

# How can a normal distribution be generated in Python, and what are some examples of its use?

Authored by  
**stats writer**

April 21, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can a normal distribution be generated in Python, and what are some examples of its use?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=137700>

A normal distribution, also known as a Gaussian distribution, is a commonly used statistical distribution that represents the probability of a continuous variable. In Python, a normal distribution can be generated using the "numpy.random.normal" function, which takes parameters such as mean, standard deviation, and size to generate a set of random numbers following a normal distribution.

Some examples of its use include simulating data for statistical analysis, generating random numbers for testing purposes, and fitting a normal distribution to a set of data to understand its underlying distribution. It is also used in various fields such as finance, biology, and physics for modeling and analyzing continuous variables. The ability to generate a normal distribution in Python makes it a powerful tool for data analysis and decision making.

## Generate a Normal Distribution in Python (With Examples)

You can quickly generate a normal distribution in Python by using the `numpy.random.normal()` function, which uses the following syntax:

```
numpy.random.normal(loc=0.0, scale=1.0, size=None)
```

**where:**

**loc:** Mean of the distribution. Default is 0.  
**scale:** Standard deviation of the distribution. Default is 1.  
**size:** Sample size.

**This tutorial shows an example of how to use this function to generate a normal distribution in Python.**

## How to Make a Bell Curve in Python

**Example: Generate a Normal Distribution in Python**

**The following code shows how to generate a normal distribution in Python:**

```
from numpy.random import seed
from numpy.random import normal

#make this example reproducible
seed(1)

#generate sample of 200 values that follow a normal
distribution
data = normal(loc=0, scale=1, size=200)

#view first six values
data

array()
```

**We can quickly find the mean and standard deviation of this distribution:**

```
import numpy as np
```

**#find mean of sample**

**np.mean(data)**

**0.1066888148479486**

**#find standard deviation of sample**

**np.std(data, ddof=1)**

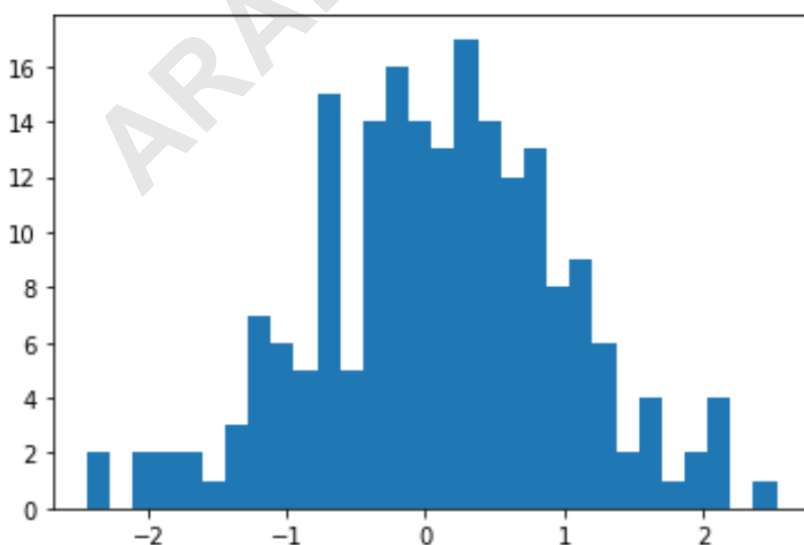
**0.9123296653173484**

**We can also create a quick histogram to visualize the distribution of data values:**

**import matplotlib.pyplot as plt**

**count, bins, ignored = plt.hist(data, 30)**

**plt.show()**



We can even perform a **Shapiro-Wilk test** to see if the dataset comes from a normal population:

```
from scipy.stats import shapiro
```

```
#perform Shapiro-Wilk test
```

```
shapiro(data)
```

```
ShapiroResult(statistic=0.9958659410,  
pvalue=0.8669294714)
```

The p-value of the test turns out to be 0.8669. Since this value is not less than .05, we can assume the sample data comes from a population that is normally distributed.