How can I use the case_when() function in dplyr to efficiently create conditional variables in my dataset?

Authored by

stats writer

May 1, 2024

The case_when() function in the dplyr package allows for efficient creation of conditional variables in a dataset. This function can be used to specify multiple conditions and corresponding values, which are then evaluated and assigned to the new variable. With its ability to handle multiple conditions in a single statement, the case_when() function streamlines the process of creating conditional variables in a dataset, making it a valuable tool for data manipulation and analysis. By utilizing this function, users can efficiently transform their data based on specific conditions, resulting in a more organized and comprehensive dataset for further analysis.

Use case_when() in dplyr

The case_when() function from the package in R can be used to create new variables from existing variables.

This function uses the following basic syntax:

```
library(dplyr)

df %>%
mutate(new_var = case_when(var1 < 15 ~ 'low',
var2 < 25 ~ 'med',
TRUE ~ 'high'))
```

Note that TRUE is equivalent to an "else" statement.

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

```
#create data frame
```

```
df <- data.frame(player = c('AJ', 'Bob', 'Chad', 'Dan', 'Eric', 'Frank'),
position = c('G', 'F', 'F', 'G', 'C', NA),
points = c(12, 15, 19, 22, 32, NA),
assists = c(5, 7, 7, 12, 11, NA))

#view data frame
df

player position points assists
1 AJ G 12 5
2 Bob F 15 7
3 Chad F 19 7
4 Dan G 22 12
5 Eric C 32 11
6 Frank NA NA NA
```

Example 1: Create New Variable from One Existing Variable

The following code shows how to create a new variable called quality whose values are derived from the points column:

```
df %>%
mutate(quality = case_when(points > 20 ~ 'high',
```

points > 15 ~ 'med',
TRUE ~ 'low'))

player position points assists quality
1 AJ G 12 5 low
2 Bob F 15 7 low
3 Chad F 19 7 med
4 Dan G 22 12 high
5 Eric C 32 11 high
6 Frank NA NA NA low

Here is exactly how the case_when() function created the values for the new column:

If the value in the points column is greater than 20, then the value in the quality column is "high"Else, if the value in the points column is greater than 15, then the value in the quality column is "med"Else, if the value in the points column is less than or equal to 15 (or a missing value like NA), then the value in the quality column is "low"

Example 2: Create New Variable from Multiple Variables

The following code shows how to create a new variable

called quality whose values are derived from both the points and assists column:

```
df %>%
mutate(quality = case_when(points > 15 & assists > 10 ~ 'great',
points > 15 & assists > 5 ~ 'good',
TRUE ~ 'average' ))
```

```
player position points assists quality
1 AJ G 12 5 average
2 Bob F 15 7 average
3 Chad F 19 7 good
4 Dan G 22 12 great
5 Eric C 32 11 great
6 Frank NA NA NA average
```

Note that we can also use the is.na() function to explicitly assign strings to NA values:

```
df %>%
mutate(quality = case_when(is.na(points) ~ 'missing',
points > 15 & assists > 10 ~ 'great',
points > 15 & assists > 5 ~ 'good',
```

```
TRUE ~ 'average' ))

player position points assists quality
1 AJ G 12 5 average
2 Bob F 15 7 average
3 Chad F 19 7 good
4 Dan G 22 12 great
5 Eric C 32 11 great
6 Frank NA NA NA missing
```