

How can the `split()` function in R be used to split data?

Authored by
stats writer

July 1, 2024

RECOMMENDED CITATION

stats writer (2024). *How can the `split()` function in R be used to split data?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=165296>

The `split()` function in R is a powerful tool that can be used to divide a dataset into smaller subsets based on a specific criterion. This function takes in a data frame or vector and splits it into multiple subsets, based on a chosen variable or factor. This can be useful for analyzing and manipulating large datasets, as it allows for a more focused examination of specific subsets. The `split()` function is particularly useful for creating subsets based on categorical variables, such as group membership or time periods. By using the `split()` function, researchers can easily organize and compare data based on different factors, allowing for a more comprehensive understanding of the dataset. Additionally, the `split()` function can be combined with other functions in R, such as the `apply()` function, to perform further analyses on each subset. Overall, the `split()` function is an essential tool for managing and analyzing data in R.

Use split() Function in R to Split Data

The `split()` function in R can be used to split data into groups based on factor levels.

This function uses the following basic syntax:

```
split(x, f, ...)
```

where:

x: Name of the vector or data frame to divide into groups
f: A factor that defines the groupings

The following examples show how to use this function to split vectors and data frames into groups.

Example 1: Use `split()` to Split Vector Into Groups

The following code shows how to split a vector of data

values into groups based on a vector of factor levels:

```
#create vector of data values
```

```
data <- c(1, 2, 3, 4, 5, 6)
```

```
#create vector of groupings
```

```
groups <- c('A', 'B', 'B', 'B', 'C', 'C')
```

```
#split vector of data values into groups
```

```
split(x = data, f = groups)
```

```
$A
```

```
1
```

```
$B
```

```
2 3 4
```

```
$C
```

```
5 6
```

The result is three groups.

Note that you can use indexing to retrieve specific groups as well:

```
#split vector of data values into groups and only display
```

second group

split(x = data, f = groups)

\$B

2 3 4

Example 2: Use split() to Split Data Frame Into Groups

Suppose we have the following data frame in R:

#create data frame

```
df <- data.frame(team=c('A', 'A', 'A', 'B', 'B', 'B'),  
position=c('G', 'G', 'F', 'G', 'F', 'F'),  
points=c(33, 28, 31, 39, 34, 44),  
assists=c(30, 28, 24, 24, 28, 19))
```

#view data frame

df

team position points assists

1 A G 33 30

2 A G 28 28

3 A F 31 24

4 B G 39 24

5 B F 34 28

6 B F 44 19

We can use the following code to split the data frame into groups based on the 'team' variable:

```
#split data frame into groups based on 'team'  
split(df, f = df$team)
```

\$A

team position points assists

1 A G 33 30

2 A G 28 28

3 A F 31 24

\$B

team position points assists

4 B G 39 24

5 B F 34 28

6 B F 44 19

Note that we can also split the data into groups using multiple factor variables. For example, the following code shows how to split the data into groups based on the 'team' and 'position' variables:

```
#split data frame into groups based on 'team' and  
'position' variables
```

```
split(df, f = list(df$team, df$position))
```

\$A.F

team position points assists

3 A F 31 24

\$B.F

team position points assists

5 B F 34 28

6 B F 44 19

\$A.G

team position points assists

1 A G 33 30

2 A G 28 28

\$B.G

team position points assists

4 B G 39 24

The result is four groups.

Additional Resources