

How can I use dplyr to calculate the mean for multiple columns?

Authored by
stats writer

June 25, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use dplyr to calculate the mean for multiple columns?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=151733>

Dplyr is a popular open-source R package that allows for efficient and intuitive data manipulation. It includes a variety of functions that can be used to easily calculate summary statistics, such as means, for multiple columns in a dataset. By using the dplyr function "summarize", users can specify the columns they wish to calculate the mean for, and the results will be returned in a new data frame. This feature of dplyr makes it a powerful tool for handling large datasets and performing quick and accurate calculations.

Calculate Mean for Multiple Columns Using dplyr

You can use the following syntax to calculate the mean value for multiple specific columns in a data frame using the dplyr package in R:

```
library(dplyr)df %>%  
rowwise() %>%  
mutate(game_mean = mean(c_across(c('game1',  
'game2', 'game3'))), na.rm=TRUE))
```

This particular example calculates the mean value of each row for only the columns named game1, game2, and game3 in the data frame.

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

Example: Calculate Mean for Multiple Columns Using dplyr

Suppose we have the following data frame that shows

the points scored by various basketball players in three different games:

#create data frame

```
df <- data.frame(team=c('A', 'A', 'A', 'B', 'B', 'B', 'C', 'C'),  
game1=c(10, 12, 17, 18, 24, 29, 29, 34),  
game2=c(8, 10, 14, 15, NA, 19, 18, 29),  
game3=c(4, 5, 5, 9, 12, 12, 18, 20))
```

#view data frame

df

```
team game1 game2 game3
```

```
1 A 10 8 4
```

```
2 A 12 10 5
```

```
3 A 17 14 5
```

```
4 B 18 15 9
```

```
5 B 24 NA 12
```

```
6 B 29 19 12
```

```
7 C 29 18 18
```

```
8 C 34 29 20
```

We can use the following syntax to calculate the mean value of each row for only the game1, game2 and game3 columns:

```
library(dplyr)#calculate mean value in each row for
game1, game2 and game3 columns
df %>%
rowwise() %>%
mutate(game_mean = mean(c_across(c('game1',
'game2', 'game3'))), na.rm=TRUE))

# A tibble: 8 x 5
# Rowwise:
team game1 game2 game3 game_mean

1 A 10 8 4 7.33
2 A 12 10 5 9
3 A 17 14 5 12
4 B 18 15 9 14
5 B 24 NA 12 18
6 B 29 19 12 20
7 C 29 18 18 21.7
8 C 34 29 20 27.7
```

The column called `game_mean` displays the mean value in each row across the `game1`, `game2` and `game3` columns.

For example:

Mean value of row 1: $(10 + 8 + 4) / 3 = 7.33$
 Mean value of row 2: $(12 + 10 + 5) / 3 = 9$
 Mean value of row 3: $(17 + 14 + 5) / 3 = 12$

And so on.

Note that we could also use the `starts_with()` function to specify that we'd like to calculate the mean value of each row for only the columns that start with 'game' in the column name:

```
library(dplyr)#calculate mean value in each row for
columns that start with 'game'
df %>%
rowwise() %>%
mutate(game_mean =
mean(c_across(c(starts_with('game'))), na.rm=TRUE))
```

A tibble: 8 x 5

Rowwise:

```
team game1 game2 game3 game_mean
```

```
1 A 10 8 4 7.33
```

```
2 A 12 10 5 9
```

```
3 A 17 14 5 12
```

4 B 18 15 9 14

5 B 24 NA 12 18

6 B 29 19 12 20

7 C 29 18 18 21.7

8 C 34 29 20 27.7

Notice that this syntax produces the same results as the previous example.

ARABPSYCHOLOGY.COM