

How can I use the relocate() function in dplyr, and what are some examples of its usage?

Authored by
stats writer

July 1, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use the relocate() function in dplyr, and what are some examples of its usage?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=164816>

The relocate() function in dplyr is a powerful tool for rearranging columns in a data frame. It allows users to easily move columns to a desired position, rename columns, or even create new columns. This function can be used to improve the organization and readability of data, as well as to prepare it for further analysis. To use the relocate() function, simply specify the columns to be moved and their desired location using the before or after argument. Some examples of its usage include reordering columns to match a desired format, moving frequently used columns to the beginning of a data frame, or creating a new column based on the values of existing ones. Overall, the relocate() function provides a flexible and efficient way to manipulate the structure of a data frame in dplyr.

Use the relocate() Function in dplyr (With Examples)

You can use the function from the **dplyr** package in R to change the column positions in a data frame.

You can use the following methods to change the column positions:

Method 1: Move One Column to Front

```
#move 'x' column to front  
df %>% relocate(x)
```

Method 2: Move Several Columns to Front

```
#move 'x' and 'y' columns to front  
df %>% relocate(x, y)
```

Method 3: Move Column to Position After Another

Column

```
#move 'x' column to position after 'y' column  
df %>% relocate(x, .after=y)
```

Method 4: Move Column to Position Before Another Column

```
#move 'x' column to position before 'y' column  
df %>% relocate(x, .before=y)
```

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

```
#create dataset  
df <- data.frame(team=c('A', 'A', 'A', 'B', 'B', 'C', 'C'),  
points=c(1, 2, 3, 4, 5, 6, 7),  
assists=c(1, 5, 2, 3, 2, 2, 0),  
rebounds=c(6, 6, 10, 12, 8, 8, 3))
```

```
#view dataset  
df
```

```
team points assists rebounds  
1 A 1 1 6
```

```
2 A 2 5 6
3 A 3 2 10
4 B 4 3 12
5 B 5 2 8
6 C 6 2 8
7 C 7 0 3
```

Example 1: Move One Column to Front

The following code shows how to use the relocate() function to move one column to the front:

```
#move 'assists' column to front
df %>% relocate(assists)
```

```
assists team points rebounds
```

```
1 1 A 1 6
2 5 A 2 6
3 2 A 3 10
4 3 B 4 12
5 2 B 5 8
6 2 C 6 8
7 0 C 7 3
```

Example 2: Move Several Columns to Front

The following code shows how to use the relocate() function to move multiple columns to the front:

```
#move 'points' and 'assists' to front  
df %>% relocate(points, assists)
```

```
points assists team rebounds
```

```
1 1 1 A 6
```

```
2 2 5 A 6
```

```
3 3 2 A 10
```

```
4 4 3 B 12
```

```
5 5 2 B 8
```

```
6 6 2 C 8
```

```
7 7 0 C 3
```

Example 3: Move Column to Position After Another Column

The following code shows how to use the relocate() function to move one column to a specific position after another column:

```
#move 'team' column to after 'assists' column  
df %>% relocate(team, .after=assists)
```

```
points assists team rebounds
```

```
1 1 1 A 6
```

```
2 2 5 A 6
3 3 2 A 10
4 4 3 B 12
5 5 2 B 8
6 6 2 C 8
7 7 0 C 3
```

Example 4: Move Column to Position Before Another Column

The following code shows how to use the `relocate()` function to move one column to a specific position before another column:

```
#move 'team' column to before 'rebounds' column
df %>% relocate(team, .before=rebounds)
```

```
points assists team rebounds
```

```
1 1 1 A 6
2 2 5 A 6
3 3 2 A 10
4 4 3 B 12
5 5 2 B 8
6 6 2 C 8
7 7 0 C 3
```

Additional Resources

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

ARABPSYCHOLOGY.COM