

How can I use dplyr to apply a function to each row of a data frame?

Authored by
stats writer

June 27, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use dplyr to apply a function to each row of a data frame?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=155700>

Dplyr is a popular R package that allows for efficient and effective manipulation of data frames. One useful feature of dplyr is the ability to apply a function to each row of a data frame. This can be done using the "rowwise()" function, which groups the data frame by row and allows for the application of any desired function. This is particularly useful for tasks such as data cleaning or creating new variables based on existing ones. By using dplyr's rowwise function, users can easily perform repetitive tasks on each row of a data frame, simplifying data analysis and reducing the need for manual coding.

Apply Function to Each Row Using dplyr

You can use the following basic syntax to apply a function to each row in a data frame in R using functions from dplyr:

```
df %>%  
  rowwise() %>%  
  mutate(mean_value = mean(c(col1, col2, col3),  
    na.rm=TRUE))
```

This particular example calculates the mean value of col1, col2, and col3 for each row in the data frame, but you can replace the mean() function with any function you'd like to calculate a different metric.

The following examples show how to use this syntax in practice with the following data frame that contains information about points scored by various basketball

players during different games:

#create data frame

```
df <- data.frame(game1=c(22, 25, 29, 13, 22, 30),  
game2=c(12, 10, 6, 6, 8, 11),  
game3=c(NA, 15, 15, 18, 22, 13))
```

#view data frame

df

game1 game2 game3

1 22 12 NA

2 25 10 15

3 29 6 15

4 13 6 18

5 22 8 22

6 30 11 13

Example 1: Mean of Specific Columns in Each Row

The following code shows how to calculate the mean value of the game1 and game3 columns for each row in the data frame:

library(dplyr)

```
#calculate mean of game1 and game3
df %>%
rowwise() %>%
mutate(mean_points = mean(c(game1, game3),
na.rm=TRUE))
```

```
# A tibble: 6 x 4
```

```
# Rowwise:
```

```
game1 game2 game3 mean_points
```

```
1 22 12 NA 22
```

```
2 25 10 15 20
```

```
3 29 6 15 22
```

```
4 13 6 18 15.5
```

```
5 22 8 22 22
```

```
6 30 11 13 21.5
```

From the output we can see:

The mean value of game1 and game3 in the first row is 22. The mean value of game1 and game3 in the second row is 20. The mean value of game1 and game3 in the third row is 22.

And so on.

Example 2: Max of Specific Columns in Each Row

The following code shows how to calculate the max value of the game2 and game3 columns for each row in the data frame:

```
library(dplyr)

#calculate max of game2 and game3
df %>%
  rowwise() %>%
  mutate(max_points = max(c(game2, game3),
    na.rm=TRUE))

# A tibble: 6 x 4
# Rowwise:
  game1 game2 game3 max_points
  <dbl> <dbl> <dbl> <dbl>
1 22 12 NA 12
2 25 10 15 15
3 29 6 15 15
4 13 6 18 18
5 22 8 22 22
6 30 11 13 13
```

From the output we can see:

The max value of game2 and game3 in the first row is 12. The max value of game2 and game3 in the second row is 15. The max value of game2 and game3 in the third row is 15.

And so on.

Example 3: Standard Deviation of Specific Columns in Each Row

The following code shows how to calculate the standard deviation of the values in the game2 and game3 columns for each row in the data frame:

```
library(dplyr)

#calculate standard deviation of game2 and game3
df %>%
  rowwise() %>%
  mutate(sd_points = sd(c(game2, game3), na.rm=TRUE))

# A tibble: 6 x 4
# Rowwise:
  game1 game2 game3 sd_points
1 22 12 NA NA
```

2 25 10 15 3.54

3 29 6 15 6.36

4 13 6 18 8.49

5 22 8 22 9.90

6 30 11 13 1.41

From the output we can see:

The standard deviation of game2 and game3 in the first row is NA (since standard deviation can't be calculated from only one value).The standard deviation of game2 and game3 in the second row is 3.54.The standard deviation of game2 and game3 in the first row 6.36.

And so on.

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

Additional Resources

The following tutorials explain how to perform other common tasks using dplyr: