

How can an outer join be performed in R, and do you have any examples of it being used?

Authored by
stats writer

June 29, 2024

RECOMMENDED CITATION

stats writer (2024). *How can an outer join be performed in R, and do you have any examples of it being used?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=158891>

An outer join in R is a method of combining two datasets based on a common variable, while also including any unmatched observations from one or both datasets. This type of join is useful for comparing and analyzing data from multiple sources.

To perform an outer join in R, the "merge" function can be used. This function allows the user to specify the type of join (left, right, or full) and the key variable to merge on. The resulting merged dataset will contain all the observations from both datasets, with missing values filled in for unmatched observations.

An example of an outer join in R would be merging a dataset of customer information with a dataset of product purchases. By using a left outer join, all customers would be included in the resulting dataset, with any missing product purchase information filled in with "NA" values.

In summary, an outer join in R is a useful tool for combining datasets and analyzing data from multiple sources. Its flexibility allows for various types of joins and can help identify relationships and patterns in the data.

Do an Outer Join in R (With Examples)

There are two common ways to perform an outer join in R:

Method 1: Use Base R

```
merge(df1, df2, by='column_to_join_on', all=TRUE)
```

Method 2: Use dplyr

```
library(dplyr)
```

```
full_join(df1, df2, by='column_to_join_on')
```

Each method will return all rows from both tables.

Both methods will produce the same result, but the dplyr method will tend to work faster on extremely large datasets.

The following examples show how to use each of these functions in practice with the following data frames:

#define first data frame

```
df1 = data.frame(team=c('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'),  
points=c(18, 22, 19, 14, 14, 11, 20, 28))
```

df1

team points

1 A 18

2 B 22

3 C 19

4 D 14

5 E 14

6 F 11

7 G 20

8 H 28

#define second data frame

```
df2 = data.frame(team=c('A', 'B', 'C', 'D', 'L', 'M'),  
assists=c(4, 9, 14, 13, 10, 8))
```

```
df2
```

```
team assists
```

```
1 A 4
```

```
2 B 9
```

```
3 C 14
```

```
4 D 13
```

```
5 L 10
```

```
6 M 8
```

Example 1: Outer Join Using Base R

We can use the `merge()` function in base R to perform an outer join, using the 'team' column as the column to join on:

```
#perform outer join using base R
```

```
df3 <- merge(df1, df2, by='team', all=TRUE)
```

```
#view result
```

```
df3
```

```
team points assists
```

1 A 18 4
2 B 22 9
3 C 19 14
4 D 14 13
5 E 14 NA
6 F 11 NA
7 G 20 NA
8 H 28 NA
9 L NA 10
10 M NA 8

Notice that all of the rows from both data frames are returned.

Example 2: Outer Join Using dplyr

We can use the `full_join()` function from the package to perform an outer join, using the 'team' column as the column to join on:

```
library(dplyr)
```

```
#perform outer join using dplyr
```

```
df3 <- full_join(df1, df2, by='team')
```

```
#view result
```

df3

team points assists

1 A 18 4

2 B 22 9

3 C 19 14

4 D 14 13

5 E 14 NA

6 F 11 NA

7 G 20 NA

8 H 28 NA

9 L NA 10

10 M NA 8

Notice that this matches the result we obtained from using the merge() function in base R.

Additional Resources