

How can I add columns to a data frame in R using dplyr?

Authored by
stats writer

May 6, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I add columns to a data frame in R using dplyr?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=143340>

Dplyr is a popular R package used for data manipulation and analysis. It provides a simple and efficient way to add columns to a data frame. To do so, one must first load the dplyr package and then use the "mutate" function, which allows for the creation of new columns based on existing ones. The new column can be specified with a desired name and the desired values can be generated using various functions or operations. This process can be further customized by using other dplyr functions such as "transmute" or "transmute_all". Overall, using dplyr makes the process of adding columns to a data frame in R more streamlined and efficient.

Add Columns to Data Frame in R Using dplyr

You can use the `mutate()` function from the dplyr package to add one or more columns to a data frame in R.

This function uses the following basic syntax:

Method 1: Add Column at End of Data Frame

```
df %>%  
mutate(new_col=c(1, 3, 3, 5, 4))
```

Method 2: Add Column Before Specific Column

```
df %>%  
mutate(new_col=c(1, 3, 3, 5, 4),  
.before=col_name)
```

Method 3: Add Column After Specific Column

```
df %>%  
mutate(new_col=c(1, 3, 3, 5, 4),  
.after=col_name)
```

Method 4: Add Column Based on Other Columns

```
df %>%  
mutate(new_col= if_else(.col_name > 10, 'A', 'B'))
```

The following examples show how to use this syntax in practice with the following data frame:

```
#create data frame  
df <- data.frame(team=c('A', 'A', 'A', 'A', 'B', 'B', 'B', 'B'),  
points=c(12, 14, 19, 24, 24, 22, 30, 9),  
assists=c(4, 6, 6, 8, 3, 7, 8, 11))
```

```
#view data frame
```

```
df
```

```
team points assists
```

```
1 A 12 4
```

```
2 A 14 6
```

```
3 A 19 6
```

```
4 A 24 8
```

5 B 24 3

6 B 22 7

7 B 30 8

8 B 9 11

Example 1: Add Column at End of Data Frame

The following code shows how to add a column at the end of the data frame:

```
#add 'blocks' column at end of data frame
```

```
df <- df %>%
```

```
mutate(blocks=c(1, 3, 3, 2, 4, 3, 6, 2))
```

```
#view data frame
```

```
df
```

```
team points assists blocks
```

```
1 A 12 4 1
```

```
2 A 14 6 3
```

```
3 A 19 6 3
```

```
4 A 24 8 2
```

```
5 B 24 3 4
```

```
6 B 22 7 3
```

```
7 B 30 8 6
```

8 B 9 11 2

Note that you can add an empty column by simply assigning NA to every value in the new column:

#add empty column at end of data frame

```
df <- df %>%
```

```
mutate(blocks=NA)
```

#view data frame

```
df
```

```
team points assists blocks
```

```
1 A 12 4 NA
```

```
2 A 14 6 NA
```

```
3 A 19 6 NA
```

```
4 A 24 8 NA
```

```
5 B 24 3 NA
```

```
6 B 22 7 NA
```

```
7 B 30 8 NA
```

```
8 B 9 11 NA
```

Example 2: Add Column Before Specific Column

The following code shows how to add a column before

a specific column in the data frame:

```
#add 'blocks' column before 'points' column
```

```
df <- df %>%
```

```
mutate(blocks=c(1, 3, 3, 2, 4, 3, 6, 2),
```

```
.before=points)
```

```
#view data frame
```

```
df
```

```
team blocks points assists
```

```
1 A 1 12 4
```

```
2 A 3 14 6
```

```
3 A 3 19 6
```

```
4 A 2 24 8
```

```
5 B 4 24 3
```

```
6 B 3 22 7
```

```
7 B 6 30 8
```

```
8 B 2 9 11
```

Example 3: Add Column After Specific Column

The following code shows how to add a column after a specific column in the data frame:

```
#add 'blocks' column after 'points' column
```

```
df <- df %>%  
mutate(blocks=c(1, 3, 3, 2, 4, 3, 6, 2),  
.after=points)
```

```
#view data frame
```

```
df
```

```
team points blocks assists
```

```
1 A 12 1 4
```

```
2 A 14 3 6
```

```
3 A 19 3 6
```

```
4 A 24 2 8
```

```
5 B 24 4 3
```

```
6 B 22 3 7
```

```
7 B 30 6 8
```

```
8 B 9 2 11
```

Example 4: Add Column Based on Other Columns

The following code shows how to add a column based on another column in the data frame:

```
#add 'status' column whose values depend on value in  
'points' column  
df <- df %>%
```

```
mutate(status= if_else(.$points > 20, 'Good', 'Bad'))
```

```
#view data frame
```

```
df
```

```
team points assists status
```

```
1 A 12 4 Bad
```

```
2 A 14 6 Bad
```

```
3 A 19 6 Bad
```

```
4 A 24 8 Good
```

```
5 B 24 3 Good
```

```
6 B 22 7 Good
```

```
7 B 30 8 Good
```

```
8 B 9 11 Bad
```

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