

How can I delete rows in R?

Authored by
stats writer

June 24, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I delete rows in R?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=149524>

Deleting rows in R is a process of removing specific rows from a data frame. This can be done by using the "subset" function and specifying the row numbers or conditions to be deleted. Alternatively, the "filter" function can be used to select and keep only the desired rows. Deleting rows in R can help in data cleaning and manipulation, allowing for a more accurate and concise analysis of the data.

R provides a **subset()** function to delete or drop a single row/multiple rows from the DataFrame (data.frame), you can also use the notation and **-c()** to delete the rows. In this article, we will discuss several ways to delete rows from the DataFrame. We can delete rows from the data frame in the following ways:

Delete Single/Multiple Rows from a DataFrame by row index
Delete Multiple Rows from a DataFrame by row name
Delete Rows by Condition

Note that R doesn't have a function that deletes the Rows from the R DataFrame however, we should use a subsetting way to drop rows. For example, to delete the second and third row in R, You can use `-c(1, 3)`, and it will return the DataFrame without the first and third rows.

1. Quick Examples of Delete Rows

Below are some quick examples of deleting or dropping rows from the R DataFrame.

```
# Below are the quick examples

# Exemple 1: delete 4th row
df2 <- df

# Exemple 2: delete 4th,5th and 1st rows
df2 <- df

# Exemple 3: delete rows by range
df2 <- df

# Exemple 4: delete rows by name
df2 <- df

# Exemple 5: Remove rows with column id less than equal 2
df2 <- subset(df, id > 2 )
```

Let's create a DataFrame with 5 rows and 3 columns.

```
# Create dataframe with 5 rows and 3 columns
df = data.frame(id=c(1,2,3,4,5),
name=c('sravan','srinu','chrisa','shivgami','jim'),
gender=c('m','m','f','f','m'))
```

```
# Display dataframe
df
```

Yields below output.

	id	name	gender
1	1	sravan	m
2	2	srinu	m
3	3	chrisa	f
4	4	shivgami	f
5	5	jim	m

2. Delete Rows by Index from the R DataFrame

To delete single row/multiple rows from R DataFrame(data.frame) you can use notation with the negative row index. Here, we delete only a single row from the R DataFrame using the row index number. The row number starts with 1.

Syntax:

```
# Syntax
df
```

Where df is the DataFrame from where you want to delete the row? Note that this doesn't remove from the existing DataFrame so you need to assign it to another DataFrame to get the desired result.

Let's remove the specified row by using its corresponding row_index.

```
# Remove specified row by index
df2 <- df
df2
```

Output:

```

      id  name gender
1  1 sravan      m
2  2  srinu      m
3  3 chrisa      f
5  5    jim      m

```

In the above code, we have deleted the 4th row and returned a new DataFrame with the remaining rows.

3. Delete Multiple Rows from the R DataFrame

You can also delete multiple rows by row index. For that, you can specify indexes of rows that you want to delete from DataFrame using a `c()` function. This function creates a vector containing the row numbers you want to exclude from the data frame `df`. The `-` sign before `c(...)` is used to remove those specific rows from the data frame.

Syntax:

```

# Syntax
df

```

Example:

In this example, we will delete multiple rows at a time.

```

#delete 4th,5th and 1st rows
df2 <- df[-c(1,4,5)]
df2

```

Output:

```

# Output
id name gender
2 2 srinu m
3 3 chrisa f

```

In the above code, we have deleted the 4th, 5th, and 1st rows at a time. so the output will be the remaining rows present in the data frame.

4. Delete Rows by Range

You can also delete rows by range using bracket notation. The following example drops all rows from row number 1 and 3.

```
# delete rows by range
df2 <- df
df2
```

Output:

```
# Output
id name gender
4 4 shivgami f
5 5 jim m
```

5. Delete Rows by Row Name

In our DataFrame we don't have custom row names instead row names are the same as row index numbers due to default behavior. If you have row names then you may need to delete rows by name. The below example demonstrates how you can do this.

```
# delete rows by name
df2 <- df
df2
```

Output:

```
# Output
id name gender
3 3 chrisa f
4 4 shivgami f
5 5 jim m
```

6. Delete Rows by Condition

In this scenario, we are selecting only a particular row from a data frame based on the condition, this way, we can ignore the rows from the data frame that fails the condition using the `subset()` method. This method takes two parameters data frame object and the condition.

Syntax:

```
# Syntax
subset(df,condition )
```

In this example, we will apply different conditions.

```
# Remove rows with column id less than equal 2
df2 <- subset(df,id > 2 )
df2
```

```
# Remove rows where gender not equal to 'm'
df2 <- subset(df,gender=='m' )
df2
```

Output:

```
# Output 1
id name gender
3 3 chrisa f
4 4 shivgami f
5 5 jim m
```

```
# Output 2
id name gender
1 1 sravan m
2 2 srinu m
5 5 jim m
```

In the above code

Frequently Asked Questions on Delete Rows in R

How do I delete specific rows from a data frame in R?

To delete specific rows from a data frame, you can use the negative index notation. For example, if you want to delete rows 3 and 5, `df <- df`

How can I remove rows with missing values in R?

To remove rows with missing values, you can use the `na.omit()` function or the `complete.cases()` function. For example, `df <- df`

What is the difference between `subset()` and direct subsetting for deleting rows?

The `subset()` function allows you to specify conditions for row removal more conveniently. For example, `df <- subset(df, condition)`

How do I delete duplicate rows from a data frame?

You can use the `duplicated()` function to identify the duplicate rows and remove them. For instance, `df <- df`

How to delete rows with specific values in a column?

You can use logical subsetting to delete rows with specific values in a column. For example, `df <- df`

7. Conclusion

In this article, we have seen three methods to delete or drop single and multiple rows from the DataFrame in R language. Based on the requirement of your application, you can use any of the above-discussed methods.

Related Articles

References