How can I use the unlist() function in R?

Authored by

stats writer

June 28, 2024

The unlist() function in R is a useful tool for converting a list into a vector. This function takes a nested list structure and flattens it into a single vector, by combining all the elements in the list. This can be helpful in situations where you want to manipulate or analyze data that is in a list format. By using the unlist() function, you can easily access and work with individual elements within the list, making your data more manageable. Overall, the unlist() function is a powerful tool for transforming your data and simplifying your code in R.

Use unlist() Function in R (3 Examples)

You can use the unlist() function in R to quickly convert a list to a vector.

This function uses the following basic syntax:

unlist(x)

where:

x: The name of an R object

The following examples show how to use this function in different scenarios.

Example 1: Use unlist() to Convert List to Vector

Suppose we have the following list in R:

#create list
my_list <- list(A = c(1, 2, 3),

```
B = c(4, 5),
C = 6)

#display list
my_list

$A
1 2 3

$B
4 5

$C
6
```

The following code shows how to convert a list to a vector using the unlist() function:

```
#convert list to vector
new_vector <- unlist(my_list)

#display vector
new_vector

A1 A2 A3 B1 B2 C
1 2 3 4 5 6
```

Note that you can specify use.names = FALSE to remove the names from the vector:

```
#convert list to vector
new_vector <- unlist(my_list, use.names = FALSE)

#display vector
new_vector
```

1 2 3 4 5 6

Example 2: Use unlist() to Convert List to Matrix

The following code shows how to use unlist() to convert a list to a matrix:

```
#create list
my_list <- list(1:3, 4:6, 7:9, 10:12, 13:15)

#convert list to matrix
matrix(unlist(my_list), ncol=3, byrow=TRUE)
```

1 2 3
4 5 6
7 8 9
10 11 12

13 14 15

The result is a matrix with five rows and three columns.

Example 3: Use unlist() to Sort Values in List

Suppose we have the following list in R:

```
#create list
some_list <- list(c(4, 3, 7), 2, c(5, 12, 19))

#view list
some_list

]
4 3 7

]
2

]
5 12 19
```

Now suppose we attempt to sort the values in the list:

```
#attempt to sort the values in the list
```

sort(some_list)

Error in sort.int(x, na.last = na.last, decreasing = decreasing, ...) :
'x' must be atomic

We receive an error because the list must first be converted to a vector for us to sort the values.

We can use the following unlist() function to sort the values:

#sort values in list
sort(unlist(some_list))

2 3 4 5 7 12 19

Notice that we're able to successfully sort the list of values without any error because we first used unlist(), which converted the list to a numeric vector.

Additional Resources