How can data be split into equal sized groups in R?

Authored by

stats writer

June 25, 2024

Data splitting is a commonly used technique in statistical analysis to divide a dataset into smaller groups for easier analysis and interpretation. In R, this can be achieved by utilizing various functions and methods that allow for the creation of equal sized groups. One approach is to use the "split" function, which takes in a dataset and a specific variable to split the data on. This function then creates a list of data frames, each containing the equal sized groups based on the specified variable. Another method is using the "cut" function, which allows for the creation of equal sized groups based on the values of a continuous variable. Additionally, the "group_by" function from the dplyr package can also be used to split data into equal sized groups based on a specific variable. Overall, R provides various options for efficiently dividing data into equal sized groups, making it a powerful tool for data analysis and manipulation.

Split Data into Equal Sized Groups in R

You can use the cut_number() function from the ggplot2 package in R to split a vector into equal sized groups.

This function uses the following basic syntax:

cut_number(x, n)

where:

x: Name of numeric vector to splitn: Number of groups

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

Example: How to Split Data into Equal Sized Groups in R

Suppose we have the following data frame in R that

contains information about the points scored by 12 different basketball players

```
#create data frame
df <- data.frame(player=LETTERS,
points=c(1, 2, 2, 2, 4, 5, 7, 9, 12, 14, 15, 22))

#view data frame
df
```

```
player points
1 A 1
2 B 2
3 C 2
4 D 2
5 E 4
6 F 5
7 G 7
8 H 9
9 I 12
10 J 14
11 K 15
12 L 22
```

How to Use LETTERS Function in R

We can use the cut_number() function from the ggplot2 package to create a new column called group that splits each row in the data frame into one of three groups based on the value in the points column:

```
library(ggplot2)#create new column that splits data into three equal sized groups based on points
df$group <- cut_number(df$points, 3)
#view updated data frame
df
```

```
player points group
1 A 1
2 B 2
3 C 2
4 D 2
5 E 4 (3.33,10]
6 F 5 (3.33,10]
7 G 7 (3.33,10]
8 H 9 (3.33,10]
9 I 12 (10,22]
10 J 14 (10,22]
11 K 15 (10,22]
12 L 22 (10,22]
```

Each of the 12 players have been placed into one of three groups based on the value in the points column.

From the output we can see that there are 3 distinct groups:

group 1: points value is between 1 and 3.33.group 2: points value is between 3.33 and 10.group 3: points value is between 10 and 22.

We can see that four players have been placed into each group.

If you would like the group column to display the groups as integer values instead, you can wrap the cut_number() function in an as.numeric() function:

library(ggplot2)#create new column that splits data into three equal sized groups based on points
df$group <- as.numeric(cut_number(df$points, 3))
#view updated data frame
df

player points group
1 A 1 1
2 B 2 1

3 C 2 1

4 D 2 1

5 E 4 2

6 F 5 2

7 G 7 2

8 H 9 2

9 I 12 3

10 J 14 3

11 K 15 3

12 L 22 3

The new group column now contains the values 1, 2 and 3 to indicate which group the player belongs to.

Once again, each group contains four players.

Note: To split the points column into more than three groups, simply change the 3 in the cut_number() function to a different number.