

# How can the Standard Error of the Mean be calculated in Python?

Authored by  
**stats writer**

April 20, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can the Standard Error of the Mean be calculated in Python?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=137299>

The Standard Error of the Mean (SEM) is a statistical measure that represents the amount of variability in the sample mean from the population mean. In Python, the SEM can be calculated by using the "scipy" library and the "sem" function. This function takes in an array of values and returns the SEM value. To calculate the SEM, the data must be first imported into Python and stored as an array. Then, the "sem" function can be applied to the array to obtain the SEM value. This allows for a quick and accurate calculation of the SEM in Python, which can be used for further statistical analysis and interpretation of data.

## Calculate the Standard Error of the Mean in Python

The standard error of the mean is a way to measure how spread out values are in a dataset. It is calculated as:

$$\text{Standard error of the mean} = s / \sqrt{n}$$

where:

**s:** sample standard deviation  
**n:** sample size

This tutorial explains two methods you can use to calculate the standard error of the mean for a dataset in Python. Note that both methods produce the exact same results.

### Method 1: Use SciPy

The first way to calculate the standard error of the mean is to use the `sem()` function from the SciPy Stats library.

The following code shows how to use this function:

```
from scipy.stats import sem

#define dataset
data =

#calculate standard error of the mean
sem(data)

2.001447
```

The standard error of the mean turns out to be 2.001447.

Method 2: Use NumPy

Another way to calculate the standard error of the mean for a dataset is to use the std() function from NumPy.

*Note that we must specify  $ddof=1$  in the argument for this function to calculate the sample standard deviation as opposed to the population standard deviation.*

The following code shows how to do so:

```
import numpy as np
```

```
#define dataset  
data = np.array()  
  
#calculate standard error of the mean  
np.std(data, ddof=1) / np.sqrt(np.size(data))  
  
2.001447
```

Once again, the standard error of the mean turns out to be 2.001447.

How to Interpret the Standard Error of the Mean

1. The larger the standard error of the mean, the more spread out values are around the mean in a dataset.

To illustrate this, consider if we change the last value in the previous dataset to a much larger number:

```
from scipy.stats import sem  
  
#define dataset  
data =  
  
#calculate standard error of the mean  
sem(data)
```

**6.978265**

Notice how the standard error jumps from 2.001447 to 6.978265. This is an indication that the values in this dataset are more spread out around the mean compared to the previous dataset.

2. As the sample size increases, the standard error of the mean tends to decrease.

To illustrate this, consider the standard error of the mean for the following two datasets:

```
from scipy.stats import sem
```

```
#define first dataset and find SEM
```

```
data1 =
```

```
sem(data1)
```

**0.7071068**

```
#define second dataset and find SEM
```

```
data2 =
```

```
sem(data2)
```

**0.4714045**

**The second dataset is simply the first dataset repeated twice. Thus, the two datasets have the same mean but the second dataset has a larger sample size so it has a smaller standard error.**

**How to Calculate the Standard Error of the Mean in R**

**How to Calculate the Standard Error of the Mean in**

**Excel**

**How to Calculate Standard Error of the Mean in Google**

**Sheets**

ARABPSYCHOLOGY.COM