Aim:

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

Procedure:

1. Import necessary libraries: numpy and seaborn.
2. Generate a random integer array of 16 elements using np.random.randint(1,100,16).
3. Compute statistical measures:
   Mean using array.mean()
   Quartiles using np.percentile(array, 25), np.percentile(array, 50), np.percentile(array, 75)
4. Define a function outDetection(array):
   Calculate Q1 and Q3.
   Compute IQR = Q3 – Q1.
   Determine lower and upper range:
   LR = Q1 – 1.5 * IQR
   UR = Q3 + 1.5 * IQR
5. Filter the array to include only values within the range (array > LR and array < UR).
6. Plot the data distribution before and after outlier removal using sns.displot() or sns.distplot().

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

Out[1]:
```
array([15, 49, 71, 43,  7, 60, 13, 59,  6, 20,  9, 92, 60, 94, 59, 39],
      dtype=int32)
```

In [2]:
```
array.mean()
```

Out[2]:
```
np.float64(43.5)
```

In [3]:
```
np.percentile(array,25)
```

Out[3]:
```
np.float64(14.5)
```

In [4]:
```
np.percentile(array,50)
```

Out[4]:
```
np.float64(46.0)
```

In [5]:
```
np.percentile(array,75)
```

Out[5]:
```
np.float64(60.0)
```

In [6]:
```
np.percentile(array,100)
```

Out[6]:
```
np.float64(94.0)
```

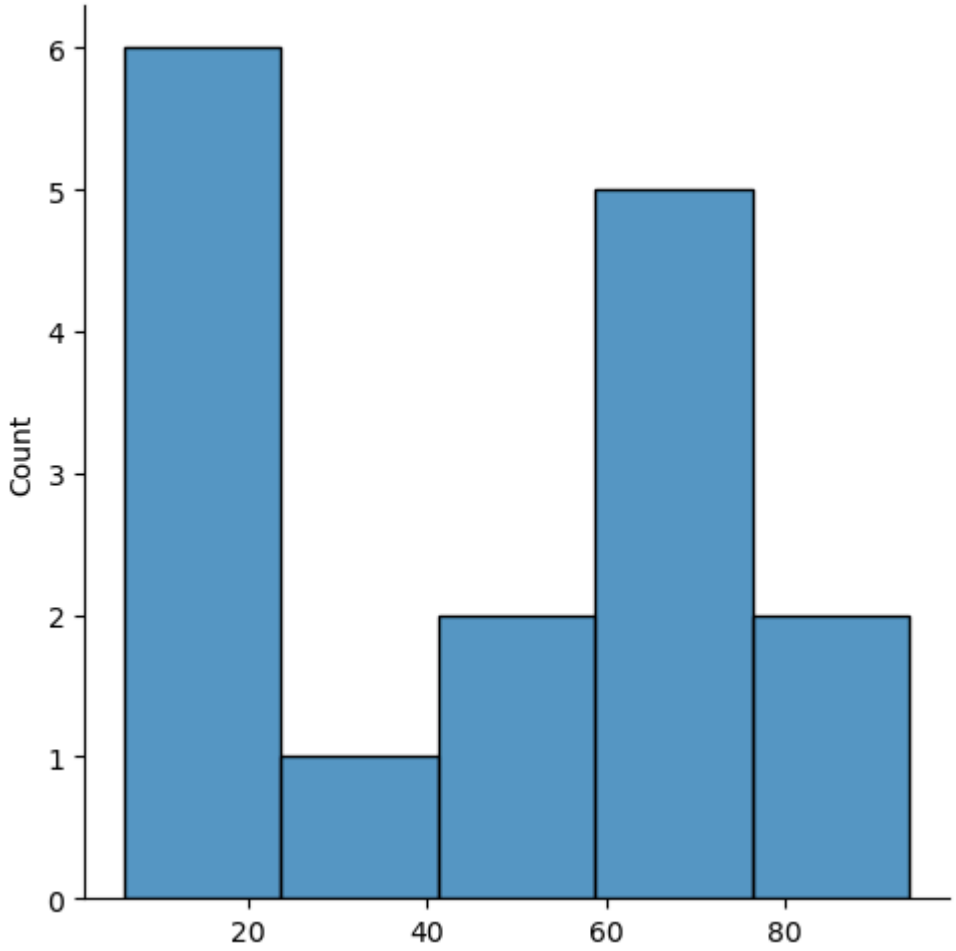
In [7]:
```
def outDetection(array):
  sorted(array)
  Q1,Q3=np.percentile(array,[25,75])
  IQR=Q3-Q1
  lr=Q1-(1.5*IQR)
  ur=Q3+(1.5*IQR)
  return lr,ur
```

In [8]:
```
lr,ur=outDetection(array)
lr,ur
```

Out[8]:
```
(np.float64(-53.75), np.float64(128.25))
```

In [9]:
```
import seaborn as sns
%matplotlib inline
sns.displot(array)
```

Out[9]:
```
<seaborn.axisgrid.FacetGrid at 0x188aea49be0>
```

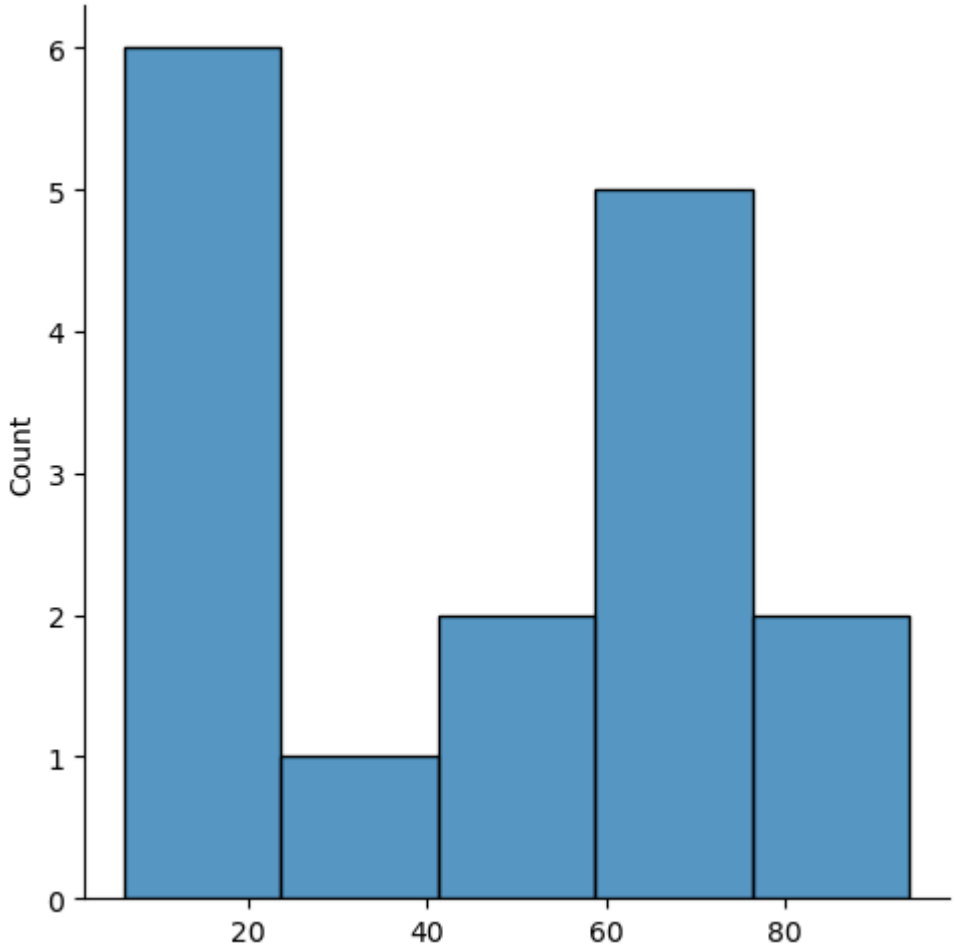


In [10]:
```
new_array=array[(array>lr) & (array<ur)]
new_array
```

Out[10]:
```
array([15, 49, 71, 43,  7, 60, 13, 59,  6, 20,  9, 92, 60, 94, 59, 39],
      dtype=int32)
```

In [11]:
```
sns.displot(new_array)
```

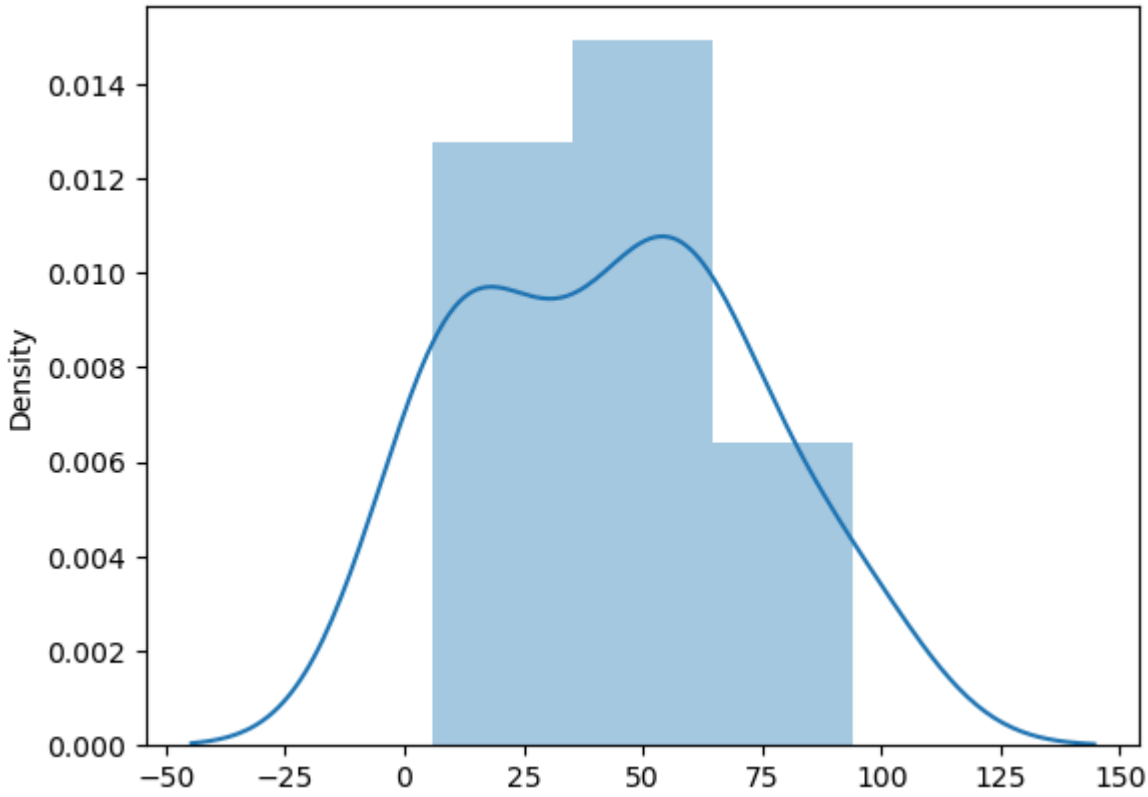
Out[11]:
```
<seaborn.axisgrid.FacetGrid at 0x188aeb90410>
```



In [12]:
```
sns.distplot(new_array)
```

```
C:\Users\Asus\AppData\Local\Temp\ipykernel_28432\4249554950.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(new_array)
```

Out[12]:
```
<Axes: ylabel='Density'>
```



In [13]:
```
lr1,ur1=outDetection(new_array)
lr1,ur1
```

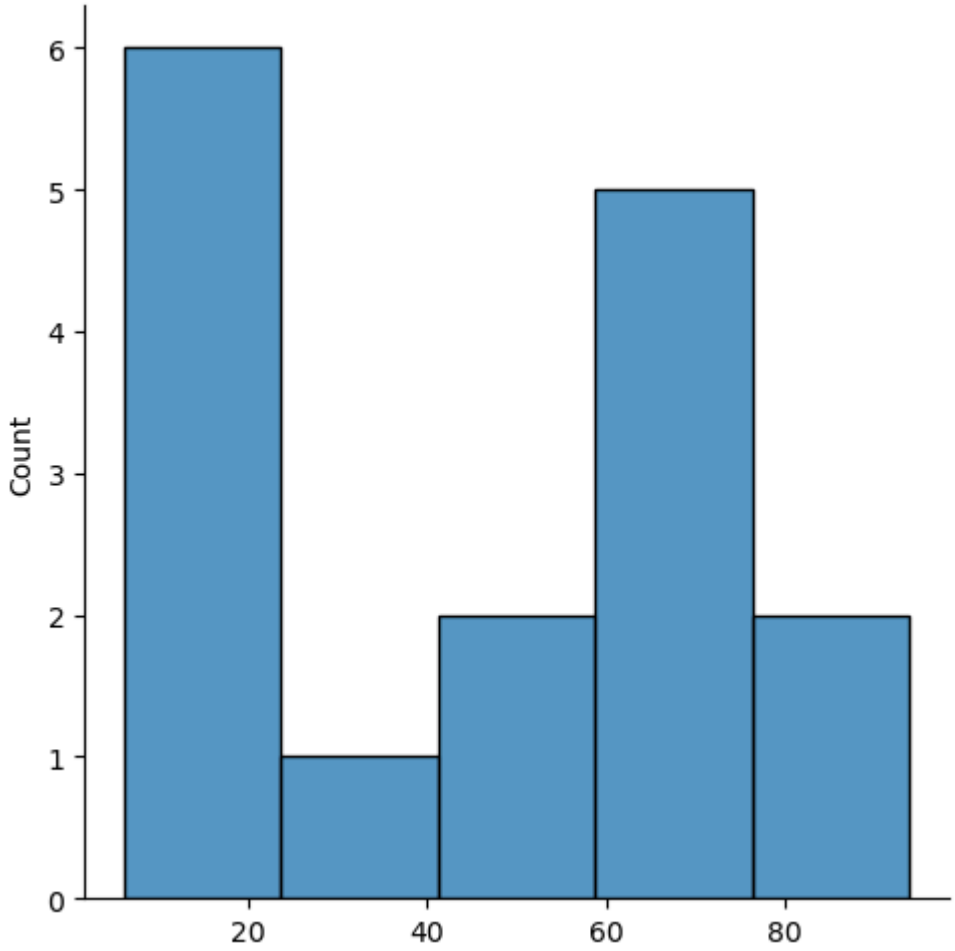
Out[13]:
```
(np.float64(-53.75), np.float64(128.25))
```

In [14]:
```
final_array=new_array[(new_array>lr1) & (new_array<ur1)]
final_array
```

Out[14]:
```
array([15, 49, 71, 43,  7, 60, 13, 59,  6, 20,  9, 92, 60, 94, 59, 39],
      dtype=int32)
```

In [15]:
```
sns.displot(final_array)
```

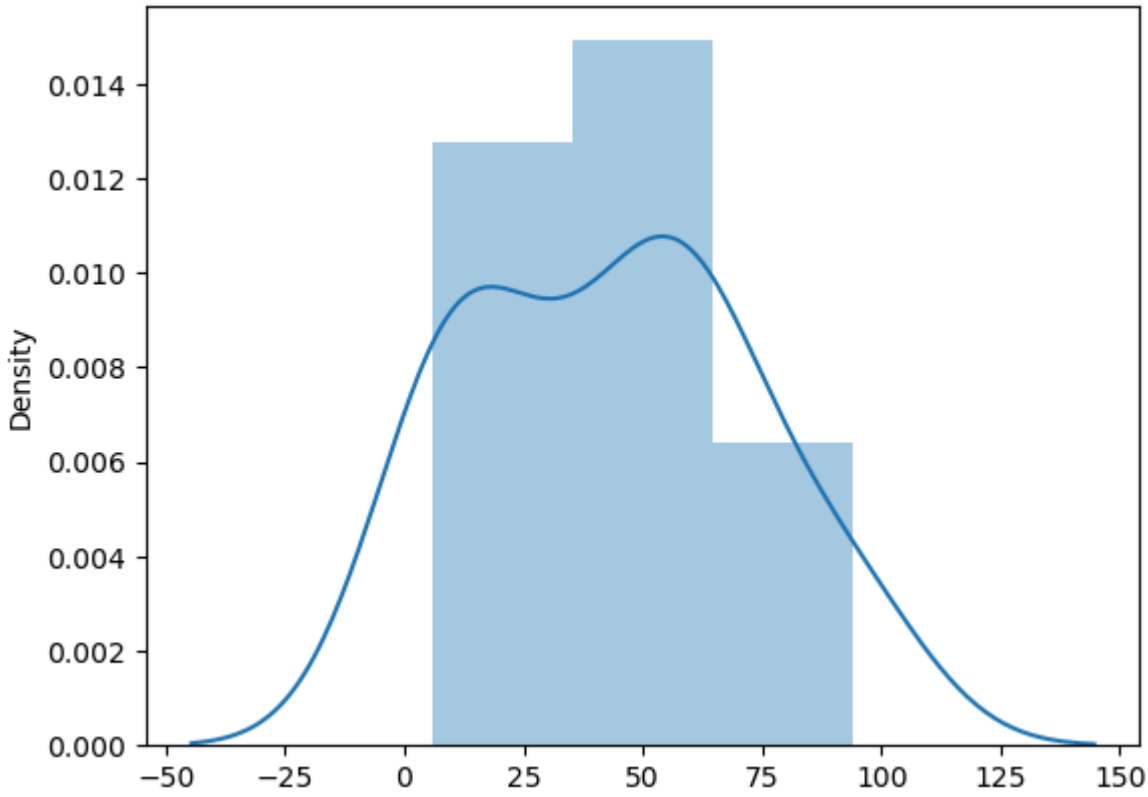
Out[15]:
```
<seaborn.axisgrid.FacetGrid at 0x188afc065d0>
```



In [16]:
```
sns.distplot(final_array)
```

```
C:\Users\Asus\AppData\Local\Temp\ipykernel_28432\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[16]:
```
<Axes: ylabel='Density'>
```


Density
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
−50
−25
0
25
50
75
100
125
150

Result:

All data points are within the normal range. The dataset contains no significant outliers.