

How can I filter a data frame in R using dplyr without losing rows containing NA values?

Authored by
stats writer

June 25, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I filter a data frame in R using dplyr without losing rows containing NA values?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=151840>

One can use the dplyr package in R to filter a data frame without losing rows that contain NA values. This can be achieved by using the filter() function and specifying the conditions for filtering, including the presence of NA values. This approach ensures that the data frame is filtered based on the desired criteria without removing any rows containing missing values.

Filter Data Frame without Losing NA Rows Using dplyr

You can use the following basic syntax to filter a data frame without losing rows that contain NA values using functions from the dplyr and tidyr packages in R:

```
library(dplyr)
```

```
library(tidyr)
```

```
#filter for rows where team is not equal to 'A' (and keep  
rows with NA)df <- df %>% filter((team != 'A') %>%  
replace_na(TRUE))
```

Note that this formula uses the replace_na() function from the tidyr package to convert NA values to TRUE so they aren't dropped from the data frame when filtering.

The following example shows how to use this syntax in practice.

Example: Filter Data Frame without Losing NA Rows Using dplyr

Suppose we have the following data frame in R that contains information about various basketball players:

```
#create data frame
```

```
df <- data.frame(team=c('A', NA, 'A', 'B', NA, 'C', 'C', 'C'),  
points=c(18, 13, 19, 14, 24, 21, 20, 28),  
assists=c(5, 7, 17, 9, 12, 9, 5, 12))
```

```
#view data frame
```

```
df
```

```
team points assists
```

```
1 A 18 5
```

```
2 <NA> 13 7
```

```
3 A 19 17
```

```
4 B 14 9
```

```
5 <NA> 24 12
```

```
6 C 21 9
```

```
7 C 20 5
```

```
8 C 28 12
```

Now suppose we use the filter() function from the dplyr package to filter the data frame to only contain rows where the value in the team column is not equal to A:

```
library(dplyr)
```

```
#filter for rows where team is not equal to 'A'
```

```
df <- df %>% filter(team != 'A')
```

```
#view updated data frame
```

```
df
```

```
team points assists
```

```
1 B 14 9
```

```
2 C 21 9
```

```
3 C 20 5
```

```
4 C 28 12
```

Notice that each row where the value in the team column is equal to A has been filtered out, including the rows where the value in the team column is equal to NA.

If we would like to filter out the rows where team is equal to A and *keep* the rows with NA values, we can use the following syntax:

```
library(dplyr)
```

```
library(tidyr)
```

```
#filter for rows where team is not equal to 'A' (and keep
```

```
rows with NA)df <- df %>% filter((team != 'A') %>%  
replace_na(TRUE))
```

```
#view updated data frame
```

```
df
```

```
team points assists
```

```
1 <NA> 13 7
```

```
2 B 14 9
```

```
3 <NA> 24 12
```

```
4 C 21 9
```

```
5 C 20 5
```

```
6 C 28 12
```

Notice that each row where the value in the team column is equal to A has been filtered out, but we kept the rows where the value in the team column is equal to NA.

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

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