

How can I use the `select_if` function in dplyr to filter a data frame based on multiple conditions?

Authored by
stats writer

June 27, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use the `select_if` function in dplyr to filter a data frame based on multiple conditions?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=155518>

The `select_if` function in the `dplyr` package allows users to filter a data frame by specifying multiple conditions. This function is useful for selecting specific columns from a data frame based on certain criteria, such as data type or variable values. By using the `select_if` function, users can efficiently extract the desired data from a data frame and create a new, filtered data frame that meets their specific needs. This function is particularly helpful in data analysis and manipulation tasks, as it allows for quick and targeted data selection.

Use `select_if` with Multiple Conditions in `dplyr`

You can use the following basic syntax with the `select_if()` function from the package to select columns in a data frame that meet one of several conditions:

```
df %>% select_if(function(x) condition1 | condition2)
```

The following examples show how to use this syntax in practice.

Example 1: Use `select_if()` with Class Types

The following code shows how to use the `select_if()` function to select the columns in a data frame that have a class type of character or numeric:

```
library(dplyr)
```

```
#create data frame
```

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

```
conference=as.factor(c('W', 'W', 'W', 'E', 'E')),  
points_for=c(99, 90, 86, 88, 95),  
points_against=c(91, 80, 88, 86, 93))
```

```
#select all character and numeric columns
```

```
df %>% select_if(function(x) is.character(x) |  
is.numeric(x))
```

```
team points_for points_against
```

```
1 A 99 91
```

```
2 B 90 80
```

```
3 C 86 88
```

```
4 D 88 86
```

```
5 E 95 93
```

Notice that the one character column (team) and the two numeric columns (points_for and points_against) are returned while the factor column (conference) is not returned.

Example 2: Use select_if() with Class Types and Column Names

The following code shows how to use the select_if() function to select the columns in a data frame that have a class type of factor or have a column name of

points_for:

library(dplyr)

#create data frame

```
df <- data.frame(team=c('A', 'B', 'C', 'D', 'E'),  
conference=as.factor(c('W', 'W', 'W', 'E', 'E')),  
points_for=c(99, 90, 86, 88, 95),  
points_against=c(91, 80, 88, 86, 93))
```

#select all factor columns and 'points_for' column

```
df %>% select_if(function(x) is.factor(x) | all(x ==  
.$points_for))
```

conference points_for

1 W 99

2 W 90

3 W 86

4 E 88

5 E 95

Notice that the one factor column and the one column titled points_for are returned.

Note: The | symbol is the "OR" logical operator in R.

Feel free to use as many `|` symbols as you'd like to select columns using more than two conditions.

The following tutorials explain how to use other common functions in `dplyr`:

ARABPSYCHOLOGY.COM