

# How can I drop multiple columns using dplyr in R? Can you provide some examples?

Authored by  
**stats writer**

June 26, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I drop multiple columns using dplyr in R? Can you provide some examples?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=153625>

Dplyr is a popular package in R that provides a set of tools for data manipulation. One of its useful functions is the ability to drop multiple columns from a data frame. This can be achieved using the "select" function, where the columns to be dropped are specified using the "!" operator. For example, the code "select(data, !col1, !col2)" will drop columns "col1" and "col2" from the data frame "data". This function allows for efficient and easy removal of unwanted columns, providing a streamlined approach to data cleaning and analysis. Below are some examples demonstrating how to use the "select" function to drop multiple columns in R using dplyr.

## Drop Multiple Columns Using dplyr (With Examples)

You can use one of the following methods to drop multiple columns from a data frame in R using the dplyr package:

### 1. Drop Multiple Columns by Name

```
df_new <- df %>% select(-c(col2, col4))
```

### 2. Drop All Columns in Range

```
df_new <- df %>% select(-c(col2:col4))
```

The following examples show how to use each of these methods in practice with the following data frame:

```
#create data frame
```

```
df = data.frame(rating = c(90, 85, 82, 88, 94, 90, 76, 75, 87, 86),
```

```
points=c(25, 20, 14, 16, 27, 20, 12, 15, 14, 19),  
assists=c(5, 7, 7, 8, 5, 7, 6, 9, 9, 5),  
rebounds=c(11, 8, 10, 6, 6, 9, 6, 10, 10, 7))
```

```
#view data frame
```

```
df
```

```
rating points assists rebounds
```

```
1 90 25 5 11
```

```
2 85 20 7 8
```

```
3 82 14 7 10
```

```
4 88 16 8 6
```

```
5 94 27 5 6
```

```
6 90 20 7 9
```

```
7 76 12 6 6
```

```
8 75 15 9 10
```

```
9 87 14 9 10
```

```
10 86 19 5 7
```

**Example 1: Drop Multiple Columns by Name**

The following code shows how to drop the columns named points and rebounds from the data frame:

```
library(dplyr)
```

```
#drop points and rebounds columns
```

```
df_new <- df %>% select(-c(points, rebounds))
```

```
#view new data frame
```

```
new_df
```

```
rating assists
```

```
1 90 5
```

```
2 85 7
```

```
3 82 7
```

```
4 88 8
```

```
5 94 5
```

```
6 90 7
```

```
7 76 6
```

```
8 75 9
```

```
9 87 9
```

```
10 86 5
```

**Notice that the columns named points and rebounds have both been dropped from the new data frame.**

**Example 2: Drop All Columns in Range**

**The following code shows how to drop all columns in the range between the points and rebounds columns:**

```
library(dplyr)
```

```
#drop all columns between points and rebounds
```

```
df_new <- df %>% select(-c(points:rebounds))
```

```
#view new data frame
```

```
new_df
```

```
rating
```

```
1 90
```

```
2 85
```

```
3 82
```

```
4 88
```

```
5 94
```

```
6 90
```

```
7 76
```

```
8 75
```

```
9 87
```

```
10 86
```

Notice that all columns between points and rebounds have been dropped from the new data frame.

**Note:** The MASS package in R also has a `select()` function. If this package is also loaded, you should use

**dplyr::select()** so that R knows to use the **select()** function from the **dplyr** package.

ARABPSYCHOLOGY.COM