

How to Generate Normally Distributed Random Samples in Python Using `numpy.random.normal()`

Authored by
stats writer

November 24, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Generate Normally Distributed Random Samples in Python Using `numpy.random.normal()`*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=100384>

In Python, the equivalent of the R function `rnorm()` is the `numpy.random.normal()` function. This function takes in three parameters, the mean, size and standard deviation, and generates a normally distributed random sample that follows the given parameters. This sample can then be used for statistical analyses or visualizations.

In the R programming language, we can use the `rnorm()` function to generate a vector of random values that follow a with a specific mean and standard deviation.

For example, the following code shows how to use `rnorm()` to create a vector of 8 random values that follow a normal distribution with a mean of 5 and standard deviation of 2:

```
#make this example reproducible
```

```
set.seed(1)
```

```
#generate vector of 8 values that follow normal distribution with mean=5 and sd=2
```

```
rnorm(n=8, mean=5, sd=2)
```

```
3.747092 5.367287 3.328743 8.190562 5.659016 3.359063 5.974858 6.476649
```

The equivalent of the `rnorm()` function in Python is the `np.random.normal()` function, which uses the following basic syntax:

```
np.random.normal(loc=0, scale=1, size=None)
```

where:

loc: Mean of the distribution

scale: Standard deviation of the distribution

size: Sample size

The following example shows how to use this function in practice.

Example: Using the Equivalent of `rnorm()` in Python

The following code shows how to use the `np.random.normal()` function to generate an array of random values that follow a normal distribution with a specific mean and standard deviation.

```
import numpy as np
```

```
#make this example reproducible
```

```
np.random.seed(1)
```

```
#generate array of 8 values that follow normal distribution with mean=5 and sd=2
np.random.normal(loc=5, scale=2, size=8)
```

```
array()
```

The result is a NumPy array that contains 8 values generated from a normal distribution with a mean of 5 and a standard deviation of 2.

You can also create a histogram using Matplotlib to visualize a normal distribution generated by the `np.random.normal()` function:

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
#make this example reproducible
```

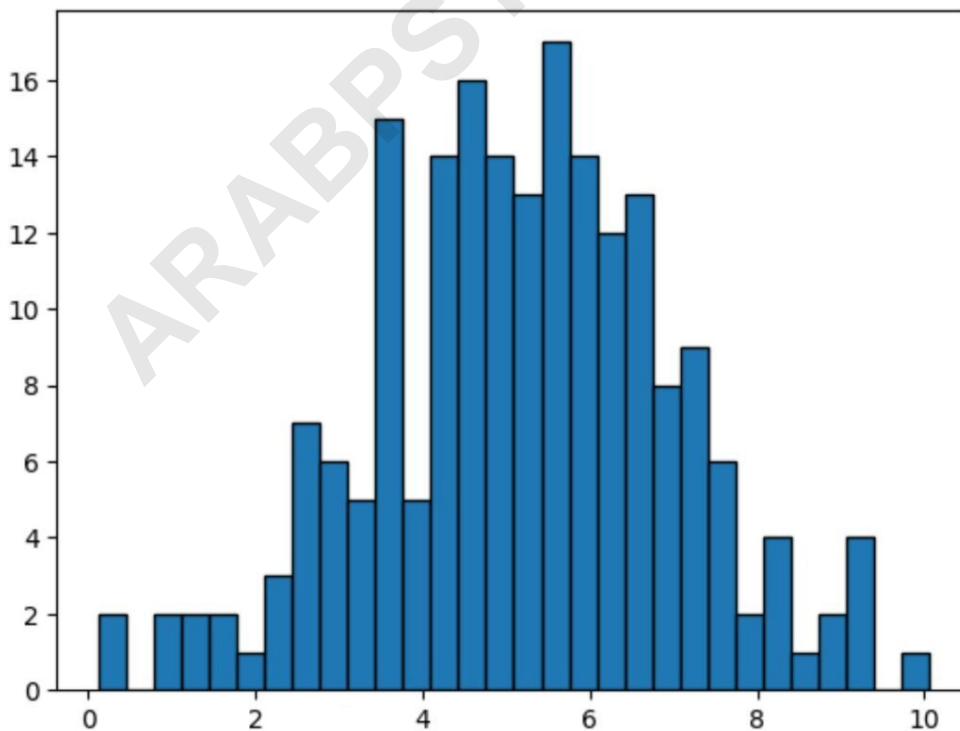
```
np.random.seed(1)
```

```
#generate array of 200 values that follow normal distribution with mean=5 and sd=2
```

```
data = np.random.normal(loc=5, scale=2, size=200)
```

```
#create histogram to visualize distribution of values
```

```
plt.hist(data, bins=30, edgecolor='black')
```



We can see that the distribution of values is roughly bell-shaped with a mean located at 5 and a standard deviation of 2.

Note: You can find the complete documentation for the `np.random.normal()` function .

The following tutorials explain how to perform other common operations in Python:

ARABPSYCHOLOGY.COM