

How can I select the top N values by group in R?

Authored by
stats writer

June 27, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I select the top N values by group in R?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=155645>

In R, the top N values by group can be selected using the "group_by" and "top_n" functions in the "dplyr" package. The "group_by" function allows for grouping the data by a specific variable, while the "top_n" function can be used to select the top N values within each group. This approach is useful for identifying the highest or lowest values within different categories or groups in a dataset. By specifying the desired number of values to be selected, the top N values can be easily determined and extracted from the dataset, providing a useful tool for data analysis and decision making.

Select Top N Values by Group in R

You can use one of the following methods to select the top N values by group in R:

Method 1: Select Top N Values by Group (Ignore Ties)

```
library(dplyr)
```

```
#select top 5 values by group
```

```
df %>%
```

```
  arrange(desc(values_column)) %>%
```

```
  group_by(group_column) %>%
```

```
  slice(1:5)
```

Method 2: Select Top N Values by Group (Include Ties)

```
library(dplyr)
```

```
#select top 5 values by group
```

```
df %>%  
group_by(group_column) %>%  
top_n(5, values_column)
```

The following examples show how to use each method with the following data frame in R:

```
#create data frame
```

```
df <- data.frame(team=c('A', 'A', 'A', 'A', 'B', 'B', 'B', 'B'),  
points=c(19, 22, 15, NA, 14, 25, 25, 25),  
rebounds=c(10, 6, 3, 7, 11, 13, 9, 12))
```

```
#view data frame
```

```
df
```

```
team points rebounds
```

```
1 A 19 10
```

```
2 A 22 6
```

```
3 A 15 3
```

```
4 A NA 7
```

```
5 B 14 11
```

```
6 B 25 13
```

```
7 B 25 9
```

```
8 B 25 12
```

Example 1: Select Top N Values by Group (Ignore Ties)

The following code shows how to select the top 2 rows with the highest points values, grouped by team:

```
library(dplyr)
```

```
#select top 2 rows with highest points values, grouped  
by team
```

```
df %>%
```

```
arrange(desc(points)) %>%
```

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

```
slice(1:2)
```

```
# A tibble: 4 x 3
```

```
# Groups: team
```

```
team points rebounds
```

```
1 A 22 6
```

```
2 A 19 10
```

```
3 B 25 13
```

```
4 B 25 9
```

The output contains the two rows with the highest points values for each team.

Note that for team B, there were actually three rows that were tied for highest points value (25) but only two rows are returned in the output.

This method simply ignores ties.

Example 2: Select Top N Values by Group (Include Ties)

The following code shows how to select the top 2 rows with the highest points values, grouped by team:

```
library(dplyr)
```

```
#select top 2 rows with highest points values, grouped  
by team
```

```
df %>%
```

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

```
top_n(2, points)
```

```
# A tibble: 5 x 3
```

```
# Groups: team
```

```
team points rebounds
```

```
1 A 19 10
```

```
2 A 22 6
```

```
3 B 25 13
```

4 B 25 9

5 B 25 12

The output contains the two rows with the highest points values for each team.

ARABPSYCHOLOGY.COM