In [ ]:
```
#Name: Neha Kamalakar Nemade
#Roll no: 22150
#Batch: G-5
```

Generating random number, array using randint() and rand() methods

In [44]:
```
#Creating random number using randint
from numpy import random
x=random.randint(500)
print(x)
```

```
148
```

In [35]:
```
print()
#generate random array
arr=random.randint(100,size=(5))
print(arr)
```

```
[10 72 66  7 87]
```

In [38]:
```
print()
#generate 2-d array
arr=random.randint(600,size=(4,6))
print(arr)
```

```
[[ 99 256 109 450 428 301]
 [473 150 444 513 518 597]
 [220 294  13 145 362 205]
 [595  96 335 588  29  62]]
```

In [45]:
```
# creating random float between 0 to 1
x=random.rand()
print(x)
```

```
0.652591262029358
```

In [33]:
```
#generating array of type float
arr=random.rand(5)
print(arr)
```

```
[0.01093999 0.13430828 0.8908858  0.35170308 0.8811336 ]
```

In [41]:
```
#generating 2-D array of type float
print()
arr=random.rand(4,5)
print(arr)
```

```
[[0.36185347 0.10583162 0.45202395 0.53218523 0.11415528]
 [0.82240462 0.36780057 0.09915455 0.91560782 0.14716615]
 [0.41277176 0.6565353  0.10559598 0.05025933 0.072034  ]
 [0.15663312 0.50936741 0.48728733 0.88358752 0.21611921]]
```

In [43]:
```
#generating random number from array using choice() method
arr=random.choice([5,6,7,8])
print(arr)
```

```
6
```

In [50]:
```
#generating 2-D array from given set of values
arr=random.choice([3,4,5,6,10],size=(4,5))
print(arr)
```

```
[[ 3  4  5  5  5]
 [ 6  5  4  3  5]
 [ 6  3 10  6  5]
 [ 4  5  5  6 10]]
```

Random data distribution

In [57]:
```
#generating 1-D array from given values and using given p obability
# The probability for the value to be 3 is set to be 0.1
# The probability for the value to be 5 is set to be 0.4
# The probability for the value to be 7 is set to be 0.3
# The probability for the value to be 9 is set to be 0.2

x=random.choice([3,5,7,9], p=[0.1,0.4,0.3,0.2], size=(100))
print("\n",x)
```

```
 [7 5 7 3 7 3 5 9 5 7 5 5 5 3 9 7 3 5 5 7 5 5 3 3 5 7 9 9 9 5 7 5 7 7 7 7 7
 5 5 3 3 7 7 7 7 7 7 7 9 7 5 5 5 5 5 5 9 5 9 7 5 9 5 5 5 7 7 5 9 5 7 5 7 5
 5 9 5 5 5 9 7 7 9 7 7 7 7 7 7 9 5 5 9 5 5 7 9 7 5 7]
```

In [63]:
```
#generating 2-D array with given probability
arr=random.choice([6,7,23,67], p=[0.2,0.4,0.1,0.3], size=(4,4))
print("\n",arr)
```

```
 [[67 67 67  6]
 [ 7  7 67  7]
 [67 67  7  7]
 [ 6  6 67 23]]
```

Random Permutations

In [78]:
```
#Shuffling array
import numpy as np
arr=np.array([42,68,30,24,80])
random.shuffle(arr)
print(arr)
```

```
[42 30 80 24 68]
```

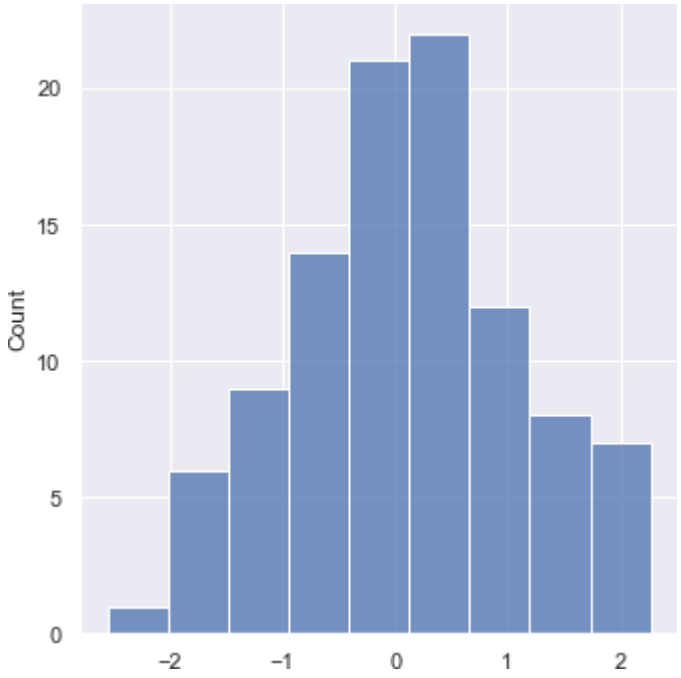
In [81]:
```
#generating random permutations of elements of array
arr=np.array([4,5,6,7,8])
print(random.permutation(arr))
```

```
[8 4 7 6 5]
```

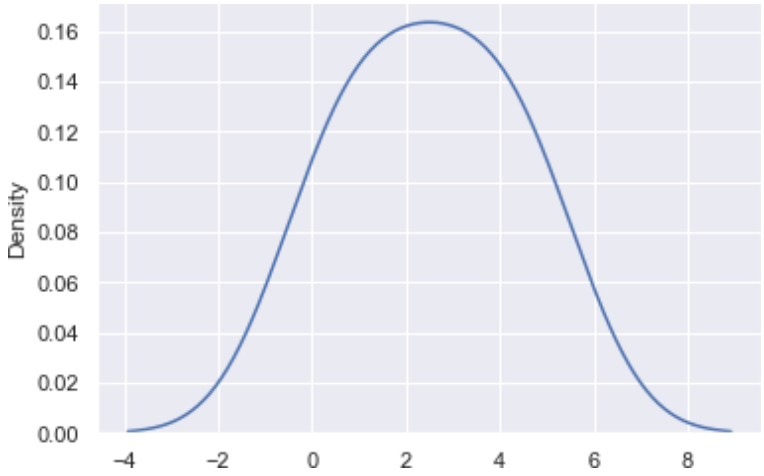
Seaborn Module

In [97]:
```
#plotting a Distplot
import matplotlib.pyplot as plt
import seaborn as sns

sns.set_theme()
np.random.seed(0)
x = np.random.randn(100)
ax = sns.displot(x)
```



In [100]:
```
sns.distplot([0,1,2,3,4,5], hist=False)
plt.show()
```



Normal Gaussian Distribution

In [106]:
```
#generating random normal distribution
x=random.normal(size=(4,5))
print(x)
```

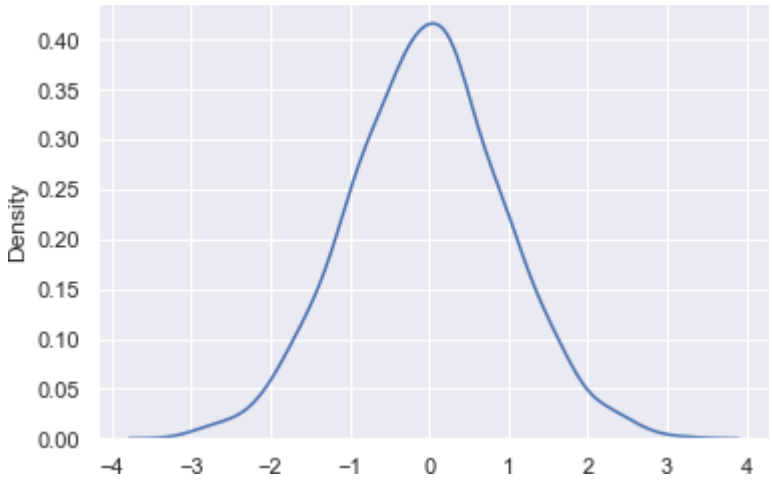
```
[[-0.36918184 -0.23937918  1.0996596   0.65526373  0.64013153]
 [-1.61695604 -0.02432612 -0.73803091  0.2799246  -0.09815039]
 [ 0.91017891  0.31721822  0.78632796 -0.4664191  -0.94444626]
 [-0.41004969 -0.01702041  0.37915174  2.25930895 -0.04225715]]
```

In [110]:
```
#generating random normal distribution with mean and standard deviation
# mean=1, standard deviation=2
arr=random.normal(loc=1, scale=2, size=(3,4))
print(arr)
```

```
[[-0.08572295  1.83210009 -1.31236486  2.5623962 ]
 [ 3.98896909 -3.13997005  1.85251746  2.35381607]
 [-0.27487405  0.20545637  0.73423884  0.40441824]]
```

Visualization of normal distribution

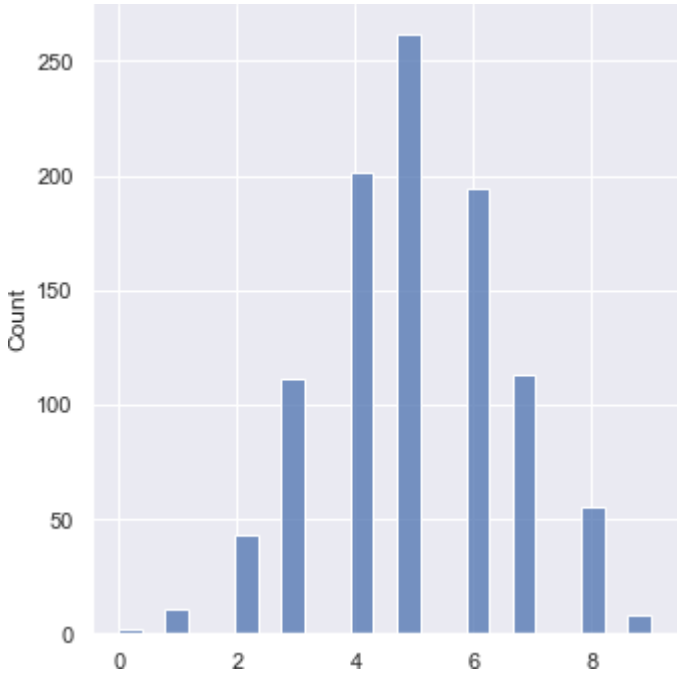
In [111]:
```
sns.distplot(random.normal(size=1000),hist=False)
plt.show()
```



In [114]:
```
arr=random.binomial(n=10,p=0.5,size=10)
print(arr)
```

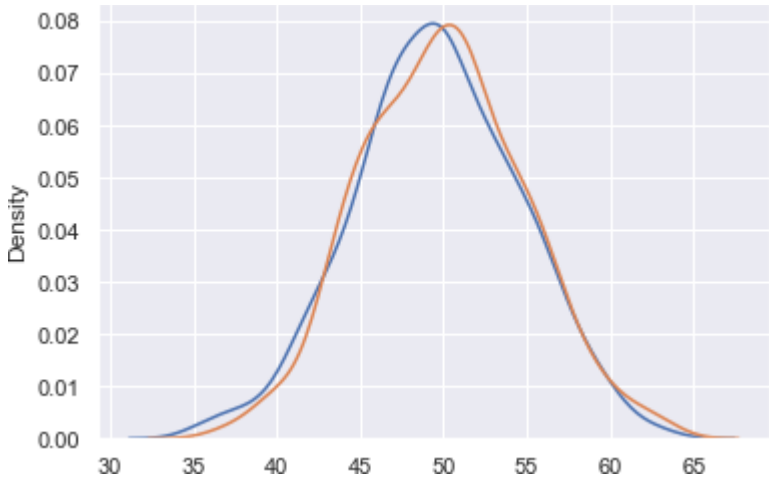
```
[6 3 6 6 4 5 5 5 6 8]
```

In [117]:
```
#vitualization of binomial distribution
sns.displot(random.binomial(n=10,p=0.5,size=1000))
plt.show()
```



Difference Between Normal and Binomial Distribution

In [122]:
```
sns.distplot(random.normal(loc=50,scale=5,size=1000), hist=False)
sns.distplot(random.binomial(n=100,p=0.5,size=1000),hist=False)
plt.show()
```



Uniform Distribution

In [125]:
```
x=random.uniform(size=(4,5))
print("\n",x)
```

```
[[0.69733635 0.59550783 0.81226642 0.05097729 0.12092097]
[0.45202225 0.17937865 0.50205434 0.32294824 0.7195445 ]
[0.78753686 0.69291673 0.24804844 0.30187789 0.0446775 ]
[0.81477633 0.505237   0.29917803 0.67580571 0.95910518]]
```

In [129]:
```
sns.distplot(random.uniform(size=1000),hist=False)
plt.show()
```