How can we use the cut() function in R?

Authored by

stats writer

June 25, 2024

The cut() function in R is a versatile tool that allows users to divide a continuous variable into categories or groups. This function takes a numeric vector and divides it into user-defined intervals, also known as bins. These bins can be based on specific numeric values or can be evenly distributed. The resulting output is a factor variable that represents the different categories or groups. This function is useful for organizing and analyzing large datasets, as well as for creating visualizations such as histograms. Overall, the cut() function provides a convenient and efficient way to categorize and analyze continuous variables in R.

Use the cut() Function in R

The cut() function in R can be used to cut a range of values into bins and specify labels for each bin.

This function uses the following syntax:

cut(x, breaks, labels = NULL, ...)

where:

x: Name of vectorbreaks: Number of breaks to make or vector of break pointslabels: Labels for the resulting bins

The following examples show how to use this function in different scenarios with the following data frame in R:

#create data frame
df <- data.frame(player=c('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I'),

points=c(4, 7, 8, 12, 14, 16, 20, 26, 36))

#view data frame
df

player points
1 A 4
2 B 7
3 C 8
4 D 12
5 E 14
6 F 16
7 G 20
8 H 26
9 I 36

Example 1: Cut Vector Based on Number of Breaks

The following code shows how to use the cut() function to create a new column called category that cuts the points column into bins of four equal sizes:

#create new column that places each player into four categories based on points
df$category <- cut(df$points, breaks=4)

```
#view updated data frame
df

player points category
1 A 4 (3.97,12]
2 B 7 (3.97,12]
3 C 8 (3.97,12]
4 D 12 (3.97,12]
5 E 14 (12,20]
6 F 16 (12,20]
7 G 20 (12,20]
8 H 26 (20,28]
9 I 36 (28,36]
```

Since we specified breaks=4, the cut() function split the values in the points column into bins of four equal sizes.

Here is how the cut() function did this:

First, it found the difference between the largest and smallest values in the points column (36 - 4 = 32)Then, it divided this difference by 4 (32 / 4 = 8)The result is four bins each with a width of 8

Note: The lowest interval is equal to 3.97 instead of 4 because of the following functionality from the cut() :

When breaks is specified as a single number, the range of the data is divided into breaks pieces of equal length, and then the outer limits are moved away by 0.1% of the range to ensure that the extreme values both fall within the break intervals.

Example 2: Cut Vector Based on Specific Break Points

The following code shows how to use the cut() function to create a new column called category that cuts the points column based on a vector of specific break points:

```
#create new column based on specific break points
df$category <- cut(df$points, breaks=c(0, 10, 15, 20, 40))

#view updated data frame
df

player points category
1 A 4 (0,10]
2 B 7 (0,10]
3 C 8 (0,10]
```

4 D 12 (10,15]
5 E 14 (10,15]
6 F 16 (15,20]
7 G 20 (15,20]
8 H 26 (20,40]
9 I 36 (20,40]

The cut() function categorized each player into bins based on the specific vector of break points we provided.

Example 3: Cut Vector Using Specific Break Points and Labels

The following code shows how to use the cut() function to create a new column called category that cuts the points column based on a vector of specific break points with custom labels:

```
#create new column based on values in points column
df$category <- cut(df$points,
breaks=c(0, 10, 15, 20, 40),
labels=c('Bad', 'OK', 'Good', 'Great'))

#view updated data frame
df
```

player points category
1 A 4 Bad
2 B 7 Bad
3 C 8 Bad
4 D 12 OK
5 E 14 OK
6 F 16 Good
7 G 20 Good
8 H 26 Great
9 I 36 Great

The new category column classifies each player as Bad, OK, Good, or Great depending on their corresponding value in the points column.

Note: The number of labels should always be one less than the number of break points to avoid the following error:

Error in cut.default(df$points, breaks = c(0, 10, 15, 20, 40), labels = c("Bad", :
lengths of 'breaks' and 'labels' differ