How can I convert a list to a vector in R, and what are some examples of doing so?

Authored by

stats writer

May 2, 2024

Converting a list to a vector in R is a simple and efficient way to organize and manipulate data. A vector is a one-dimensional data structure that can hold multiple elements of the same data type. To convert a list to a vector in R, the "unlist" function can be used, which combines all the elements of the list into a single vector. This can be useful for performing calculations, filtering data, or creating plots.

For example, if we have a list of temperatures recorded over a week, we can convert it to a vector using the "unlist" function. This will create a vector with all the temperature values, which can then be used to calculate the average temperature, plot a line graph, or perform other operations.

Another example is converting a list of student grades to a vector. This can help in analyzing the overall performance of the class, identifying the highest and lowest grades, or finding the median grade.

In summary, converting a list to a vector in R simplifies data manipulation and allows for efficient analysis and visualization. It is a useful technique for handling large datasets and is widely used in data science and statistical analysis.

Convert List to Vector in R (With Examples)

You can use one of the following methods to convert a list to a vector in R:

#use unlist() function
new_vector <- unlist(my_list, use.names = FALSE)

#use flatten_*() function from purrr library
new_vector <- purrr::flatten(my_list)

The following examples show how to use each of these methods in practice with the following list:

```
#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
```

Example 1: Convert List to Vector Using unlist() Function

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: Convert List to Vector Using flatten_* Function

The following code shows how to convert a list to a vector using the family of flatten_* functions from the package:

```
library(purrr)

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

#display vector
new_vector

1 2 3 4 5 6

The flatten_dbl() function specifically converts the list to a vector of type double.

Note that we could use flatten_chr() to convert a character list to a vector of type character:

library(purrr)

#define character list
my_char_list <- list(A = c('a', 'b', 'c'),
B = c('d', 'e'),
C = 'f')

#convert character list to character vector
new_char_vector <- flatten_chr(my_char_list)

#display vector
new_char_vector

"a" "b" "c" "d" "e" "f"

Check out for a complete list of the family of flatten_* functions.

Note: If you're working with an extremely large list, the flatten_* functions will perform quicker than the unlist() function from base R.