EXPERIMENT:4 OUTLIERS

Aim:

To understand and identify outliers in a dataset using statistical and visualization techniques.

Procedure:

1. Load a sample dataset containing numerical values (e.g., student marks, salaries, or ages).
2. Use **descriptive statistics** like mean, median, and standard deviation to understand data distribution.
3. Apply the **Interquartile Range (IQR) method** to detect outliers:
   - Calculate Q1 (25th percentile) and Q3 (75th percentile).
   - Compute IQR = Q3 - Q1 and find values outside **[Q1 - 1.5×IQR, Q3 + 1.5×IQR]**.
4. Visualize the data using **boxplot** or **scatter plot** to highlight outliers.
5. Optionally, remove or replace outliers and observe the effect on analysis or model accuracy.

In [21]:

```
import numpy as np
array=np.random.randint(1,100,16)
array
```

Out[21]:

```
array([36, 64, 70, 22, 69, 66, 50, 10,  4, 86, 47,  2, 36, 58, 71, 60])
```

In [22]:

```
array.mean()
```

Out[22]:

```
46.9375
```

In [23]:

```
np.percentile(array,25)
```

Out[23]:

```
32.5
```

In [24]:

```
np.percentile(array,50)
```

Out[24]:

```
54.0
```

In [25]:

```
np.percentile(array,75)
```

Out[25]:

```
66.75
```

In [26]:

EXPERIMENT:4 OUTLIERS

```
np.percentile(array,100)
```

Out[26]:

```
86.0
```

In [32]:

```
def outDetection(array):
    sorted(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)
lr,ur
```

Out[32]:

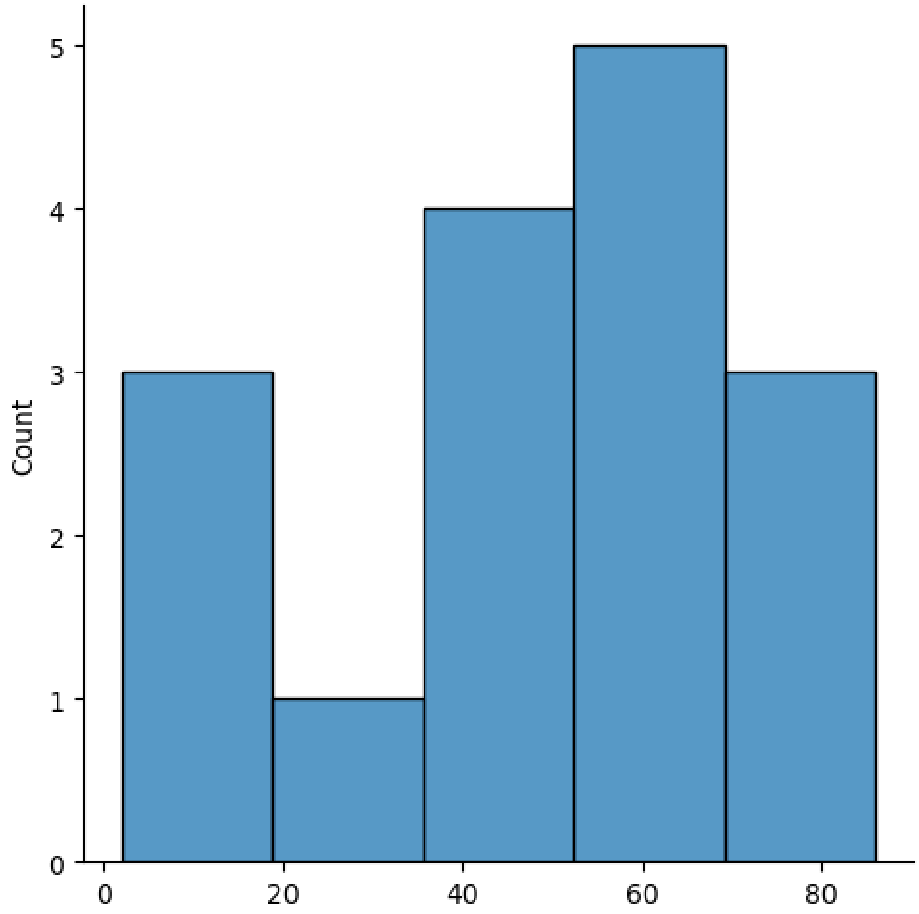
```
(-18.875, 118.125)
```

In [33]:

```
import seaborn as sns
%matplotlib inline
sns.displot(array)
```

Out[33]:

```
<seaborn.axisgrid.FacetGrid at 0x20a7f05ba90>
```



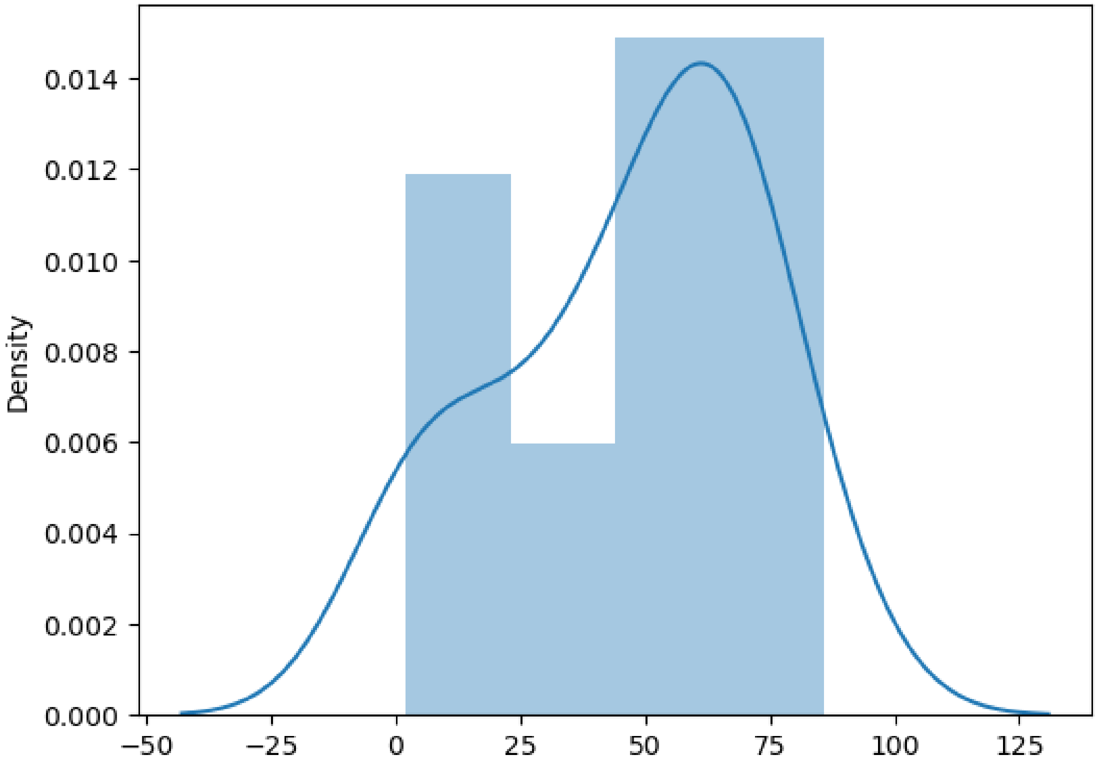
EXPERIMENT:4 OUTLIERS

In [34]:

```
sns.distplot(array)
C:\Users\kaviy\AppData\Local\Temp\ipykernel_16172\1133588802.py:1: 
UserWarning: 

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function 
with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(array)
```

Out[34]:

```
<Axes: ylabel='Density'>
```



In [40]:

```
new_array=array[(array>lr) & (array<ur)]
new_array
```

Out[40]:

```
array([36, 64, 70, 22, 69, 66, 50, 10,  4, 86, 47,  2, 36, 58, 71, 60])
```

In [41]:

```
lr1,ur1=outDetection(new_array)
```

```
lr1,ur1
```

Out[41]:

```
(-18.875, 118.125)
```

In [39]:

```
final_array=new_array[(new_array>lr1) & (new_array<ur1)]
final_array
```

Out[39]:

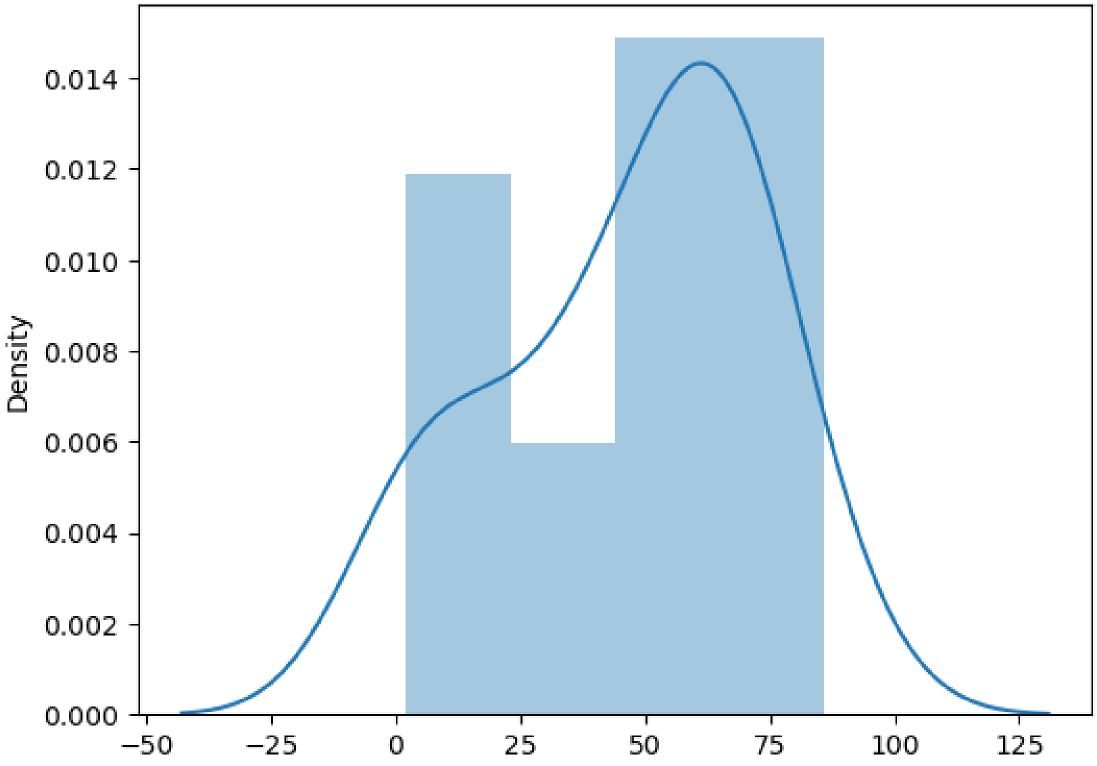
```
array([36, 64, 70, 22, 69, 66, 50, 10,  4, 86, 47,  2, 36, 58, 71, 60])
```

In [42]:

```
sns.distplot(final_array)
C:\Users\kaviy\AppData\Local\Temp\ipykernel_16172\209491988.py:1:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function
with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(final_array)
```

Out[42]:

```
<Axes: ylabel='Density'>
```



EXPERIMENT:4 OUTLIERS

Result:

Outliers were successfully detected using both statistical and graphical methods.
They appeared as extreme values far from the main data range in the plots.
Identifying and handling outliers improved the dataset’s quality and model performance.