# How to Perform Bootstrapping in Python?

Authored by

**stats writer**

November 29, 2025

Bootstrapping is a statistical technique used to estimate the reliability of a particular sample by randomly selecting observations from a larger population with replacement. In Python, bootstrapping can be performed using the scikit-learn library, which provides functions for generating bootstrapped samples and computing summary statistics on them. Additionally, the statsmodels library can be used to fit various regression models to the bootstrapped samples.

**Bootstrapping** is a method that can be used to construct a confidence interval for a when the sample size is small and the underlying distribution is unknown.

The basic process for bootstrapping is as follows:

Take *k* repeated samples with replacement from a given dataset.
For each sample, calculate the statistic you're interested in.
This results in *k* different estimates for a given statistic, which you can then use to calculate a confidence interval for the statistic.

The easiest way to perform bootstrapping in Python is to use the function from the **SciPy** library.

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

## Example: Perform Bootstrapping in Python

Suppose we create a dataset in Python that contains 15 values:

**#define array of data values**
**data =**

We can use the following code to calculate a 95% bootstrapped confidence interval for the median value:

**from scipy.stats import bootstrap**
**import numpy as np**

#convert array to sequence
data = (data,)

#calculate 95% bootstrapped confidence interval for median
bootstrap_ci = bootstrap(data, np.median, confidence_level=0.95,
random_state=1, method='percentile')

#view 95% boostrapped confidence interval

print(bootstrap_ci.confidence_interval)

ConfidenceInterval(low=10.0, high=20.0)

The 95% bootstrapped confidence interval for the median turns out to be .

Here's what the **boostrap()** function actually did under the hood:

The **bootstrap()** function generated 9,999 samples with replacement. (The default is 9,999 but you can use the **n_resamples** argument to change this number)
For each bootstrapped sample, the median was calculated.
The median value of each sample was arranged from smallest to largest and the median value at percentile 2.5% and percentile 97.5% were used to construct the lower and upper limits of the 95% confidence interval.

Note that you can calculate a bootstrapped confidence interval for virtually any statistic.

For example, we can change **np.median** to **np.std** within the **bootstrap()** function to instead calculate a 95% confidence interval for the standard deviation:

**from scipy.stats import bootstrap**
**import numpy as np**

#convert array to sequence
data = (data,)

#calculate 95% bootstrapped confidence interval for median
bootstrap_ci = bootstrap(data, np.std, confidence_level=0.95,
random_state=1, method='percentile')

#view 95% boostrapped confidence interval
print(bootstrap_ci.confidence_interval)

ConfidenceInterval(low=3.3199732261303283, high=5.66478399066117)

The 95% bootstrapped confidence interval for the standard deviation turns out to be .

The following tutorials explain how to perform bootstrapping in other statistical software: