

How can I use the mutate() function in dplyr to change factor levels in my dataset?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use the mutate() function in dplyr to change factor levels in my dataset?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=153599>

The mutate() function in dplyr is a useful tool for manipulating factor levels in a dataset. This function allows users to change the levels of a factor variable by specifying the desired changes in the code. By using mutate(), users can easily modify the levels of a factor variable without altering the original dataset. This function is particularly helpful for data cleaning and analysis, as it allows for efficient and flexible changes to be made to factor levels in a dataset. Overall, the mutate() function is a valuable tool for manipulating factor levels and enhancing the accuracy and efficiency of data analysis.

dplyr: Change Factor Levels Using mutate()

You can use the following basic syntax in to change the levels of a factor variable by using the mutate() function:

```
library(dplyr)
```

```
df <- df %>% mutate(team=recode(team,  
'H' = 'Hawks',  
'M' = 'Mavs',  
'C' = 'Cavs'))
```

This particular syntax makes the following changes to the team variable in the data frame:

'H' becomes 'Hawks' 'M' becomes 'Mavs' 'C' becomes 'Cavs'

The following example shows how to use this syntax in

practice.

Example: Change Factor Levels Using mutate()

Suppose we have the following data frame in R that contains information about various basketball players:

```
#create data frame
```

```
df <- data.frame(team=factor(c('H', 'H', 'M', 'M', 'C', 'C')),  
points=c(22, 35, 19, 15, 29, 23))
```

```
#view data frame
```

```
df
```

```
team points
```

```
1 H 22
```

```
2 H 35
```

```
3 M 19
```

```
4 M 15
```

```
5 C 29
```

```
6 C 23
```

We can use the following syntax with the mutate() function from the dplyr package to change the levels of the team variable:

```
library(dplyr)
```

```
#change factor levels of team variable
```

```
df <- df %>% mutate(team=recode(team,  
'H' = 'Hawks',  
'M' = 'Mavs',  
'C' = 'Cavs'))
```

```
#view updated data frame
```

```
df
```

```
team points
```

```
1 Hawks 22
```

```
2 Hawks 35
```

```
3 Mavs 19
```

```
4 Mavs 15
```

```
5 Cavs 29
```

```
6 Cavs 23
```

Using this syntax, we were able to make the following changes to the team variable in the data frame:

'H' becomes 'Hawks'
'M' becomes 'Mavs'
'C' becomes 'Cavs'

We can verify that the factor levels have been changed by using the levels() function:

```
#display factor levels of team variable  
levels(df$team)
```

```
"Cavs" "Hawks" "Mavs"
```

Also note that you can choose to change just one factor level instead of all of them.

For example, we can use the following syntax to only change 'H' to 'Hawks' and leave the other factor levels unchanged:

```
library(dplyr)
```

```
#change one factor level of team variable
```

```
df <- df %>% mutate(team=recode(team, 'H' = 'Hawks'))
```

```
#view updated data frame
```

```
df
```

```
team points
```

```
1 Hawks 22
```

```
2 Hawks 35
```

3 M 19

4 M 15

5 C 29

6 C 23

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

ARABPSYCHOLOGY.COM