

# How do I perform a left join in dplyr when the column names are different in the two data frames?

Authored by  
**stats writer**

June 23, 2024

## RECOMMENDED CITATION

stats writer (2024). *How do I perform a left join in dplyr when the column names are different in the two data frames?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=148447>

A left join in dplyr is a method used to combine two data frames based on a common column or key. In cases where the column names are different in the two data frames, the join can still be performed by using the "by" argument in the dplyr function. This argument allows the user to specify which columns to match on, even if they have different names in each data frame. This ensures that the join is performed accurately and no data is lost. By using the "by" argument, the user can perform a left join and merge the data from both data frames while accounting for any differences in column names.

## Left Join in dplyr with Different Column Names

You can use the following basic syntax in dplyr to perform a left join on two data frames when the columns you're joining on have different names in each data frame:

```
library(dplyr)
```

```
final_df <- left_join(df_A, df_B, by = c('team' =  
'team_name'))
```

This particular example will perform a left join on the data frames called df\_A and df\_B, joining on the column in df\_A called team and the column in df\_B called team\_name.

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

## Example: Perform Left Join with Different Column Names in dplyr

Suppose we have the following two data frames in R:

```
#create first data frame
```

```
df_A <- data.frame(team=c('A', 'B', 'C', 'D', 'E'),  
points=c(22, 25, 19, 14, 38))
```

```
df_A
```

```
team points
```

```
1 A 22
```

```
2 B 25
```

```
3 C 19
```

```
4 D 14
```

```
5 E 38
```

```
#create second data frame
```

```
df_B <- data.frame(team=c('A', 'C', 'D', 'F', 'G'),  
rebounds=c(14, 8, 8, 6, 9))
```

```
df_B
```

```
team_name rebounds
```

```
1 A 14
```

```
2 C 8
```

3 D 8

4 F 6

5 G 9

We can use the following syntax in dplyr to perform a left join based on matching values in the team column of df\_A and the team\_name column of df\_B:

```
library(dplyr)
```

```
#perform left join based on different column names in  
df_A and df_B
```

```
final_df <- left_join(df_A, df_B, by = c('team' =  
'team_name'))
```

```
#view final data frame
```

```
final_df
```

```
team points rebounds
```

```
1 A 22 14
```

```
2 B 25 NA
```

```
3 C 19 8
```

```
4 D 14 8
```

```
5 E 38 NA
```

The resulting data frame contains all rows from `df_A` and only the rows in `df_B` where the team values matched the `team_name` values.

Note that you can also match on multiple columns with different names by using the following basic syntax:

```
library(dplyr)
```

```
#perform left join based on multiple different column names
```

```
final_df <- left_join(df_A, df_B, by = c('A1' = 'B1', 'A2' = 'B2', 'A3' = 'B3'))
```

Note: You can find the complete documentation for the `left_join()` function in `dplyr` .