

How can I use the rbindlist function in R to combine multiple data tables into one?

Authored by
stats writer

June 29, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use the rbindlist function in R to combine multiple data tables into one?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=159951>

The rbindlist function in R allows users to efficiently combine multiple data tables into one. This function can be used to merge data tables with the same columns and order, or to append new columns to an existing table. It is particularly useful for working with large datasets as it is faster than other merging methods. By using the rbindlist function, users can easily and effectively organize and manipulate their data in a single table. This function is a powerful tool for data analysis and management in R.

Use rbindlist in R to Make One Data Table from Many

The rbindlist() function in R can be used to create one data.table from a list of many data.table or data.frame objects.

This function uses the following basic syntax:

```
rbindlist(l, use.names="check", fill=FALSE, idcol=NULL)
```

where:

l: List containing data.table, data.frame, or list objects.
use.names: TRUE binds by column names. FALSE binds by position.
fill: TRUE fills missing values with NA.
idcol: Creates column showing which list item those rows came from.

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

Example: Use rbindlist to Make One Data Table

Suppose we have the following list of data.table and data.frame objects in R:

```
library(data.table)
```

```
#create data frames and data tables
```

```
data1 <- data.table(team=c('A', 'B', 'C'),  
points=c(22, 27, 38))
```

```
data2 <- data.table(team=c('D', 'E', 'F'),  
points=c(22, 14, 20))
```

```
data3 <- data.frame(team=c('G', 'H', 'I'),  
points=c(11, 15, 18))
```

```
#view data frames and data tables
```

```
print(data1)
```

```
print(data2)
```

```
print(data3)
```

```
team points
```

```
1: A 22
```

```
2: B 27
```

```
3: C 38
```

team points

1: D 22

2: E 14

3: F 20

team points

1 G 11

2 H 15

3 I 18

We can use the following `rbindlist()` function to bind together the list of `data.table` and `data.frame` objects into one `data.table`:

```
#define list of objects to bind together  
data_list <- list(data1, data2, data3)
```

```
#bind together list of objects  
big_data <- rbindlist(data_list)
```

```
#view result  
big_data
```

team points

1: A 22

2: B 27

3: C 38

4: D 22

5: E 14

6: F 20

7: G 11

8: H 15

9: I 18

The result is a `data.table` object with nine rows that are composed of the rows from the list of objects we provided.

We can also use the `class()` function to verify that the result is indeed a `data.table` object:

```
#view class of resulting object  
class(big_data)
```

```
"data.table" "data.frame"
```

We can see that the result is indeed a `data.table` object.

The Benefit of Using `rbindlist`

The alternative to using `rbindlist` would be to use `do.call` with the function in base R:

```
#use rbind to bind together list of objects  
do.call("rbind", data_list)
```

team points

1: A 22

2: B 27

3: C 38

4: D 22

5: E 14

6: F 20

7: G 11

8: H 15

9: I 18

This code produces the same result but it turns out that rbindlist is significantly faster, especially for extremely large data.table or data.frame objects.

Additional Resources