

**The NumPy. Random package**

jupyter notebook explaining the concepts behind and the use of the numpy random package,

including plots of the various distributions.

1. Rand

|  |  |
| --- | --- |
| **Parameters:** | **d0, d1, …, dn** : int, optional  The dimensions of the returned array should all be positive. If no argument is given a single Python float is returned. |
| **Returns:** | **output**: ndarray, shape (d0, d1, ..., dn)  Random values. |

Create an array of the given shape and populate it with random samples from a uniform distribution over `` [0, 1) ``.

Ref: <https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.random.rand.html#numpy.random.rand>

This function takes in either an integer value or empty value, if you only put in one value it returns one random number as a float value. If you pass an argument integer value to this function such as 100 it will return an array (numpy.ndarray) of 100 float values at random in the range 0 to 1 excluding 1.

1. Uniform

|  |  |
| --- | --- |
| **Parameters:** | **low** : float or arraylike of floats, optional  Lower boundary of the output interval. All values generated will be greater than or equal to low. The default value is 0.  **high** : float or arraylike of floats  Upper boundary of the output interval. All values generated will be less than high. The default value is 1.0.  **size** : int or tuple of ints, optional  Output shape. If the given shape is, e.g., (m, n, k), then m \* n \* k samples are drawn. If size is None (default), a single value is returned if low and high are both scalars. Otherwise, np.broadcast(low, high).size samples are drawn. |
| **Returns:** | **out** : ndarray or scalar  Drawn samples from the parameterized uniform distribution. |

Ref: <https://docs.scipy.org/doc/numpy-1.15.1/reference/generated/numpy.random.uniform.html#numpy.random.uniform> , <https://www.statisticshowto.datasciencecentral.com/uniform-distribution/>

The second function I will be researching here is the uniform function. This function allows for more input parameters such as the lower values, the higher values and the size. When the user inputs values for the low and high, if the higher value is lower than the low value the function will switch these values and still run, so it is not a good idea to rely on this function if you are relying on the inequality condition.

A **uniform distribution**, also called a **rectangular distribution**, is a probability distribution that has **constant probability**.

1. Normal

|  |  |
| --- | --- |
| **Parameters:** | **loc** : float or array\_like of floats  Mean (“centre”) of the distribution.  **scale** : float or array\_like of floats  Standard deviation (spread or “width”) of the distribution.  **size** : int or tuple of ints, optional  Output shape. If the given shape is, e.g., (m, n, k), then m \* n \* k samples are drawn. If size is None (default), a single value is returned if loc and scale are both scalars. Otherwise, np.broadcast(loc, scale).size samples are drawn. |
| **Returns:** | **out** : ndarray or scalar  Drawn samples from the parameterized normal distribution. |

The normal function found in NumPy states Draw random samples from a normal (Gaussian) distribution. this function takes in 3 inputs: loc (mean), scale (Standard Deviation) and the size. Returning a scaler graphical representation of what is commonly referred to as the bell curve.

ref: <https://en.wikipedia.org/wiki/Normal_distribution>

1. Randint
2. Power
3. Poisson