EXPERIMENT NO: 4

Outlier Detection and Visualization using IQR

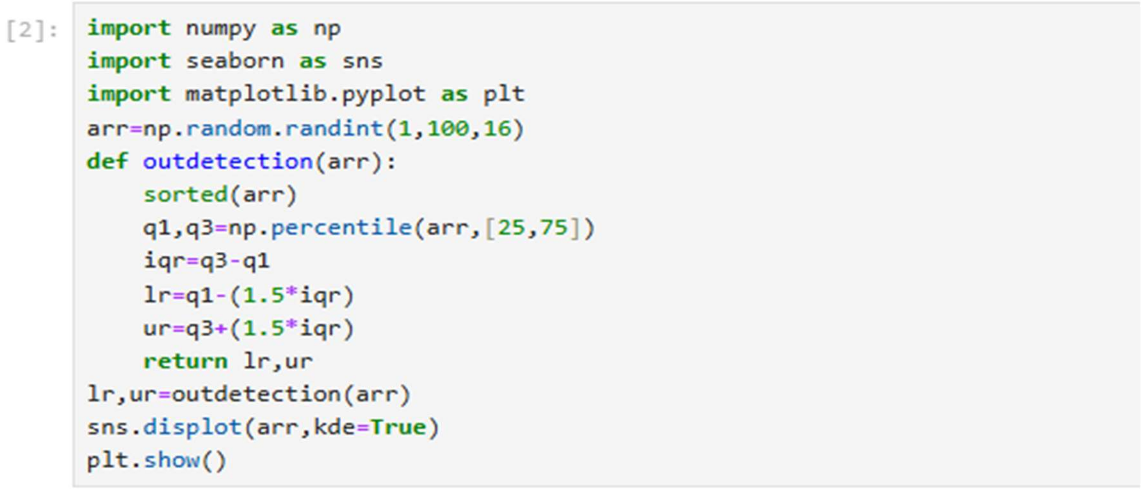
Aim:

To detect and remove outliers from a dataset using the Interquartile Range (IQR) method and visualize the data distribution before and after outlier removal.

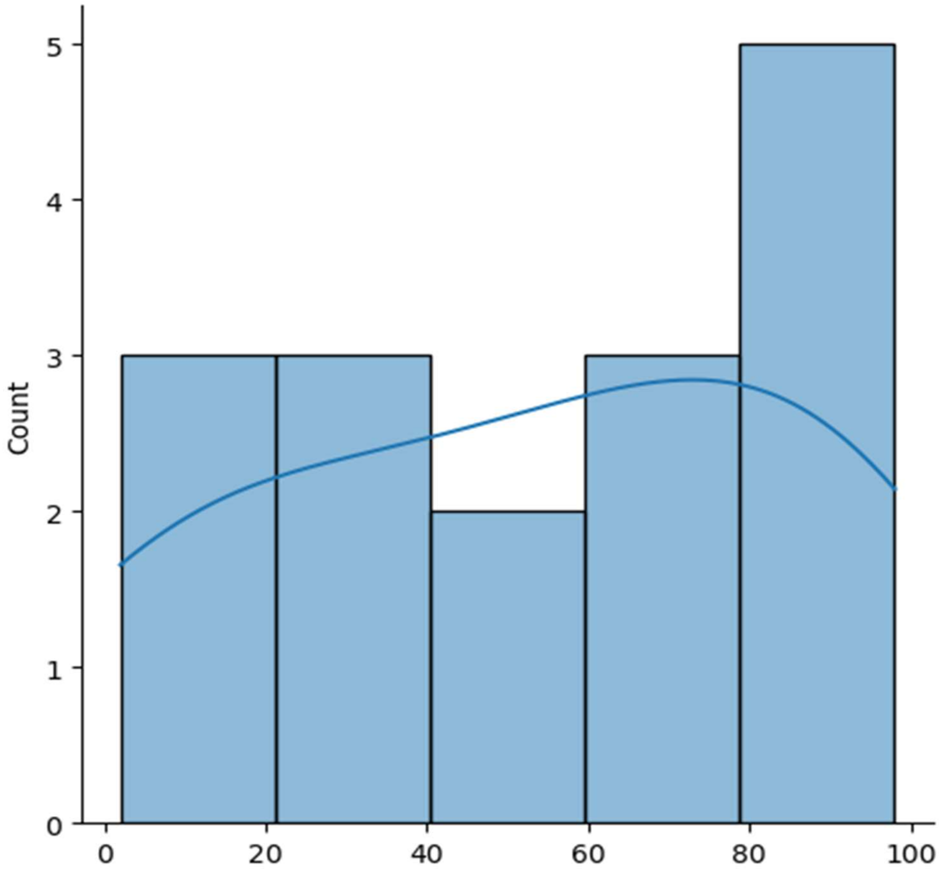
Algorithm:

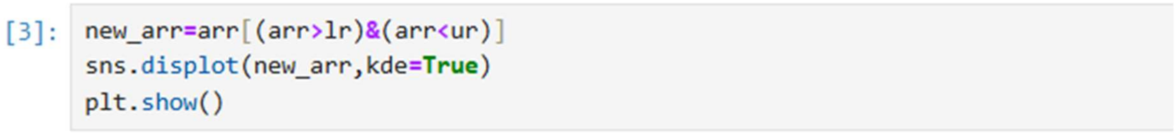
1) **Generate Data:** Create a random array of 16 integers between 1 and 100.
2) **Define Outlier Detection Function:**
    a. Compute Q1 (25th percentile) and Q3 (75th percentile).
    b. Calculate IQR = Q3 − Q1.
    c. Determine lower and upper bounds: `lr = Q1 - 1.5*IQR`, `ur = Q3 + 1.5*IQR`.
3) **Visualize Original Data:** Plot distribution with KDE.
4) **First Outlier Removal:** Filter values within `[lr, ur]` and plot the distribution.
5) **Second Outlier Removal:** Apply IQR again on filtered data and plot final distribution.

Program:

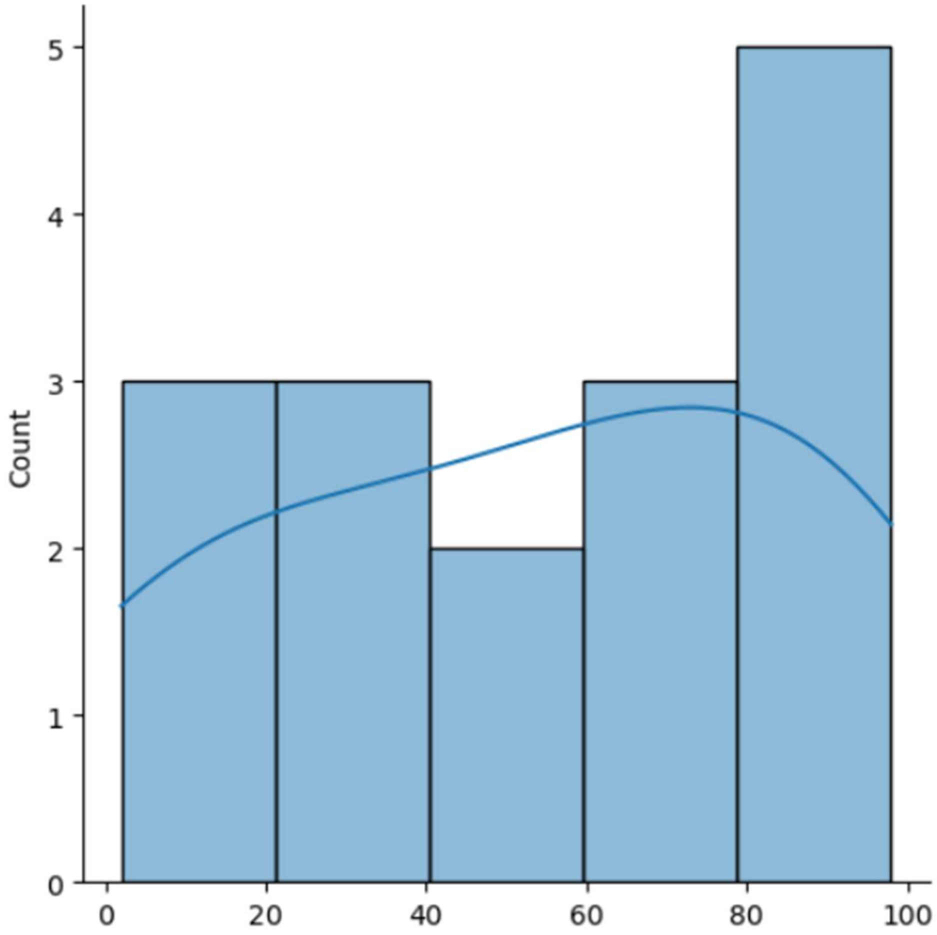


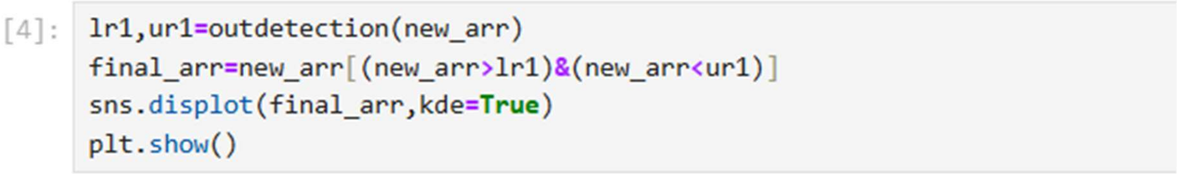
```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
arr=np.random.randint(1,100,16)
def outdetection(arr):
    sorted(arr)
    q1,q3=np.percentile(arr,[25,75])
    iqr=q3-q1
    lr=q1-(1.5*iqr)
    ur=q3+(1.5*iqr)
    return lr,ur
lr,ur=outdetection(arr)
sns.displot(arr,kde=True)
plt.show()
```



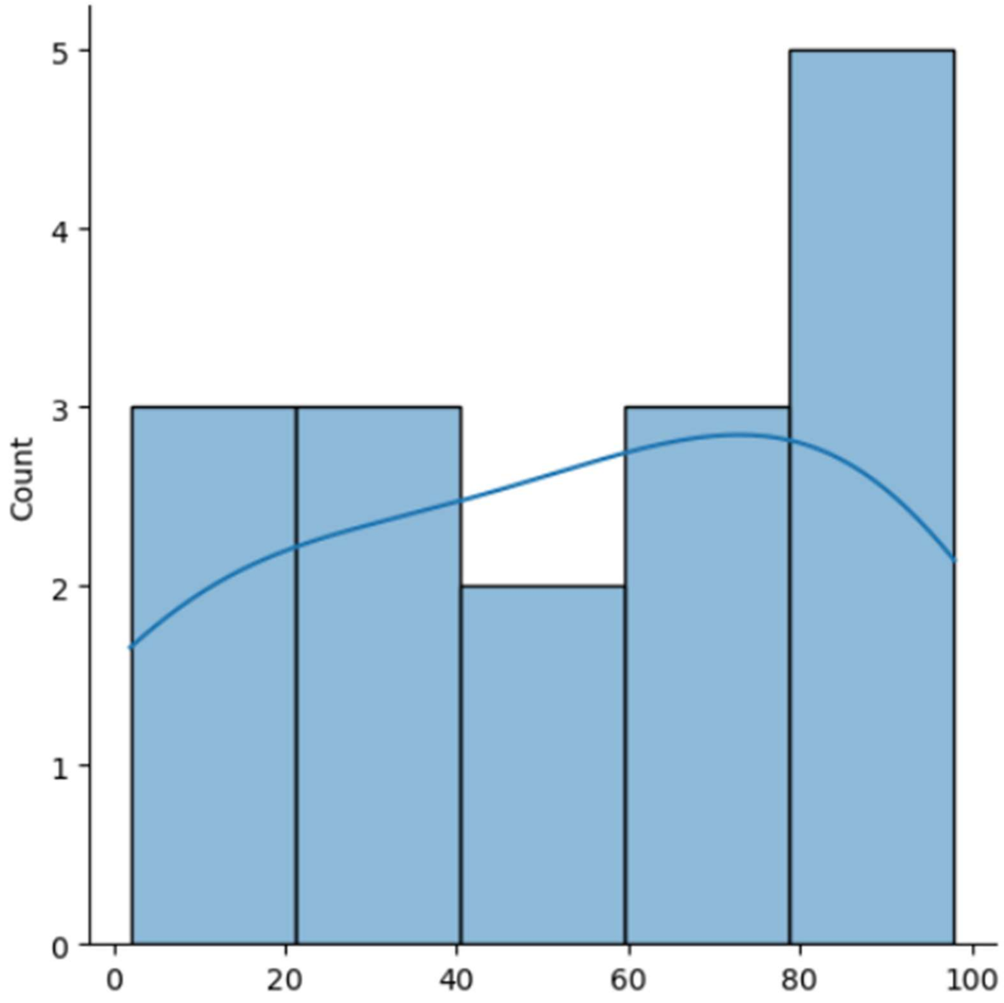


```
[3]: new_arr=arr[(arr>lr)&(arr<ur)]
     sns.displot(new_arr,kde=True)
     plt.show()
```





```
[4]: lr1,ur1=outdetection(new_arr)
     final_arr=new_arr[(new_arr>lr1)&(new_arr<ur1)]
     sns.displot(final_arr,kde=True)
     plt.show()
```



Result:

The code removes extreme values from a random dataset using IQR, showing three plots: original data, after first outlier removal, and after final outlier removal, resulting in a cleaner, more representative dataset.