# Random Sampling

## NumPy Random Module

The `numpy.random` module provides us with random number generators (RNG). You can find the documentation here. As there name suggests, random number generators produce random numbers. In this section we highlight a few essential functions from the module:

### `np.random.random()`

This function produces random floating point numbers from a uniform probability distribution function (PDF) on the interval $[0, 1)$ (1 is excluded). If no arguments are provided a single number is generated:

```
[2]: np.random.random()
```

```
[2]: 0.12055240311517734
```

If the length or shape is specified, `random()` returns an array of random numbers:

```
[4]: np.random.random(5)
```

```
[4]: array([0.08250099, 0.6587371 , 0.53175303, 0.67709712, 0.1558734 ])
```

```
[5]: np.random.random((2, 3))
```

```
[5]: array([[0.59302342, 0.84327141, 0.05504497],
            [0.98913576, 0.63069964, 0.73478334]])
```

If you want to produce uniformly distributed random numbers $R$ on the interval $[a, b)$, you can use random numbers $r$ from the interval $[0, 1)$ by scaling and shifting them:

$$R = a + r * (b - a)$$

For example, to generate uniform random numbers on the interval $[18, 30)$:

```
[6]: np.random.random(4)*(30 -18) + 18
```

```
[6]: array([28.90915681, 26.98028702, 29.02897959, 20.18055287])
```

To read more about `numpy.random.random()`, see the documentation.

### `np.random.randint()`

This function produces random integers sampled from a uniform probability distribution on a **specified** interval.

The interval is defined by the first 2 arguments of `randint()`, the end of the interval (second number) is not included in the interval:

```
[10]: #Random numbers from 1 up to 10
      np.random.randint(1, 10)
```

```
[10]: 9
```

Again, you can specify a size or shape of the output array:

```
[11]: np.random.randint(1, 10, 3)
```

```
[11]: array([7, 9, 7])
```

```
 [9]: np.random.randint(1, 10, (2, 4))
```

```
 [9]: array([[5, 8, 7, 1],
             [9, 4, 9, 7]])
```

### Random Numbers From Other Distributions

`numpy.random` provides us with many more RNG functions that sample from many of the most popular PDFs. You can see the full list in the documentation.

For example, the `np.random.norm()` function produces random numbers sampled from the normal (Gaussian) distribution. Parameters like the mean and standard deviation (or first 2 moments) can be specified.

All of these functions can generate array outputs