

How can columns be reordered in R?

Authored by
stats writer

April 18, 2024

RECOMMENDED CITATION

stats writer (2024). *How can columns be reordered in R?*. PSYCHOLOGICAL SCALES.
Retrieved from <https://scales.arabpsychology.com/?p=136648>

In R, columns can be reordered by using the "select" function from the dplyr package. This function allows for the selection and rearrangement of columns in a dataframe by specifying the desired order of columns using their names or indices. Additionally, the "select" function also allows for the exclusion of certain columns from the rearrangement. By utilizing this function, users can easily and efficiently reorder columns in R to fit their specific needs and preferences.

Reorder Columns in R

Often you may want to reorder columns in a data frame in R.

Fortunately this is easy to do using the **select()** function from the **dplyr** package.

```
library(dplyr)
```

This tutorial shows several examples of how to use this function in practice using the following data frame:

```
#create data frame
```

```
df <- data.frame(player = c('a', 'b', 'c', 'd', 'e'),  
position = c('G', 'F', 'F', 'G', 'G'),  
points = c(12, 15, 19, 22, 32),  
rebounds = c(5, 7, 7, 12, 11))
```

```
#view data frame
```

```
df
```

player position points rebounds

1 a G 12 5

2 b F 15 7

3 c F 19 7

4 d G 22 12

5 e G 32 11

Example 1: Move a Column to the First Position

The following code shows how to move a specific column in a data frame to the first position:

```
#move column 'points' to first position  
df %>% select(points, everything())
```

points player position rebounds

1 12 a G 5

2 15 b F 7

3 19 c F 7

4 22 d G 12

5 32 e G 11

This code tells dplyr to select the points column first, then include every other column after points.

Example 2: Move a Column to the Last Position

The following code shows how to move a specific column in a data frame to the last position:

```
#move column 'points' to last position  
df %>% select(-points, points)
```

```
player position rebounds points
```

```
1 a G 5 12
```

```
2 b F 7 15
```

```
3 c F 7 19
```

```
4 d G 12 22
```

```
5 e G 11 32
```

This code tells dplyr to select all columns *except* the points column, then to select the points column again. This has the effect of moving the points column to the last position in the data frame.

Example 3: Reorder Multiple Columns

The following code shows how to reorder several columns at once in a specific order:

```
#change all column names to uppercase
```

```
df %>% select(rebounds, position, points, player)
```

```
rebounds position points player
```

```
1 5 G 12 a
```

```
2 7 F 15 b
```

```
3 7 F 19 c
```

```
4 12 G 22 d
```

```
5 11 G 32 e
```

Example 4: Reorder Columns Alphabetically

```
#order columns alphabetically
```

```
df %>% select(order(colnames(.)))
```

```
player points position rebounds
```

```
1 a 12 G 5
```

```
2 b 15 F 7
```

```
3 c 19 F 7
```

```
4 d 22 G 12
```

```
5 e 32 G 11
```

Example 5: Reverse Column Order

The following code shows how to reverse the column order in a data frame:

```
#reverse column order
```

```
df %>% select(rebounds:player, everything())
```

```
rebounds points position player
```

```
1 5 12 G a
```

```
2 7 15 F b
```

```
3 7 19 F c
```

```
4 12 22 G d
```

```
5 11 32 G e
```

Note: You can find the complete documentation for the `select()` function [here](#).

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