

How can I use dplyr to convert multiple columns to numeric format?

Authored by
stats writer

June 29, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use dplyr to convert multiple columns to numeric format?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=160254>

Dplyr is a popular R package that provides a simple and efficient way to manipulate data frames. It offers a variety of functions that can be used to transform data in different ways. One common task in data analysis is converting columns to numeric format. This can be easily achieved using dplyr's ``mutate()`` function, which allows for the creation of new columns or the modification of existing ones. By specifying the ``as.numeric()`` function within ``mutate()``, multiple columns can be converted to numeric format simultaneously. This allows for efficient and streamlined data manipulation, making dplyr a valuable tool for data analysis.

Convert Multiple Columns to Numeric Using dplyr

You can use the following methods to convert multiple columns to numeric using the package:

Method 1: Convert Specific Columns to Numeric

```
library(dplyr)
```

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

Method 2: Convert All Character Columns to Numeric

```
library(dplyr)
```

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

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

Example 1: Convert Specific Columns to Numeric

Suppose we have the following data frame in R:

```
#create data frame
```

```
df <- data.frame(team=c('A', 'B', 'C', 'D', 'E'),  
position=c('G', 'G', 'G', 'F', 'F'),  
assists=c('33', '28', '31', '39', '34'),  
rebounds=c('30', '28', '24', '24', '28'))
```

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

```
str(df)
```

```
'data.frame': 5 obs. of 4 variables:
```

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

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

```
$ assists : chr "33" "28" "31" "39" ...
```

```
$ rebounds: chr "30" "28" "24" "24" ...
```

We can see that every column in the data frame is currently a character.

To convert just the assists and rebounds columns to numeric, we can use the following syntax:

```
library(dplyr)
```

```
#convert assists and rebounds columns to numeric  
df <- df %>% mutate_at(c('assists', 'rebounds'),  
as.numeric)
```

```
#view structure of updated data frame  
str(df)
```

```
'data.frame': 5 obs. of 4 variables:
```

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

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

```
$ assists : num 33 28 31 39 34
```

```
$ rebounds: num 30 28 24 24 28
```

We can see that the assists and rebounds columns are now both numeric.

Example 2: Convert All Character Columns to Numeric

Suppose we have the following data frame in R:

```
#create data frame
```

```
df <- data.frame(ranking=factor(c(1, 4, 3, 2, 5)),
```

```
assists=c('12', '10', '8', '11', '15'),
```

```
points=c('33', '28', '31', '39', '34'),
```

```
rebounds=c('30', '28', '24', '24', '28'))
```

```
#view structure of data frame  
str(df)
```

```
'data.frame': 5 obs. of 4 variables:  
 $ ranking : Factor w/ 5 levels "1","2","3","4",...: 1 4 3 2 5  
 $ assists : chr "12" "10" "8" "11" ...  
 $ points : chr "33" "28" "31" "39" ...  
 $ rebounds: chr "30" "28" "24" "24" ...
```

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

```
library(dplyr)
```

```
#convert all character columns to numeric  
df <- df %>% mutate_if(is.character, as.numeric)
```

```
#view structure of updated data frame  
str(df)
```

```
'data.frame': 5 obs. of 4 variables:  
 $ ranking : Factor w/ 5 levels "1","2","3","4",...: 1 4 3 2 5  
 $ assists : num 12 10 8 11 15  
 $ points : num 33 28 31 39 34  
 $ rebounds: num 30 28 24 24 28
```

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

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

Additional Resources

ARABPSYCHOLOGY.COM