# How can I use the numpy.digitize() function in Python to bin variables?

Authored by

**stats writer**

April 17, 2024

The numpy.digitize() function is a useful tool in Python for binning numerical variables. It allows you to group data into categories or bins based on a specified set of cutoff values. This function takes in an array of values and an array of bins, and assigns each value to the corresponding bin based on its numerical value. This is particularly helpful in data analysis and visualization, as it allows for easier comparison and interpretation of data. By utilizing the numpy.digitize() function, you can efficiently organize and analyze large datasets with numerical variables in Python.

## Bin Variables in Python Using numpy.digitize()

## Often you may be interested in placing the values of a variable into "bins" in Python.

## Fortunately this is easy to do using the function, which uses the following syntax:

## numpy.digitize(x, bins, right=False)

## where:

## x: Array to be binned.bins: Array of bins.right: Indicating whether the intervals include the right or the left bin edge. Default is that the interval does not include the right edge.

## This tutorial shows several examples of how to use this function in practice.

### Example 1: Place All Values into Two Bins

**The following code shows how to place the values of an array into two bins:**

**0 if x < 201 if x ≥ 20**

**import numpy as np**

**#create data**
**data =**

**#place values into bins**
**np.digitize(data, bins=)**

**array()**

### Example 2: Place All Values into Three Bins

**The following code shows how to place the values of an array into three bins:**

**0 if x < 101 if 10 ≤ x < 202 if x ≥ 20**

**import numpy as np**

**#create data**
**data =**

**#place values into bins**
**np.digitize(data, bins=)**

**array()**

**Note that if we specify right=True then the values would be placed into the following bins:**

**0 if x ≤ 101 if 10 < x ≤ 202 if x > 20**

**Each interval would include the right bin edge. Here's what that looks like:**

**import numpy as np**

**#create data**
**data =**

**#place values into bins**
**np.digitize(data, bins=, right=True)**

**array()**

### Example 3: Place All Values into Four Bins

**The following code shows how to place the values of an**

**array into three bins:**

**0 if x < 101 if 10 ≤ x < 202 if 20 ≤ x < 303 if x ≥ 30**

```
import numpy as np

#create data
data =

#place values into bins
np.digitize(data, bins=)

array()
```

### Example 4: Count the Frequency of Each Bin

**Another useful NumPy function that complements the numpy.digitize() function is the function, which counts the frequencies of each bin.**

**The following code shows how to place the values of an array into three bins and then count the frequency of each bin:**

```
import numpy as np

#create data
```

**data =**

**#place values into bins**
**bin_data = np.digitize(data, bins=)**

**#view binned data**
**bin_data**

**array()**

**#count frequency of each bin**
**np.bincount(bin_data)**

**array()**

**The output tells us that:**

**Bin "0" contains 4 data values.Bin "1" contains 2 data values.Bin "2" contains 5 data values.**

***Find more Python tutorials .***