

How can multiple data frames be merged in R? Can you provide examples?

Authored by
stats writer

May 15, 2024

RECOMMENDED CITATION

stats writer (2024). *How can multiple data frames be merged in R? Can you provide examples?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=144141>

In R, multiple data frames can be merged using the "merge" function. This function combines two or more data frames based on a common variable or column. The resulting merged data frame will contain all the rows and columns from the original data frames.

For example, let's say we have two data frames: "df1" with columns "ID", "Name", and "Age" and "df2" with columns "ID", "Gender", and "Occupation". To merge these two data frames based on the "ID" column, we can use the following code:

```
merged_df <- merge(df1, df2, by = "ID")
```

This will create a new data frame "merged_df" with columns "ID", "Name", "Age", "Gender", and "Occupation". If there are any matching values in the "ID" column, the corresponding rows from both data frames will be merged into one row in the new data frame.

In addition, the "merge" function also allows for more advanced merging options such as merging based on multiple columns or choosing specific columns to merge. By using this function, multiple data frames can be efficiently merged in R, allowing for easier data analysis and manipulation.

Merge Multiple Data Frames in R (With Examples)

You can use one of the following two methods to merge multiple data frames in R:

Method 1: Use Base R

#put all data frames into list

```
df_list <- list(df1, df2, df3)
```

#merge all data frames in list

```
Reduce(function(x, y) merge(x, y, all=TRUE), df_list)
```

Method 2: Use Tidyverse

```
library(tidyverse)
```

```
#put all data frames into list
```

```
df_list <- list(df1, df2, df3)
```

```
#merge all data frames in list
```

```
df_list %>% reduce(full_join, by='variable_name')
```

The following examples show how to use each method in practice.

Method 1: Merge Multiple Data Frames Using Base R

Suppose we have the following data frames in R:

```
#define data frames
```

```
df1 <- data.frame(id=c(1, 2, 3, 4, 5),  
revenue=c(34, 36, 40, 49, 43))
```

```
df2 <- data.frame(id=c(1, 2, 5, 6, 7),  
expenses=c(22, 26, 31, 40, 20))
```

```
df3 <- data.frame(id=c(1, 2, 4, 5, 7),  
profit=c(12, 10, 14, 12, 9))
```

We can use the following syntax to merge all of the data

frames using functions from base R:

```
#put all data frames into list
```

```
df_list <- list(df1, df2, df3)
```

```
#merge all data frames together
```

```
Reduce(function(x, y) merge(x, y, all=TRUE), df_list)
```

```
id revenue expenses profit
```

```
1 1 34 22 12
```

```
2 2 36 26 10
```

```
3 3 40 NA NA
```

```
4 4 49 NA 14
```

```
5 5 43 31 12
```

```
6 6 NA 40 NA
```

```
7 7 NA 20 9
```

Notice that each of the "id" values from each original data frame is included in the final data frame.

Method 2: Merge Multiple Data Frames Using Tidyverse

Suppose we have the following data frames in R:

```
#define data frames
```

```
df1 <- data.frame(id=c(1, 2, 3, 4, 5),
```

```
revenue=c(34, 36, 40, 49, 43))
```

```
df2 <- data.frame(id=c(1, 2, 5, 6, 7),  
expenses=c(22, 26, 31, 40, 20))
```

```
df3 <- data.frame(id=c(1, 2, 4, 5, 7),  
profit=c(12, 10, 14, 12, 9))
```

We can use the following syntax to merge all of the data frames using functions from `dplyr` - a collection of packages designed for data science in R:

```
library(tidyverse)
```

```
#put all data frames into list
```

```
df_list <- list(df1, df2, df3)
```

```
#merge all data frames together
```

```
df_list %>% reduce(full_join, by='id')
```

```
id revenue expenses profit
```

```
1 1 34 22 12
```

```
2 2 36 26 10
```

```
3 3 40 NA NA
```

```
4 4 49 NA 14
```

```
5 5 43 31 12
```

6 6 NA 40 NA

7 7 NA 20 9

Note: The tidyverse approach will be noticeably quicker if you're working with extremely large data frames.

The following tutorials explain how to perform other common functions in R:

ARABPSYCHOLOGY.COM