

# How can a conditional filter be applied in dplyr?

Authored by  
**stats writer**

June 25, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can a conditional filter be applied in dplyr?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=151830>

A conditional filter can be applied in dplyr by using the "filter" function and specifying the condition that needs to be met. This allows for the selection of specific rows in a dataset based on a given criteria. The condition can be any logical statement, such as greater than, less than, or equal to a certain value. By using a conditional filter, data can be easily subsetted and manipulated in a more efficient and organized manner. This feature in dplyr allows for a streamlined data analysis process, making it a valuable tool for data manipulation and management.

## Use a Conditional Filter in dplyr

You can use the following basic syntax to apply a conditional filter on a data frame using functions from the dplyr package in R:

```
library(dplyr)
```

```
#filter data frame where points is greater than some value (based on team)
```

```
df %>%
```

```
filter(case_when(team=='A' ~ points > 15,
```

```
team=='B' ~ points > 20,
```

```
TRUE ~ points > 30))
```

This particular example filters the rows in a data frame where the value in the points column is greater than a certain value, *conditional* on the value in the team column.

The following example shows how to use this syntax in practice.

Example: How to Use Conditional Filter in dplyr

Suppose we have the following data frame in R that contains information about various basketball players:

```
#create data frame
```

```
df <- data.frame(team=c('A', 'A', 'A', 'B', 'B', 'B', 'C', 'C',  
'C'),  
points=c(10, 12, 17, 18, 24, 29, 29, 34, 35))
```

```
#view data frame
```

```
df
```

```
team points
```

```
1 A 10
```

```
2 A 12
```

```
3 A 17
```

```
4 B 18
```

```
5 B 24
```

```
6 B 29
```

```
7 C 29
```

```
8 C 34
```

## 9 C 35

Now suppose we would like to apply the following conditional filter:

Only keep rows for players on team A where points is greater than 15  
Only keep rows for players on team B where points is greater than 20  
Only keep rows for players on team C where points is greater than 30

We can use the `filter()` and `case_when()` functions from the `dplyr` package to apply this conditional filter on the data frame:

```
library(dplyr)
```

```
#filter data frame where points is greater than some value (based on team)
```

```
df %>%
```

```
filter(case_when(team=='A' ~ points > 15,
```

```
team=='B' ~ points > 20,
```

```
TRUE ~ points > 30))
```

```
team points
```

```
1 A 17
```

**2 B 24**

**3 B 29**

**4 C 34**

**5 C 35**

The rows in the data frame are now filtered where the value in the points column is greater than a certain value, *conditional* on the value in the team column.

**Note #1:** In the `case_when()` function, we use `TRUE` in the last argument to represent any values in the team column that are not equal to 'A' or 'B'.

**Note #2:** You can find the complete documentation for the dplyr `case_when()` function .

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