

How can the droplevels function be used in R, and what are some examples of its application?

Authored by
stats writer

April 25, 2024

RECOMMENDED CITATION

stats writer (2024). *How can the droplevels function be used in R, and what are some examples of its application?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=139127>

The droplevels function in R is a useful tool for managing categorical variables in data sets. It allows users to remove unused levels from a factor variable, making the data more manageable and accurate for analysis. This function can be applied in various scenarios, such as cleaning messy data, removing irrelevant categories, or preparing data for modeling. For example, if a dataset includes a factor variable for different countries, but only a few countries are relevant for the analysis, the droplevels function can be used to remove the unused countries and create a more concise and relevant dataset. Additionally, it can also be used to avoid errors in statistical modeling, as some models may not be able to handle unused levels in factor variables. Overall, the droplevels function is a practical tool for organizing and preparing data for analysis in R.

Use the droplevels Function in R (With Examples)

The droplevels() function in R can be used to drop unused factor levels.

This function is particularly useful if we want to drop factor levels that are no longer used due to subsetting a vector or a data frame.

This function uses the following syntax:

droplevels(x)

where x is an object from which to drop unused factor levels.

This tutorial provides a couple examples of how to use this function in practice.

Example 1: Drop Unused Factor Levels in a Vector

Suppose we create a vector of data with five factor levels. Then suppose we define a new vector of data with just three of the original five factor levels.

```
#define data with 5 factor levels
```

```
data <- factor(c(1, 2, 3, 4, 5))
```

```
#define new data as original data minus 4th and 5th  
factor levels
```

```
new_data <- data
```

```
#view new data
```

```
new_data
```

```
1 2 3
```

```
Levels: 1 2 3 4 5
```

Although the new data only contains three factors, we can see that it still contains the original five factor levels.

To remove these unused factor levels, we can use the `droplevels()` function:

```
#drop unused factor levels
```

```
new_data <- droplevels(new_data)
```

```
#view data
```

```
new_data
```

```
1 2 3
```

```
Levels: 1 2 3
```

The new data now contains just three factor levels.

Example 2: Drop Unused Factor Levels in a Data Frame

Suppose we create a data frame in which one of the variables is a factor with five levels. Then suppose we define a new data frame that happens to remove two of these factor levels:

```
#create data frame
```

```
df <- data.frame(region=factor(c('A', 'B', 'C', 'D', 'E')),  
sales = c(13, 16, 22, 27, 34))
```

```
#view data frame
```

```
df
```

```
region sales
```

1 A 13

2 B 16

3 C 22

4 D 27

5 E 34

#define new data frame

```
new_df <- subset(df, sales < 25)
```

#view new data frame

```
new_df
```

```
region sales
```

1 A 13

2 B 16

3 C 22

#check levels of region variable

```
levels(new_df$region)
```

```
"A" "B" "C" "D" "E"
```

Although the new data frame contains only three factors in the *region* column, it still contains the original five factor levels. This would create some problems if

we tried to create any plots using this data.

#drop unused factor levels

```
new_df$region <- droplevels(new_df$region)
```

#check levels of region variable

```
levels(new_df$region)
```

```
"A" "B" "C"
```

Now the *region* variable only contains three factor levels.

You can find more R tutorials on .