

How can I use dplyr to remove rows from a dataset?

Authored by
stats writer

May 6, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use dplyr to remove rows from a dataset?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=143320>

Dplyr is a powerful data manipulation package in R that allows users to easily remove rows from a dataset. This can be achieved by using the filter function, which allows for the selection of specific rows based on certain criteria. Alternatively, the slice function can be used to remove rows by their position in the dataset. Both of these functions can be further customized by using logical operators, such as "!=" or "%in%", to specify the exact rows to be removed. Overall, dplyr provides efficient and seamless methods for removing rows from a dataset, making it a valuable tool for data cleaning and analysis.

Remove Rows Using dplyr (With Examples)

You can use the following basic syntax to remove rows from a data frame in R using dplyr:

1. Remove any row with NA's

```
df %>%  
na.omit()
```

2. Remove any row with NA's in specific column

```
df %>% filter(!is.na(column_name))
```

3. Remove duplicates

```
df %>%  
distinct()
```

4. Remove rows by index position

```
df %>% filter(!row_number() %in% c(1, 2, 4))
```

5. Remove rows based on condition

```
df %>%  
filter(column1=='A' | column2 > 8)
```

The following examples show how to use each of these methods in practice with the following data frame:

```
library(dplyr)  
  
#create data frame  
df <- data.frame(team=c('A', 'A', 'B', 'B', 'C', 'C'),  
points=c(4, NA, 7, 5, 9, 9),  
assists=c(1, 3, 5, NA, 2, 2))  
  
#view data frame  
df  
  
team points assists  
1 A 4 1  
2 A NA 3  
3 B 7 5  
4 B 5 NA
```

5 C 9 2

6 C 9 2

Example 1: Remove Any Row with NA's

The following code shows how to remove any row with NA values from the data frame:

```
#remove any row with NA  
df %>%  
na.omit()
```

```
team points assists
```

```
1 A 4 1
```

```
3 B 7 5
```

```
5 C 9 2
```

```
6 C 9 2
```

Example 2: Remove Any Row with NA's in Specific Columns

#remove any row with NA in 'points' column:

```
df %>%  
filter(!is.na(points))
```

```
team points assists
```

```
1 A 4 1
```

2 B 7 5

3 B 5 NA

4 C 9 2

5 C 9 2

Example 3: Remove Duplicate Rows

The following code shows how to remove duplicate rows:

```
#remove duplicate rows
```

```
df %>%
```

```
distinct()
```

```
team points assists
```

```
1 A 4 1
```

```
2 A NA 3
```

```
3 B 7 5
```

```
4 B 5 NA
```

```
5 C 9 2
```

Example 4: Remove Rows by Index Position

The following code shows how to remove rows based on index position:

```
#remove rows 1, 2, and 4df %>%  
filter(!row_number() %in% c(1, 2, 4))
```

team points assists

1 B 7 5

2 C 9 2

3 C 9 2

Example 5: Remove Rows Based on Condition

The following code shows how to remove rows based on specific conditions:

```
#only keep rows where team is equal to 'A' or points is  
greater than 8df %>%  
filter(column1=='A' | column2 > 8)
```

team points assists

1 A 4 1

2 A NA 3

3 C 9 2

4 C 9 2

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