# What is the difference between merge() and join() in R?

Authored by

**stats writer**

June 29, 2024

Merge() and join() are two commonly used functions in the R programming language for combining data frames. While both functions serve a similar purpose, there are some key differences between them.

The merge() function is used to combine two or more data frames based on a common variable or set of variables. It creates a new data frame by matching observations from each data frame and combining them into a single row. This function is useful for joining data frames that have different variables but share a common identifier.

On the other hand, the join() function is used to combine data frames based on a common set of variables. It is similar to the merge() function, but it provides more flexibility in terms of the type of join to be performed (left, right, inner, or full) and the variables to be used for joining. Additionally, join() supports joining multiple data frames at once, whereas merge() can only join two data frames at a time.

In summary, the main difference between merge() and join() in R is that merge() is more suitable for combining data frames with a single common variable, while join() is better for joining data frames based on multiple common variables and performing different types of joins.



## The Difference Between merge() vs. join() in R

## The merge() function in base R and the various join() functions from the package can both be used to join two data frames together.

## There are two main differences between these two functions:

## 1. The join() functions from dplyr tend to be much faster than merge() on extremely large data frames.

## 2. The join() functions from dplyr preserve the original order of rows in the data frames while the merge()

**function automatically sorts the rows alphabetically based on the column you used to perform the join.**

**The following example illustrates difference #2 in practice.**

### Example: The Difference Between merge() and join()

**Suppose we have the following two data frames in R:**

```
#define first data frame
df1 <- data.frame(team=c('Mavs', 'Hawks', 'Spurs', 'Nets'),
points=c(99, 93, 96, 104))

df1

team points
1 Mavs 99
2 Hawks 93
3 Spurs 96
4 Nets 104

#define second data frame
df2 <- data.frame(team=c('Mavs', 'Hawks', 'Spurs', 'Nets'),
```

```
assists=c(19, 18, 22, 25))

df2

team assists
1 Mavs 19
2 Hawks 18
3 Spurs 22
4 Nets 25
```

Suppose we use the merge() function in base R to perform a left join, using the 'team' column as the column to join on:

```
#perform left join using base R
merge(df1, df2, by='team', all.x=TRUE)

team points assists
1 Hawks 93 18
2 Mavs 99 19
3 Nets 104 25
4 Spurs 96 22
```

Notice that the rows are sorted in alphabetical order based on the values in the 'team' column.

**Now suppose we use the left_join() function from dplyr to perform a left join, again using the 'team' column as the column to join on:**

```
library(dplyr)
#perform left join using dplyr
left_join(df1, df2, by='team')

team points assists
1 Mavs 99 19
2 Hawks 93 18
3 Spurs 96 22
4 Nets 104 25
```

**Notice that the order of the rows match the original order of the rows in the data frame before performing the left join.**

### Additional Resources