

How can I remove rows in R, and what are some examples of doing so?

Authored by
stats writer

April 30, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I remove rows in R, and what are some examples of doing so?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=141534>

Removing rows in R refers to the process of eliminating specific rows of data from a data frame. This can be done using the "subset" or "filter" functions, where you can specify the condition or criteria for removing the rows. For example, you can remove rows with missing values, or rows that do not meet certain criteria, such as a specific value or range. Another way of removing rows is by using the "drop_na" function, which removes rows with missing values. This can be useful in data cleaning and manipulation tasks, as it allows for a more streamlined and accurate analysis of the remaining data. Overall, by utilizing these various methods, you can easily remove unwanted rows in R and customize your data frame for further analysis.

Remove Rows in R (With Examples)

You can use the following syntax to remove specific row numbers in R:

```
#remove 4th row  
new_df <- df
```

```
#remove 2nd through 4th row  
new_df <- df
```

```
#remove 1st, 2nd, and 4th row  
new_df <- df
```

You can use the following syntax to remove rows that don't meet specific conditions:

```
#only keep rows where col1 value is less than 10 and  
col2 value is less than 6
```

```
new_df <- subset(df, col1<10 & col2<6)
```

And you can use the following syntax to remove rows with an NA value in any column:

```
#remove rows with NA value in any column  
new_df <- na.omit(df)
```

The following examples show how to use each of these functions in practice.

Example 1: Remove Rows by Number

The following code shows how to remove rows by specific row numbers in R:

```
#create data frame  
df <- data.frame(player=c('A', 'B', 'C', 'D', 'E'),  
pts=c(17, 12, 8, 9, 25),  
rebs=c(3, 3, 6, 5, 8),  
blocks=c(1, 1, 2, 4, NA))
```

```
#view data frame
```

```
df
```

```
player pts rebs blocks
```

```
1 A 17 3 1
```

```
2 B 12 3 1
3 C 8 6 2
4 D 9 5 4
5 E 25 8 NA
```

```
#remove 4th row
df
```

```
player pts rebs blocks
```

```
1 A 17 3 1
2 B 12 3 1
3 C 8 6 2
5 E 25 8 NA
```

```
#remove 2nd through 4th row
df
```

```
player pts rebs blocks
```

```
1 A 17 3 1
5 E 25 8 NA
```

```
#remove 1st, 2nd, and 4th row
df
```

```
player pts rebs blocks
```

```
3 C 8 6 2
```

5 E 25 8 NA

Example 2: Remove Rows by Condition

The following code shows how to remove rows that don't meet a specific condition:

```
#create data frame
```

```
df <- data.frame(player=c('A', 'B', 'C', 'D', 'E'),  
pts=c(17, 12, 8, 9, 25),  
rebs=c(3, 3, 6, 5, 8),  
blocks=c(1, 1, 2, 4, NA))
```

```
#view data frame
```

```
df
```

```
player pts rebs blocks
```

```
1 A 17 3 1
```

```
2 B 12 3 1
```

```
3 C 8 6 2
```

```
4 D 9 5 4
```

```
5 E 25 8 NA
```

```
#only keep rows where pts is less than 10 and rebs is  
less than 6
```

```
subset(df, pts<10 & rebs<6)
```

```
player pts rebs blocks  
4 D 9 5 4
```

Example 3: Remove Rows with NA Values

The following code shows how to remove rows with a NA value in any row:

```
#create data frame  
df <- data.frame(player=c('A', 'B', 'C', 'D', 'E'),  
pts=c(17, 12, 8, 9, 25),  
rebs=c(3, 3, 6, 5, 8),  
blocks=c(1, 1, 2, 4, NA))
```

```
#view data frame  
df  
player pts rebs blocks  
1 A 17 3 1  
2 B 12 3 1  
3 C 8 6 2  
4 D 9 5 4  
5 E 25 8 NA
```

```
#remove rows with NA value in any row:  
na.omit(df)
```

player pts rebs blocks

1 A 17 3 1

2 B 12 3 1

3 C 8 6 2

4 D 9 5 4

ARABPSYCHOLOGY.COM