

**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

|  |  |
| --- | --- |
| **Parameters:** | **low** : int  Lowest (signed) integer to be drawn from the distribution (unless high=None, in which case this parameter is one above the *highest* such integer).  **high** : int, optional  If provided, one above the largest (signed) integer to be drawn from the distribution (see above for behavior if high=None).  **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. Default is None, in which case a single value is returned.  **dtype** : dtype, optional  Desired dtype of the result. All dtypes are determined by their name, i.e., ‘int64’, ‘int’, etc, so byteorder is not available and a specific precision may have different C types depending on the platform. The default value is ‘np.int’.  New in version 1.11.0. |
| **Returns:** | **out** : int or ndarray of ints  *size*-shaped array of random integers from the appropriate distribution, or a single such random int if *size* not provided. |
|  |  |

ref: <https://docs.scipy.org/doc/numpy-1.15.1/reference/generated/numpy.random.randint.html#numpy.random.randint>

randint is very similar to rand however, it only returns value in integer format. This function can take up to 4 inputs, the low, high, size and dtype. for the dtype it is optional and can specify what the data type of the integers you want back, int64 or int etc.

1. Power

|  |
| --- |
| **a** : float or array\_like of floats  Parameter of the distribution. Should be greater than zero.  **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 a is a scalar. Otherwise, np.array(a).size samples are drawn. |
| **Returns:** | **out** : ndarray or scalar  Drawn samples from the parameterized power distribution. |
| **Raises:** | **ValueError**  If a < 1. |

ref: <https://docs.scipy.org/doc/numpy-1.15.1/reference/generated/numpy.random.power.html#numpy.random.power>

The power distribution of the random dependency takes 2 parameters, a being the size of the distribution which is a float value, and the samples value. The greater the number of samples the smoother the curve will be.

adapted from:

<https://docs.scipy.org/doc/numpy-1.15.0/reference/generated/numpy.random.power.html#numpy.random.power>

<https://en.wikipedia.org/wiki/Pareto_principle>

<https://en.wikipedia.org/wiki/Power_law>

Including some examples of the power function are <https://en.wikipedia.org/wiki/Initial_mass_function>

1. Poisson

|  |  |
| --- | --- |
| **Parameters:** | **lam** : float or array\_like of floats  Expectation of interval, should be >= 0. A sequence of expectation intervals must be broadcastable over the requested size.  **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 lam is a scalar. Otherwise, np.array(lam).size samples are drawn. |
| **Returns:** | **out** : ndarray or scalar  Drawn samples from the parameterized Poisson distribution. |

The Poisson distribution may be useful to model events such as:

The number of meteorites greater than 1-meter diameter that strike Earth in a year

The number of patients arriving in an emergency room between 10 and 11 pm

The number of photons hitting a detector in a time interval

src: https://en.wikipedia.org/wiki/Poisson\_distribution

The poisson function in NumPy takes 2 values:

lam = Expected Interval value (float)

size = The shape that you want to be displayed.