Random Numbers in NumPy

Random no. does not mean a diff. no. eve time Random means something that canno be predicated logically.

Pseudo Random and True Random.

Generate Random Number

e.g.1) Generate a random integer from 0 to 100

```
from numpy import random
x = random.randint(100)
print(x)
```

Output

```
95 (Each time diff. no. print)
```

e.g 2) Generate a random float from 0 to 1.

```
from numpy import random
x= random.rand( )
print (x)
```

O/P

```
0.29429030532180533
```

Generate Random Array

In NumPy we work with arrays, and you can use the two methods from the above examp to make random arrays.

Integer

The 'randint()' method takes a 'size'.

e.g.) Generate a 1-D array containing 5 random integers from 0 to 100

```
from ~~[illegible]~~ numpy import random
x = random.randint(100, size=(5))
print (x)
```

O/P

```
[3 91 45 10 92]
```

Random Numbers in NumPy

Random no does not mean a diff. no. every time Random means something that cannot be predicated logically.

Pseudo Random and True Random.

Generate Random Number

e.g 1) Generate a random integer from 0 to 100.

```
from numpy import random
x = random.randint(100)
print(x)
```

Output

```
95
```

(Each time diff. no. print)

e.g 2) Generate a random float from 0 to 1.

```
from numpy import random
x = random.rand( )
print(x)
```

O/P

```
0.29429030532180533
```

Generate Random Array

In NumPy we work with arrays, and you can use the two methods from the above example to make random arrays.

Integer

The 'randint()' method takes a 'size'.

e.g.) Generate a 1-D array containing 5 random integers from 0 to 100

```
from ~~nay~~ numpy import random
x = random.randint(100, size=(5))
print(x)
```

O/P

```
[3 91 45 10 92]
```

eg2) Generate a 2-D array with 3 rows, each row containing 5 random integers from 0 to 100.

```
from numpy import random
x = random.randint (100, size =(3,5))
print (x)
```

O/P

```
[[90 99  11  30 34]
 [66 40  63 36  37]
 [63  35  89  51  58]]
```

Floats

The rand( ) method also allow you to specify the shape of the array.

e.g1) Generate a 1-D array containing is random floats.

```
from numpy import random
x = random.rand(5)
print (x)
```

O/P

```
[0.9439043   0.0615613   0.8464911   0.8322703
                                     0.2007617]
```

e.g. 2) Generate a 2-D array with 3 rows. each row containing 5 random numbers.

```
from numpy import random
x = random.rand (3,5)
print (x)
```

O/P [[ ]]

Generate Random Number From Array

choice() method allows you to generate a random value based on an array of values.
choice() method takes an array as a parameter and randomly returns one of the values.

e.g-1) Return one of the values in an array:

```
from numpy import random
x = random.choice([3,5,7,9])
print(x)
```

O/P

```
5
```

⊛ The choice() method also allows you to return an array of values.
And a ~~size()~~ size parameter to specify the shape of the array.

e.g-2) Generate a 2-D array that consists of the values in the array parameter (3,5,7 and 9):

```
from numpy import random
x = random.choice([3,5,7,9],size=(3,5))
print(x)
```

O/P

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

Random Data Distribution

Random Distribution

e.g) Generate a 1-D array containing 100 values. where each values has to be 3,5,7 or 9.

The probability for the value to be 3 is set to be 0.1
" " " " " " " 5 " " 0.3
— — — — 7 " " 0.6
— — — — 9 " " 0

```
from numpy import random
x = random.choice([3,5,7,9], p=[0.1, 0.3, 0.6,
                               0.0], size=(100))

print(x)
```

O/P

Same example as above, but return a 2-D array with 3 rows, each containing 5 values.

```
from numpy import random
x = random.choice([3,5,7,9], p=[0.1, 0.3, 0.6, 0.0]
        size = (3,5))
print(x)
```

O/P

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

Random Permutations

Random Permutation of Elements.

Two method shuffle and permutation()

⊙ Shuffling Arrays:- Shuffle means changing arrangement of element in place i.e. in the array ~~to~~ itself.

e.g Randomly shuffle elements of following array:

```
from numpy import random
import numpy as np
arr = np.array([1,2,3,4,5])
random.shuffle(arr)
print(arr)
```

O/P

```
[2 5 4 3 1]
```

Generating Permutation of Arrays

```
from numpy import random
import numpy as np
arr = np.array([1,2,3,4,5])
print(random.permutation(arr))
```

O/P

```
[4 5 3 2 1]
```