

How do you remove columns in R (with examples)?

Authored by
stats writer

April 18, 2024

RECOMMENDED CITATION

stats writer (2024). *How do you remove columns in R (with examples)?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=136650>

Removing columns in R refers to the process of eliminating specific columns from a data frame or matrix. This can be done using the "subset" function or by using the index notation. The "subset" function allows you to specify the columns to be removed, while the index notation uses the column numbers to identify which columns to delete. For example, to remove the first and third columns from a data frame named "df", you can use the code "df <- subset(df, select = -c(1,3))" or "df <- df[, 2]". This process is useful for cleaning and organizing data, as well as for performing statistical analyses on specific subsets of data.

Remove Columns in R (With Examples)

Often you may want to remove one or more columns from 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: Remove Columns by Name

The following code shows how to remove columns from a data frame by name:

```
#remove column named 'points'  
df %>% select(-points)
```

player position rebounds

1 a G 5

2 b F 7

3 c F 7

4 d G 12

5 e G 11

Example 2: Remove Columns in List

The following code shows how to remove columns from a data frame that are in a specific list:

```
#remove columns named 'points' or 'rebounds'  
df %>% select(-one_of('points', 'rebounds'))
```

player position

1 a G

2 b F

3 c F

4 d G

5 e G

Example 3: Remove Columns in Range

The following code shows how to remove all columns in the range from 'position' to 'rebounds':

```
#remove columns in range from 'position' to 'rebounds'  
df %>% select(-(position:rebounds))
```

player

1 a

2 b

3 c

4 d

5 e

Example 4: Remove Columns that Contain a Phrase

The following code shows how to remove all columns that contain the word 'points'

```
#remove columns that contain the word 'points'  
df %>% select(-contains('points'))
```

player position rebounds

1 a G 5

2 b F 7

3 c F 7

4 d G 12

5 e G 11

Example 5: Remove Columns that Start with Certain Letters

```
#remove columns that start with 'po'  
df %>% select(-starts_with('po'))
```

player rebounds

1 a 5

2 b 7

3 c 7

4 d 12

5 e 11

Example 6: Remove Columns that End with Certain Letters

The following code shows how to remove all columns that end with the letter 's':

```
#remove columns that end with 's'  
df %>% select(-ends_with('s'))
```

player position

1 a G

2 b F

3 c F

4 d G

5 e G

Example 7: Remove Columns by Position

The following code shows how to remove columns in specific positions:

```
#remove columns in position 1 and 4  
df %>% select(-1, -4)
```

position points

1 G 12

2 F 15

3 F 19

4 G 22

5 G 32

Note: You can find the complete documentation for the `select()` function .

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

ARABPSYCHOLOGY.COM