

# How can I group and filter data using dplyr in R?

Authored by  
**stats writer**

June 26, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I group and filter data using dplyr in R?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=153939>

Dplyr is a powerful R package that allows for efficient and easy manipulation of data frames. One of its main functions is the ability to group and filter data, which is particularly useful for data analysis and visualization. With dplyr, users can group data by one or more variables and then apply various functions, such as summarizing or mutating, to each group separately. Additionally, dplyr allows for the filtering of data based on specific conditions, making it easy to extract subsets of data for further analysis. Overall, utilizing dplyr in R provides a streamlined and flexible approach to grouping and filtering data, making it an essential tool for data manipulation and exploration.

## Group By and Filter Data Using dplyr

You can use the following basic syntax to group by and filter data using the dplyr package in R:

```
df %>%  
group_by(team) %>%  
filter(any(points == 10))
```

This particular syntax groups a data frame by the column called team and filters for only the groups where at least one value in the points column is equal to 10.

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

**Example: Group By and Filter Data Using 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, 15, 8, 4, 10, 10, 12, 12, 7))
```

**#view data frame**

**df**

**team points**

**1 A 10**

**2 A 15**

**3 A 8**

**4 B 4**

**5 B 10**

**6 B 10**

**7 C 12**

**8 C 12**

**9 C 7**

**We can use the following code to group the data frame by the value in the team column and then filter out all groups that do not have at least one value in the points column equal to 10:**

```
library(dplyr)
```

```
#group by team and filter out teams where no points  
value is equal to 10
```

```
df %>%
```

```
group_by(team) %>%
```

```
filter(any(points ==10))
```

```
# A tibble: 6 x 2
```

```
# Groups: team
```

```
team points
```

```
1 A 10
```

```
2 A 15
```

```
3 A 8
```

```
4 B 4
```

```
5 B 10
```

```
6 B 10
```

Notice that all rows where the team is equal to C" are filtered out because there is no value in the points column for team C "equal to 10.

Note that this is just one example of a filter that we could apply.

For example, we could apply another filter where we filter for teams where at least one value in the points column is greater than 13:

```
library(dplyr)
```

```
#group by team and filter out teams where no points  
value is greater than 13
```

```
df %>%
```

```
group_by(team) %>%
```

```
filter(any(points >13))
```

```
# A tibble: 3 x 2
```

```
# Groups: team
```

```
team points
```

```
1 A 10
```

```
2 A 15
```

```
3 A 8
```

Notice that only the rows where the team is equal to "A" are kept since this is the only team with at least one points value greater than 13.

**Note:** You can find the complete documentation for the

## **filter function in dplyr .**

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

ARABPSYCHOLOGY.COM