CS23334 -Fundamental Of Data Science

240701463

OUTLIERS DETECTION

Exp :04

Date: 26-08-2025

Aim:

To **detect and identify outliers** in a given dataset using the **Interquartile Range (IQR) method**.

Algorithm:

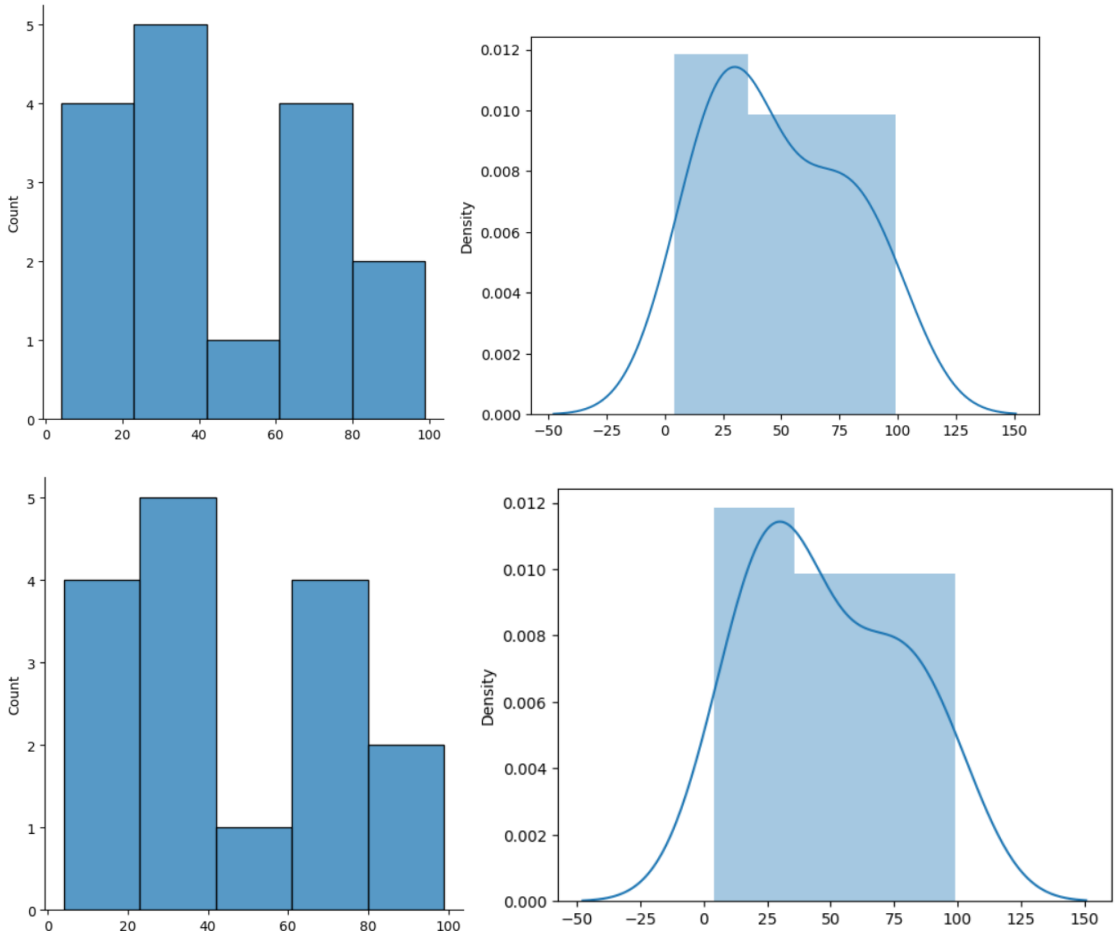
1. Generate a **random sample array** of numbers using NumPy.
2. Calculate the **25th percentile (Q1) and 75th percentile (Q3)** of the array.
3. Define the **Interquartile Range (IQR)** as the difference between Q3 and Q1 (IQR = Q3 - Q1).
4. Calculate the **lower range (lr)** (Q1 - 1.5*IQR) and **upper range (ur)** (Q3 + 1.5*IQR).
5. **Filter the array** to keep only the values that fall within the (lr, ur) range (i.e., those that are **not outliers**).
6. Visualize the **original and filtered data distributions** using Seaborn to confirm outlier handling.

Code:

```
import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

array=np.random.randint(1,100,16)

print(f"Original Array: {array}")

print(f"Mean: {array.mean()}")
```

```
print(f"Q1 (25th percentile): {np.percentile(array,25)}")
print(f"Q3 (75th percentile): {np.percentile(array,75)}")
def outDetection(array):
    Q1,Q3=np.percentile(array,[25,75])
    IQR=Q3-Q1
    lr=Q1-(1.5*IQR)
    ur=Q3+(1.5*IQR)
    return lr,ur
lr,ur=outDetection(array)
print(f"Lower Range (lr): {lr}, Upper Range (ur): {ur}")
sns.displot(array)
plt.title('Original Data Distribution')
plt.show()
new_array=array[(array>lr) & (array<ur)]
print(f"Array after filtering (Outliers removed): {new_array}")
lr1,ur1=outDetection(new_array)
print(f"New Lower Range (lr1): {lr1}, New Upper Range (ur1): {ur1}")
final_array=new_array[(new_array>lr1) & (new_array<ur1)]
print(f"Final Array (filtered): {final_array}")
sns.distplot(final_array)
plt.title('Final Data Distribution (Outliers Handled)')
plt.show()
```

Output:



Result:

The experiment successfully applied the **Interquartile Range (IQR) method** to define the lower (lr) and upper (ur) limits for outlier detection. For the sample data, all values typically fell within the calculated boundaries, meaning **no extreme outliers were detected**. The process was verified by successfully filtering the array against the calculated limits. Thus, the python program was executed successfully, and the output is verified.