How can I remove rows with NA values in R?

Authored by

stats writer

June 24, 2024

To remove rows with NA values in R, one can use the "complete.cases" function. This function identifies and removes rows that contain missing values, indicated by NA, in a specified data frame. This ensures that the remaining data is complete and suitable for further analysis. Additionally, one can also use the "na.omit" function to remove NA values from a specific column or a subset of columns in the data frame. Both of these methods are effective in handling missing data and ensuring the integrity of the dataset for analysis purposes.

How do you remove rows with NA values (missing values) from an R DataFrame (data.frame)? NA stands for Not Available and it is not a number that is considered a missing value. So our task is to remove the rows that contain either some or all NA values. In this article, we'll cover how to remove rows that contain any NA values, as well as those that contain all NA values.

If a row contains some NA values, the following methods are used to drop these rows however, you can also replace NA with 0 or replace NA with an empty string.

`na.omit()complete.cases()rowSums()drop_na()`

If a row contains all NA values, these two methods are used to remove them.

`rowSums()` with ncol`filter()` with `rowSums()`

1. Quick Examples of Removing Rows with NA Values

Following are quick examples of how to remove/delete rows with NA on R DataFrame (data.frame).

```
# Below are the quick examples

# Example 1: Remove rows with NA's using na.omit()
df <- na.omit(df)

# Example 2: Remove rows with NA's using complete.cases
df <- df

# Example 3: Remove rows with NA's using rowSums()
df <- df

# Example 4: Import the tidyr package
library("tidyr")

# Remove rows with NA's using drop_na()
df <- df %>% drop_na()
```

# Example 5: Remove rows that contains all NA's
df <- df

# Example 6: Load the dplyr package
library("dplyr")

# Remove rows that contains all NA's
df <- filter(df, rowSums(is.na(df)) != ncol(df))

Let's create a data frame with 5 rows and 3 columns such that one row contains all NA and some rows contain at least one NA.

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

# Display dataframe
print(df)

Yields below output.

```
  id     name gender
1  2   sravan   <NA>
2  1     <NA>      m
3  3   chrisa   <NA>
4  4 shivgami      f
5 NA     <NA>   <NA>
```

2. Remove Rows with NA From the R Dataframe

By using `na.omit()`, `complete.cases()`, `rowSums()`, and `drop_na()` methods you can remove rows that contain NA (missing values) from the R data frame. Let's see an example for each of these methods.

2.1. Remove Rows with NA using na.omit()

The `na.omit()` function is used to remove any rows with NA values from a data frame and returns the modified data frame.

Syntax of na.omit():

```
# Syntax of na.omit()
na.omit(df)
```

Where `df` is the input data frame

Example:

In this example, we will apply `na.omit()` to the given data frame and drop the rows that contain some NA values.

```
# Remove rows with NA's using na.omit()
print(na.omit(df))
```

Yields below output.

```
  id     name gender
4  4 shivgami      f
```

Notice that the above resultant data frame has no rows with NA values.

2.2. Remove Rows with NA using complete.cases()

The `complete.cases()` function removes rows that contain some NA values and returns the modified data frame having no NA values.

Syntax of complete.cases() function

```
# Syntax of complete.cases()
df
```

Example:

In this example, you can apply this function to a given data frame, it will remove the rows which contain some NA. Let's pass the data frame into this function to remove the rows having at least one NA value.

```
# Remove rows with NA's using complete.cases
print(df )
```

Output:

```
# Output
id name gender
4 4 shivgami f
```

We can see that the above row has no NA values.

2.3. Remove rows with NA using rowSums()

In this example, you can use the `rowSums()` function to filter out rows without any NA values. `rowSums(is.na(df)) == 0` this syntax calculates the sum of NA values for each row in the dataframe. (`is.na(df)` creates a logical matrix of the same dimensions as `df`. TRUE for every NA value and `FALSE` otherwise), and then checks if the sum is equal to 0. Using this condition you can remove the rows having NA values.

Syntaxof rowSums() function:

```
# Syntax of rowSums() function
df
```

Example:

In this example, we will apply `rowSums()` to the data frame and remove the rows having some NA. df, this syntax subsets the dataframe, keeping only the rows where the condition is `TRUE`. In other words, it selects rows that have no NA values in any of their columns.

```
# Remove rows with NA's using rowSums()
print(df )
```

Output:

```
# Output
id name gender
```

```
4 4 shivgami f
```

2.4. Remove rows with NA using drop_na()

`drop_na()` function will drop the rows that contain at least one NA value. It is available in tidyr package. tidyr is a third-party library hence, to use the `tidyr` library, first, you need to install it by using install.packages('tidyr'). Once installation is completed, load the tidyr library to use this `dro_na()` method. To load a library in R language you can use `library("tidyr")`.

Syntax:

```
# Syntax
df %>% drop_na()
```

where `df` is the input data frame and `%>%` loads the method to the data frame.

Example:

In this example, we will apply `drop_na()` to remove rows with some NA. Let's apply this method and get the rows without having the NA values.

```
#import the tidyr package
library("tidyr")
```

```
# Remove rows with NA's using drop_na()
print(df %>% drop_na())
```

Output:

```
# Output
id name gender
4 4 shivgami f
```

3. Remove Rows Containing all NA Values in the R Dataframe

So far, we have seen how to remove rows that have NA on any columns. In this section, we will remove the rows with NA on all columns in an R data frame (data.frame).

3.1. Remove Rows with All NA using rowSums() with ncol

Here, we are comparing `rowSums()` count with `ncol()` count, if they are not equal, we can say that the row doesn't contain all NA values. Hence the row that contains all NA will not be selected. When the counts are equal then the row is considered with all NA values and the row is considered to be removed from the R dataframe.

Syntax of rowSums():

```
# Syntax
df
```

Example:

In this example, we will apply `rowSums()` and `ncol()` methods to remove rows with all NA.

```
#Remove rows that contains all NA's
print(df)
```

Output:

```
# Output
id name gender
1 2 sravan <NA>
2 1 <NA> m
3 3 chrisa <NA>
4 4 shivgami f
```

We can see that the id with the value 5 row is deleted since it contains all NA values.

3.2. Delete Rows with NA using filter() with rowSums()

It is similar to the above method, we are comparing `rowSums()` count with `ncol()` count, if they are not equal, we can say that the row doesn't contain all NA values. Hence the row that contains all NA will not be selected. To do this, we have to use the `filter()` method.

Syntax:

```
# Syntax
filter(df, rowSums(is.na(df)) != ncol(df))
```

Example:

In this example, we will apply `filter()` with `rowSums()` to remove rows with all NA.

```
#Load the dplyr package
library("dplyr")
```

#Remove rows that contains all NA's
print(filter(df, rowSums(is.na(df)) != ncol(df)))

Output:

```
# Output
id name gender
1 2 sravan <NA>
2 1 <NA> m
3 3 chrisa <NA>
4 4 shivgami f
```

We can see that the id-5 row is deleted since it contains all NA.

4. Conclusion

From this article, we have seen how to remove the rows that contain NA values from the R dataframe. If you want to remove the rows that contain all NA values, you can use the combination of `rowSums()`, `ncol()`, and `filter()` from the `dplyr` package. If you want to delete the rows that have some NA values, you can use `rowSums()`, `drop_na()` from `tidyr` package, `na.omit()` and `complete.cases()`

Related Articles

References

Missing values in RrowSums()