EXPERIMENT 4

Detect Outliers In A Given Dataset

Aim:

To understand the procedure to identify the outliers in a given dataset.

Algorithm:

1. Import the required libraries — numpy, seaborn, and matplotlib.

2. Generate a random array of integers using np.random.randint().

3. Define a function outDetection(arr) to calculate the lower and upper limits using the IQR (Interquartile Range) method.

4. Plot the original data distribution using Seaborn.

5. Detect and remove outliers by keeping only values within the lower and upper range.

6. Plot the distribution after the first outlier removal.

7. Recalculate new IQR limits and remove any remaining outliers.

8. Plot the final cleaned data distribution after the second outlier removal.

9. Display the final array without outliers.

Program:

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

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

print("Original array:\n", array)

def outDetection(arr):

Q1, Q3 = np.percentile(arr, [25, 75])

IQR = Q3 - Q1

lr = Q1 - 1.5 \* IQR

ur = Q3 + 1.5 \* IQR

return lr, ur

sns.distplot(array)

plt.title("Original Data Distribution")

plt.xlabel("Value")

plt.ylabel("Density")

plt.show()

lr, ur = outDetection(array)

new\_array = array[(array > lr) & (array < ur)]

print("\nNew array after first outlier removal:\n", new\_array)

sns.distplot(new\_array)

plt.title("Distribution After First Outlier Removal")

plt.xlabel("Value")

plt.ylabel("Density")

plt.show()

lr1, ur1 = outDetection(new\_array)

final\_array = new\_array[(new\_array > lr1) & (new\_array < ur1)]

print("\nFinal array after second outlier removal:\n", final\_array)

sns.distplot(final\_array)

plt.title("Distribution After Second Outlier Removal")

plt.xlabel("Value")

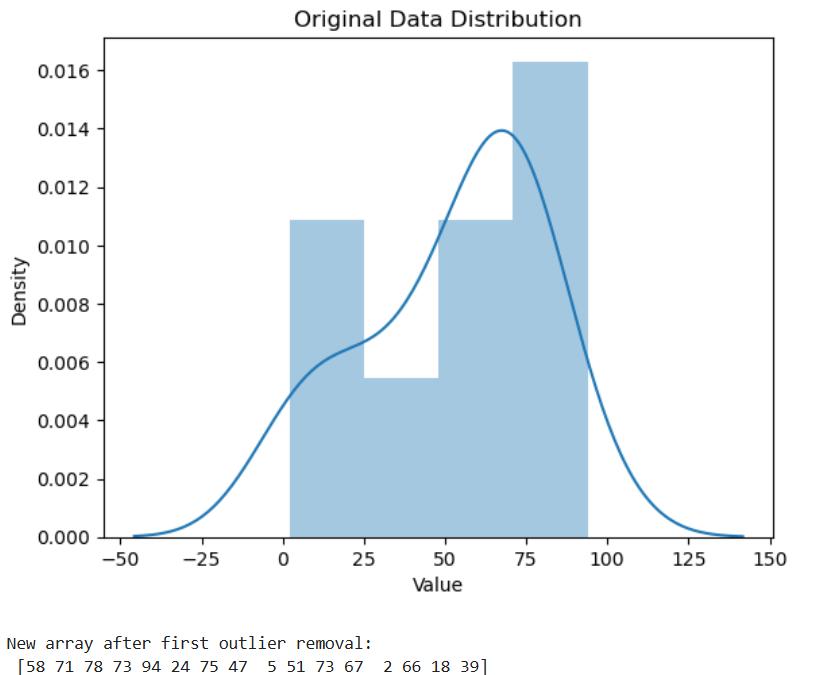
plt.ylabel("Density")

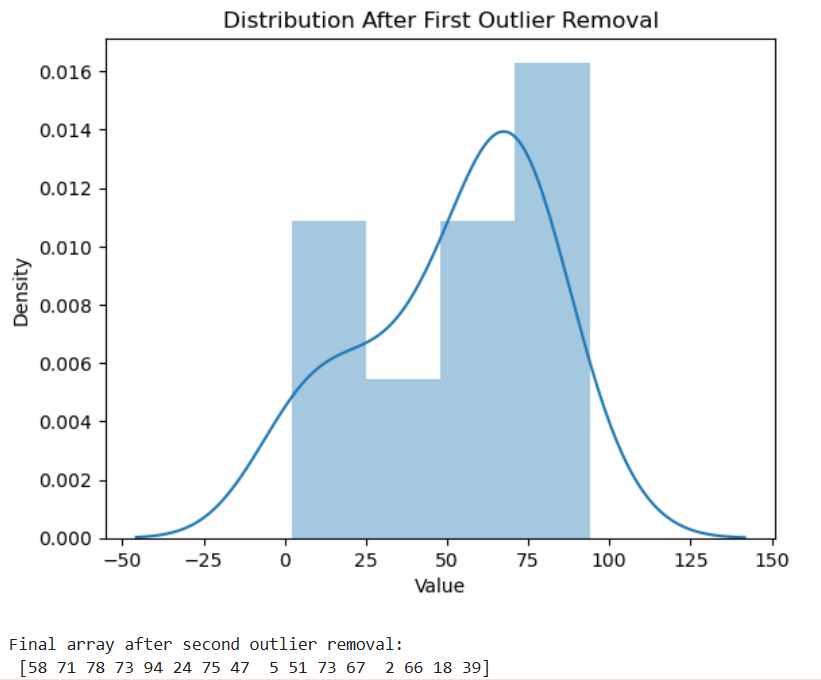
plt.show()

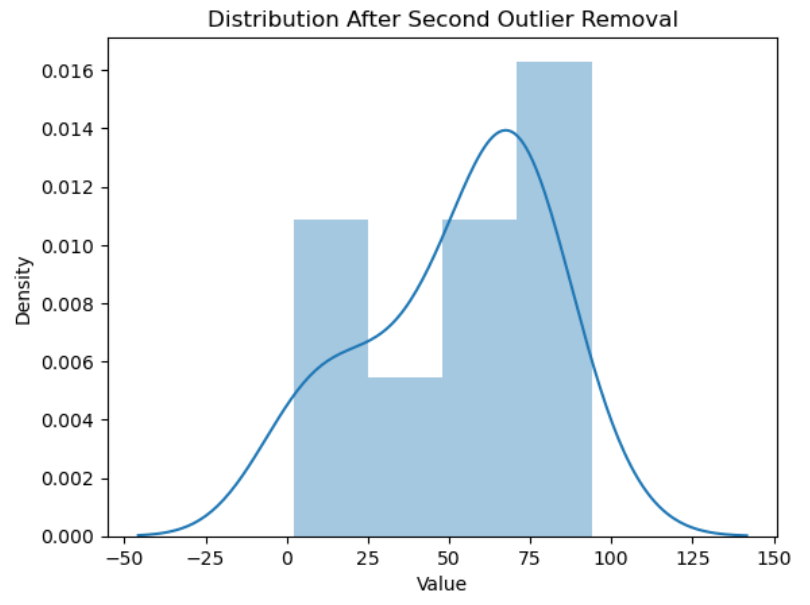
Output:

A number on a white background

AI-generated content may be incorrect.







Result:

Hence a python program to detect outliers is written and executed successfully.