How can I replace all instances of NA with an empty string in a DataFrame using the -R option?

Authored by

stats writer

June 24, 2024

The -R option is a command line function that can be used to replace all instances of "NA" with an empty string in a DataFrame. This function is useful for cleaning and organizing data in a more efficient manner. By using the -R option, all "NA" values in the DataFrame will be replaced with an empty string, allowing for easier data manipulation and analysis. This method is particularly helpful for handling missing data in a dataset.

How to replace NA (missing values) with blank space or an empty string in an R dataframe? You can replace NA values with blank spaces on columns of the R dataframe (data.frame) using is.na(), replace() methods. And use dplyr::mutate_if() to replace only character columns when you have mixed numeric and character columns, use dplyr::mutate_at() to replace multiple selected columns by index and name.

Typically, NA values are treated as missing data, and performing operations on them can lead to inconsistent results. Therefore, it is advisable to address these missing values before processing the data. Similarly, using these you can also replace NA with zero (0) in R.

1. Quick Examples of Replacing NA's with Empty String

Following are quick examples of replacing NA values with an empty string in R.

```
# Quick Examples

#Example 1 - Replace na values with blank using is.na()
my_dataframe <- ""

#Example 2 - By using replace() & is.na()
my_dataframe <- replace(my_dataframe, is.na(my_dataframe), "")

# All below examples need to load these libraries
library("dplyr")
library("tidyr")

#Example 3 - Replace only string columns
my_dataframe <- my_dataframe %>%
mutate_if(is.character, ~replace_na(.,""))

# Example 4 - Replace on selected columns by Name
my_dataframe <- my_dataframe %>%
mutate_at(c('name','gender'), ~replace_na(.,""))

# Example 5 - Replace on selected columns by Index
```

my_dataframe <- my_dataframe %>%
mutate_at(c(1,2), ~replace_na(.,""))

Let's create a dataframe having NA values,

```
#Create dataframe
my_dataframe=data.frame(
name=c('sravan',NA,'chrisa','shivgami',NA),
gender=c(NA,'m',NA,'f',NA))
```

#Display dataframe
print(my_dataframe)

Output:

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

2. Replace NA's with Empty String

The `is.na()` function in R determines if a particular data frame column value equals NA. When a value is NA, it generates `TRUE`, otherwise `FALSE`. So by specifying it within square brackets, it replaces NA with an empty string. This method facilitates the replacement of NA values with an empty string in an R data frame.

Syntax:

```
#Syntax
df = "value to replace"
```

where `my_dataframe` is the input dataframe. Let's run an example to update NA values with blank space in R dataframe.

```
#Replace na values with blank using is.na()
my_dataframe <- ""
```

#Display the dataframe
print(my_dataframe)

Output:

```
#Output
name gender
1 sravan
2 m
3 chrisa
4 shivgami f
5
```

In the above output, we can see that NA values are replaced with blank spaces.

3. Replace NA values with Blank Space using replace()

Let's explore another method to replace NA values with empty string using the replace() function, which requires three parameters.

Syntax:

```
#Syntax
replace(df,is.na(df),"value to replace")
```

Parameters:

The first parameter is the input data frame. The second parameter uses the `is.na()` method to check for missing values (NA). The final parameter specifies the value "", which will replace any missing values identified by the second parameter.

Example: Replace NA with blank space in the dataframe using replace()

```
#By using replace() & is.na()
my_dataframe <- replace(my_dataframe, is.na(my_dataframe), "")
```

```
#Display dataframe
print(my_dataframe)
```

Yields the same output as above.

Alternatively, you can also write the above statement using `%>%` operator. To use this, load the library `dplyr`.

```
#Example 2 - Using %>%
library("dplyr")
my_dataframe <- my_dataframe %>%
replace(is.na(my_dataframe), "")
print(my_dataframe)
```

4. Replace NA with Empty String only on Character Columns

All examples above use dataframe with only characters hence renaming NA with an empty string is straight forward but, in real-time we would get a mix of numeric and character columns, and running the above examples results in an error hence, we need to use qualifiers to apply the change only on character columns ignoring numeric columns.

You can apply conditions by using `dplyr::mutate_if()` and `is.character` is used to check if the column is a character or not and apply `tidyr::replace_na()` only on character columns.

```
#Create dataframe
my_dataframe=data.frame(
id=c(2,1,3,4,NA),
name=c('sravan',NA,'chrisa','shivgami',NA),
gender=c(NA,'m',NA,'f',NA))

#Load library
library("dplyr")
library("tidyr")

#Replace only character columns
my_dataframe <- my_dataframe %>%
mutate_if(is.character, ~replace_na(., ""))
print(my_dataframe)
```

Yields below output. Notice that colid `id` still have NA values as it's been ignored because it holds numeric values.

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

5. Replace NA with Empty String on Selected Multiple Columns

To replace NA with an empty string on selected multiple columns by name use `mutate_at()` function with vector `c()` of column names.

```
# Replace on selected multiple columns
library("dplyr")
library("tidyr")
my_dataframe <- my_dataframe %>%
mutate_at(c('name','gender'), ~replace_na(.,""))
print(my_dataframe)
```

Yields the same output as above.

6. Replace NA with Empty String on Selected Multiple Index

Finally, if you wanted to replace NA with an empty string on selected multiple r dataframe columns by index use `mutate_at()` function with vector `c()` of index values.

```
# Replace on selected multiple index
library("dplyr")
library("tidyr")
my_dataframe <- my_dataframe %>%
mutate_at(c(2,3), ~replace_na(.,""))
print(my_dataframe)
```

Yields the same output as above. You can find more details on the `mutate()` function and its

variants in the R Documentation.

6. Conclusion

In this article, I have explained multiple ways to replace NA also called missing values with blank space or an empty string in the R dataframe by using `is.na()`, `replace()` methods. And use `dplyr::mutate_if()` to replace only on character columns when you have mixed numeric and character columns, use `dplyr::mutate_at()` to replace on multiple selected columns by index and name.

Related Articles

References

replace() in RimputeTS() package in RWhat is NA or Missing Values?