



Random Data Generation with NumPy

NumPy random module allows you to generate random numbers and data arrays. Here are some fundamental functions:

1. `numpy.random.rand()`
2. `numpy.random.randn()`
3. `numpy.random.randint()`
4. `numpy.random.uniform()`
5. `numpy.random.normal()`

1. **numpy.random.rand:** Generates random numbers from a uniform distribution over $[0, 1)$.

Example Code:

```
import numpy as np

# Generate a 1D array of 5 random numbers
random_numbers = np.random.rand(3)
print(random_numbers)
```

Output

```
[0.272784  0.09733427 0.457863  ]
```

The code generates 3 random numbers.

2. **numpy.random.randn:** Generates random numbers from a standard normal distribution (mean=0, variance=1).

Example Code:

```
# Generate a 1D array of 5 random numbers from a normal
distribution
random_numbers_normal = np.random.randn(3)
print(random_numbers_normal)
```

Output

```
[ 0.04495402 -1.09919444  1.82620763]
```

This Python code generate a 1D array (`random_numbers_normal`) containing 3 random numbers sampled from a standard normal



distribution (mean = 0, standard deviation = 1). The print statement displays the generated array.

3. **numpy.random.randint**: Generates random integers within a specified range.

Example Code:

```
# Generate a 1D array of 5 random integers between 1
and 10
random_integers = np.random.randint(1, 11, 5)
print(random_integers)
```

Output

```
[ 4 10 10  2  6]
```

The code generates a 1-D array (random_integers) containing 5 random integers between 1 and 10 (inclusive). The np.random.randint(1, 11, 5) function is used for this purpose, where 1 is the lower bound (inclusive), 11 is the upper bound (exclusive), and 5 specifies the number of random integers to generate. The print statement displays the resulting array, showing an example of 5 random integers between 1 and 10.

4. **numpy.random.uniform**: Generates random numbers from a uniform distribution over a specified range.

Example Code:

```
import numpy as np
# Generate a random number between 2 and 5
random_number = np.random.uniform(2, 5)
print(random_number)
```

Output

```
4.833586883593855
```

The code generates a random floating-point number between 2 (inclusive) and 5 (exclusive). The **np.random.uniform(2, 5)** function accomplishes this.



5. **numpy.random.normal**: Generates random numbers from a normal (Gaussian) distribution with a specified mean and standard deviation.

```
import numpy as np

# Generate a random number from a normal distribution
with mean=0 and std=1
random_number = np.random.normal(0, 1)
print(random_number)
```

Output
1.481601952386309

The code generates a random number from a standard normal distribution (mean=0 and standard deviation=1) using NumPy's **random.normal** function and then prints the generated random number.

Applications of Random Data:

1. **Simulations and Modelling**: Random data generation is widely used in simulations and modelling. For example, simulating the behaviour of a system under uncertain conditions.
2. **Monte Carlo Simulations**: In finance and engineering, Monte Carlo simulations involve using random numbers to model the probability of different outcomes in a process.
3. **Data Augmentation**: In machine learning, random data generation is used for data augmentation to increase the diversity of the training dataset.
4. **Statistical Testing**: Random data is essential for statistical testing and hypothesis checking.
5. **Encryption and Security**: Random data is often used in **cryptographic applications** for key generation and other security-related tasks.