

How to Convert Multiple Columns to Numeric Using dplyr

Authored by
stats writer

November 29, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Convert Multiple Columns to Numeric Using dplyr*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=101828>

Using the dplyr package, it is possible to convert multiple columns to numeric by utilizing the `mutate()` and `as.numeric()` functions. The `mutate()` function allows for the modification of existing columns in a data frame, and the `as.numeric()` function converts the columns to a numeric data type. This can be done using a single line of code for each column, or multiple columns can be converted at the same time using the `select()` and `mutate_at()` functions. This is a useful tool for quickly changing the data type of columns in a data frame.

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.