

# How can I normalize a NumPy matrix and what are some examples of this process?

Authored by  
**stats writer**

July 1, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I normalize a NumPy matrix and what are some examples of this process?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=165306>

Normalizing a NumPy matrix is the process of rescaling the values of the matrix so that they fall within a specific range. This is important for data analysis and machine learning as it eliminates the impact of varying scales on the results. The most commonly used method for normalization is the min-max scaling, where the values are transformed to fall between 0 and 1. This is achieved by subtracting the minimum value from each element and then dividing it by the difference between the maximum and minimum values. Other methods include Z-score, where the values are transformed to have a mean of 0 and standard deviation of 1, and decimal scaling, where the values are divided by a constant power of 10 to bring them within a specific range. Normalizing a NumPy matrix can improve the performance of algorithms and make the data more interpretable. Examples of this process include normalizing pixel values in image processing, normalizing stock prices in financial analysis, and normalizing gene expression levels in bioinformatics.

## Normalize a NumPy Matrix (With Examples)

**To normalize a matrix means to scale the values such that that the range of the row or column values is between 0 and 1.**

**The easiest way to normalize the values of a NumPy matrix is to use the function from the sklearn package, which uses the following basic syntax:**

```
from sklearn.preprocessing import normalize
```

```
#normalize rows of matrix
```

```
normalize(x, axis=1, norm='l1')
```

```
#normalize columns of matrix
```

```
normalize(x, axis=0, norm='l1')
```

The following examples show how to use this syntax in practice.

#### Example 1: Normalize Rows of NumPy Matrix

Suppose we have the following NumPy matrix:

```
import numpy as np
```

```
#create matrix
```

```
x = np.arange(0, 36, 4).reshape(3,3)
```

```
#view matrix
```

```
print(x)
```

```
]
```

The following code shows how to normalize the rows of the NumPy matrix:

```
from sklearn.preprocessing import normalize
```

```
#normalize matrix by rows
```

```
x_normed = normalize(x, axis=1, norm='l1')
```

```
#view normalized matrix
```

```
print(x_normed)
```

```
]
```

**Notice that the values in each row now sum to one.**

**Sum of first row:  $0 + 0.33 + 0.67 = 1$  Sum of second row:  
 $0.25 + 0.33 + 0.417 = 1$  Sum of third row:  $0.2857 + 0.3333$   
 $+ 0.3809 = 1$**

**Example 2: Normalize Columns of NumPy Matrix**

**Suppose we have the following NumPy matrix:**

```
import numpy as np
```

```
#create matrix
```

```
x = np.arange(0, 36, 4).reshape(3,3)
```

```
#view matrix
```

```
print(x)
```

```
]
```

**The following code shows how to normalize the rows of the NumPy matrix:**

```
from sklearn.preprocessing import normalize
```

```
#normalize matrix by columns  
x_normed = normalize(x, axis=0, norm='l1')  
  
#view normalized matrix  
print(x_normed)  
  
]
```

**Notice that the values in each column now sum to one.**

**Sum of first column:  $0 + 0.33 + 0.67 = 1$**   
**Sum of second column:  $0.083 + 0.333 + 0.583 = 1$**   
**Sum of third column:  $0.133 + 0.333 + 0.5333 = 1$**

**Additional Resources**

**The following tutorials explain how to perform other common operations in Python:**