

How can I use dplyr to convert multiple columns in a data frame to factor variables?

Authored by
stats writer

June 25, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use dplyr to convert multiple columns in a data frame to factor variables?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=151930>

Dplyr is a popular R package that offers a variety of functions for data manipulation. One of its useful features is the ability to easily convert multiple columns in a data frame to factor variables. This can be achieved by using the "mutate_at" function, which allows for the selection of specific columns and the application of a transformation function. In this case, the "factor" function can be used to convert the columns to factors. This process can save time and effort when working with large datasets and can also help in organizing and analyzing categorical data.

Convert Multiple Columns to Factor Using dplyr

You can use the following methods to convert multiple columns to factor using functions from the package:

Method 1: Convert Specific Columns to Factor

```
library(dplyr)
```

```
df %>% mutate_at(c('col1', 'col2'), as.factor)
```

Method 2: Convert All Character Columns to Factor

```
library(dplyr)
```

```
df %>% mutate_if(is.character, as.factor)
```

The following examples show how to use each method in practice.

Example 1: Convert Specific Columns to Factor

Suppose we have the following data frame in R:

```
#create data frame
```

```
df <- data.frame(team=c('A', 'A', 'A', 'B', 'B', 'C', 'C', 'D'),  
position=c('G', 'G', 'F', 'F', 'G', 'G', 'F', 'F'),  
starter=c('Y', 'Y', 'Y', 'N', 'N', 'Y', 'N', 'N'),  
points=c(12, 24, 25, 35, 30, 14, 19, 11))
```

```
#view structure of data frame
```

```
str(df)
```

```
'data.frame': 8 obs. of 4 variables:
```

```
$ team : chr "A" "A" "A" "B" ...
```

```
$ position: chr "G" "G" "F" "F" ...
```

```
$ starter : chr "Y" "Y" "Y" "N" ...
```

```
$ points : num 12 24 25 35 30 14 19 11
```

We can see that the team, position, and starter columns are characters while the points column is numeric.

To convert just the team and position columns to factors, we can use the following syntax:

```
library(dplyr)
```

```
#convert team and position columns to factor
```

```
df <- df %>% mutate_at(c('team', 'position'), as.factor)
```

```
#view structure of updated data frame
```

```
str(df)
```

```
'data.frame': 8 obs. of 4 variables:
```

```
$ team : Factor w/ 4 levels "A","B","C","D": 1 1 1 2 2 3 3 4
```

```
$ position: Factor w/ 2 levels "F","G": 2 2 1 1 2 2 1 1
```

```
$ starter : chr "Y" "Y" "Y" "N" ...
```

```
$ points : num 12 24 25 35 30 14 19 11
```

We can see that the team and position columns are now both factors.

Example 2: Convert All Character Columns to Factor

Suppose we have the following data frame in R:

```
#create data frame
```

```
df <- data.frame(team=c('A', 'A', 'A', 'B', 'B', 'C', 'C', 'D'),
```

```
position=c('G', 'G', 'F', 'F', 'G', 'G', 'F', 'F'),
```

```
starter=c('Y', 'Y', 'Y', 'N', 'N', 'Y', 'N', 'N'),
```

```
points=c(12, 24, 25, 35, 30, 14, 19, 11))
```

```
#view structure of data frame
```

```
str(df)
```

```
'data.frame': 8 obs. of 4 variables:
```

```
$ team : chr "A" "A" "A" "B" ...
```

```
$ position: chr "G" "G" "F" "F" ...
```

```
$ starter : chr "Y" "Y" "Y" "N" ...
```

```
$ points : num 12 24 25 35 30 14 19 11
```

We can see that three of the columns in the data frame are character columns.

```
library(dplyr)
```

```
#convert all character columns to factor
```

```
df <- df %>% mutate_if(is.character, as.factor)
```

```
#view structure of updated data frame
```

```
str(df)
```

```
'data.frame': 8 obs. of 4 variables:
```

```
$ team : Factor w/ 4 levels "A","B","C","D": 1 1 1 2 2 3 3 4
```

```
$ position: Factor w/ 2 levels "F","G": 2 2 1 1 2 2 1 1
```

```
$ starter : Factor w/ 2 levels "N","Y": 2 2 2 1 1 2 1 1
```

\$ points : num 12 24 25 35 30 14 19 11

We can see that all of the character columns are now factors.

Note: Refer to the for a complete explanation of the mutate_at and mutate_if functions.

ARABPSYCHOLOGY.COM