

# How do you calculate the expected value in Python, and what are some examples?

Authored by  
**stats writer**

July 1, 2024

## RECOMMENDED CITATION

stats writer (2024). *How do you calculate the expected value in Python, and what are some examples?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=165165>

Expected value is a mathematical concept used to determine the average outcome of a random event. In Python, the expected value can be calculated by multiplying each possible outcome by its corresponding probability and summing all the values together. This can be achieved using the "sum()" function or by using the "for" loop to iterate through the outcomes and probabilities. Some examples of calculating the expected value in Python include determining the average earnings from a game of chance or predicting the average return on investment for a portfolio. By calculating the expected value, we can make informed decisions and manage risks in various scenarios.

## Calculate Expected Value in Python (With Examples)

**A probability distribution tells us the probability that a takes on certain values.**

**For example, the following probability distribution tells us the probability that a certain soccer team scores a certain number of goals in a given game:**

Goals (X)	Probability P(X)
0	0.18
1	0.34
2	0.35
3	0.11
4	0.02

**To find the expected value of a probability distribution, we can use the following formula:**

$$\mu = \sum x * P(x)$$

where:

**x: Data value P(x): Probability of value**

For example, the expected number of goals for the soccer team would be calculated as:

$\mu = 0*0.18 + 1*0.34 + 2*0.35 + 3*0.11 + 4*0.02 = 1.45$   
goals.

To calculate expected value of a probability distribution in Python, we can define a simple function:

```
import numpy as np
```

```
def expected_value(values, weights):
```

```
    values = np.asarray(values)
```

```
    weights = np.asarray(weights)
```

```
    return (values * weights).sum() / weights.sum()
```

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

Example: Calculating Expected Value in Python

The following code shows how to calculate the expected value of a probability distribution using the

**expected\_value()** function we defined earlier:

```
#define values
```

```
values =
```

```
#define probabilities
```

```
probs =
```

```
#calculate expected value
```

```
expected_value(values, probs)
```

```
1.450000
```

The expected value is 1.45. This matches the value that we calculated earlier by hand.

Note that this function will return an error if the length of the values array and the probabilities array are not equal.

```
#define values
```

```
values =
```

```
#define probabilities
```

```
probs =
```

```
#attempt to calculate expected value  
expected_value(values, probs)
```

**ValueError: operands could not be broadcast together with shapes (5,) (7,)**

**We receive an error because the length of the first array is 5 while the length of the second array is 7.**

**In order for this expected value function to work, the length of both arrays must be equal.**

**Additional Resources**

**The following tutorials explain how to calculate other metrics in Python:**