How can rows be arranged in R?

Authored by

stats writer

April 18, 2024

Rows in R can be arranged in various ways depending on the specific needs of the user. One common method is to use the "order" function, which allows the user to specify the columns by which the rows should be arranged. Another method is to use the "sort" function, which arranges the rows in ascending or descending order based on the values in a specific column. Additionally, rows can also be rearranged using the "dplyr" package, which offers a variety of functions such as "arrange" and "slice" to manipulate the order of rows. Overall, the flexibility and versatility of R allows for efficient and customizable arrangements of rows in datasets.

Arrange Rows in R

Often you may be interested in arranging the rows of a data frame in R in a specific order. Fortunately this is easy to do using the arrange() function from the dplyr library.

This tutorial explains 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', 'F', 'G'),
points = c(12, 14, 14, 15, 20, 18, 29),
assists = c(3, 5, 7, 8, 14, NA, 9))

#view data frame
df

player points assists
1 A 12 3
```

2 B 14 5

3 C 14 7

4 D 15 8

5 E 20 14

6 F 18 NA

7 G 29 9

Example 1: Arrange by One Column

The following code shows how to arrange the data frame in ascending order based on the values in the 'points' column:

```
library(dplyr)

df %>% arrange(points)
```

player points assists

1 A 12 3

2 B 14 5

3 C 14 7

4 D 15 8

5 F 18 NA

6 E 20 14

7 G 29 9

To sort in descending order, we can use the desc() function:

df %>% arrange(desc(points))

player points assists
1 G 29 9
2 E 20 14
3 F 18 NA
4 D 15 8
5 B 14 5
6 C 14 5
7 A 12 3

Note that NA's will be sorted to the end, whether or not you sort ascending or decsending:

df %>% arrange(assists)

player points assists
1 A 12 3
2 B 14 5
3 C 14 7
4 D 15 8
5 G 29 9

6 E 20 14

7 F 18 NA

df %>% arrange(desc(assists))

player points assists

1 E 20 14

2 G 29 9

3 D 15 8

4 C 14 7

5 B 14 5

6 A 12 3

7 F 18 NA

Example 2: Arrange by Multiple Columns

To arrange the rows by multiple columns, we can simply provide more column names as arguments:

#sort by points, then assists

df %>% arrange(points, assists)

player points assists

1 A 12 3

2 B 14 5

3 C 14 7

4 D 15 8

5 F 18 NA

6 E 20 14

7 G 29 9

We can also arrange the rows by one column ascending and another descending:

```
#sort by points ascending, then assists descending
df %>% arrange(points, desc(assists))
```

```
player points assists
1 A 12 3
2 C 14 7
3 B 14 5
4 D 15 8
5 F 18 NA
6 E 20 14
7 G 29 9
```


Example 3: Arrange Rows in a Custom order

Occasionally you may also want to sort the rows in a custom order. You can easily do this by using a factor with specific levels:

```
#sort by player with custom order
df %>% arrange(factor(player, levels = c('D', 'C', 'A', 'B', 'E', 'F', 'G')))
```

```
player points assists
1 D 15 8
2 C 14 7
3 A 12 3
4 B 14 5
5 E 20 14
6 F 18 NA
7 G 29 9
```

You can find the complete documentation for the arrange() function here.