling and decision-making.