

How can I calculate the geometric mean in Python, and could you provide some examples?

Authored by
stats writer

May 3, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I calculate the geometric mean in Python, and could you provide some examples?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=142356>

The geometric mean is a type of average that is used to calculate the central tendency of a set of numbers. It is calculated by taking the n th root of the product of n numbers. In Python, the geometric mean can be calculated using the "scipy" library's "gmean" function. This function takes in a list of numbers as its argument and returns the geometric mean. For example, if we have a list of numbers, the geometric mean can be calculated as $\text{gmean}() = 4$. Therefore, the geometric mean can be used to find the average growth rate or to measure the average of ratios.

Calculate Geometric Mean in Python (With Examples)

There are two ways to calculate the geometric mean in Python:

Method 1: Calculate Geometric Mean Using SciPy

```
from scipy.stats import gmean
```

```
#calculate geometric mean  
gmean()
```

Method 2: Calculate Geometric Mean Using NumPy

```
import numpy as np
```

```
#define custom function  
def g_mean(x):  
    a = np.log(x)  
    return np.exp(a.mean())
```

```
#calculate geometric mean  
g_mean()
```

Both methods will return the exact same results.

The following examples show how to use each of these methods in practice.

Example 1: Calculate Geometric Mean Using SciPy

The following code shows how to use the `gmean()` function from the library to calculate the geometric mean of an array of values:

```
from scipy.stats import gmean
```

```
#calculate geometric mean  
gmean()
```

```
4.81788719702029
```

The geometric mean turns out to be 4.8179.

Example 2: Calculate Geometric Mean Using NumPy

The following code shows how to write a custom function to calculate a geometric mean using built-in

functions from the library:

```
import numpy as np
```

```
#define custom function
```

```
def g_mean(x):
```

```
    a = np.log(x)
```

```
    return np.exp(a.mean())
```

```
#calculate geometric mean
```

```
g_mean()
```

```
4.81788719702029
```

The geometric mean turns out to be 4.8179, which matches the result from the previous example.

How to Handle Zeros

Note that both methods will return a zero if there are any zeros in the array that you're working with.

```
#create array with some zeros
```

```
x =
```

```
#remove zeros from array
```

```
x_new =
```

```
#view updated array
```

```
print(x_new)
```

ARABPSYCHOLOGY.COM