How can I replace an empty string with NA using the -R function?

Authored by

stats writer

June 24, 2024

The -R function is a powerful tool that allows users to modify and manipulate data in various ways. One common task is replacing empty strings with the value "NA". This can be achieved by using the -R function and specifying the empty string as the target to be replaced, followed by the desired value "NA". This process can be useful for data cleaning and formatting, as well as for analysis purposes. By using the -R function to replace empty strings with "NA", users can ensure that their data is properly formatted and ready for further analysis.

How to replace an empty string with NA in R DataFrame (data.frame)? By using methods from `R built-in`, and `dplyr` package we can replace empty strings with NA values on the data frame. In this article, I have covered different ways to replace. Also, I have covered replacing empty string with NA on a single column, multiple columns, and by index position with examples.

In case you want to replace zero with NA, refer to this article.

1. Quick Examples of Replace Empty String with NA Value

Following are quick examples of how to replace an empty string with an NA value in an R Dataframe.

```
# Quick Examples of replacing empty string with NA

# Example 1 - Replace on all columns
df <- NA
print(df)

# Example 2 - Replace on selected columns
df == ''] <- NA
print(df)

# Example 3 - Using replace() function
df <- replace(df, df=='', NA)
print(df)

# Example 4 - Replace using dplyr::na_if()
library(dplyr)
df <- na_if(df, '')
print(df)

# Example 5 - Replace using dplyr::mutate_all()
library(dplyr)
df <- df %>% mutate_all(~na_if(., ''))
```

```
print(df)

# Example 6 - Replace only on all Numeric columns
library(dplyr)
df <- df %>% mutate_if(is.character, ~na_if(., ''))
print(df)

# Example 7 - Replace only on selected columns
library(dplyr)
df <- df %>% mutate_at(c('name'), ~na_if(., ''))
print(df)

# Example 8 - Replace only on selected column index
library(dplyr)
df <- df %>% mutate_at(c(2), ~na_if(., ''))
print(df)

# Example 9 - Replacing on tibble
df2 <-tibble(
col1 = c("A", "B", "NA"),
col2 = c(0, 2, NA),
col3 = c(1, NA, 5)
)
df2 <- df2 %>% mutate_if(is.numeric , replace_na, replace = '')
print(df2)
```

Let's create an R data frame, run these examples, and validate the results.

```
# Create dataframe with numeric columns
df=data.frame(id=c(2,1,3),
name=c('ram','','chrisa'),
gender=c('','m',''))
df
# Output
# id name gender
# 1 2 ram
# 2 1 m
# 3 3 chrisa
```

2. Replace Empty String with NA in an R Dataframe

As you saw above R provides several ways to replace Empty/Blank String with NA on a data frame, among all the first approach would be using the directly R base feature. Use df to check if the value of a data frame column is an empty string, if it is an empty string you can assign the value `NA`. The below example replaces all blank string values on all columns with NA. I have created another article replace NA with empty string which is the reverse of what we are learning here.

```
#Example 1 - Replace on all columns
df <- NA
print(df)
```

#Output
id name gender
#1 2 ram <NA>
#2 1 <NA> m
#3 3 chrisa <NA>

This is the most generic approach where you can use this on vector as well to replace its values.

3. Replace Selected Columns

When you have multiple columns in R data frame and you would require to select a single column to replace the empty string with NA, you can achieve this by following. This updates only column `name`.

```
#Example 2 - Replace on selected olumns
df == ''] <- NA
print(df)
```

#Output
id name gender
#1 2 ram
#2 1 <NA> m
#3 3 chrisa

4. Using R replace() function to update Empty String with NA

R has a built-in function called `replace()` that replaces values in a vector with another value, for example, blank space with NAs.

```
#Example 3 - Using replace() function
df <- replace(df, df=='', NA)
print(df)
```

#Output
id name gender
#1 2 ram <NA>
#2 1 <NA> m
#3 3 chrisa <NA>

5. Update Empty String with NA using R dplyr::na_if()

All previous examples use the Base R built-in functions that can be used on a smaller dataset but, for bigger data sets, you have to use methods from `dplyr` package as they perform 30% faster. `dplyr` package uses C++ code to evaluate.

The dplyr is third-party package that is required to install first using `install.packages('dplyr')` and load it using `library("dplyr")`. `na_if()` is a method from `dplyr` package.

```
#Example 4 - Replace using dplyr::na_if()
library(dplyr)
df <- na_if(df, '')
print(df)
```

#Output
id name gender
#1 2 ram <NA>
#2 1 <NA> m
#3 3 chrisa <NA>

6. Update Empty String with NA using dplyr::mutate_all()

`mutate_all()` is another method in dplyr package to substitute the empty string with NA value on

all data frame columns.

```
#Example 5 - Replace using dplyr::mutate_all()
library(dplyr)
df <- df %>% mutate_all(~na_if(., ''))
print(df)
```

#Output
id name gender
#1 2 ram <NA>
#2 1 <NA> m
#3 3 chrisa <NA>

7. Replace on All Character columns

`mutate_if()` affects variables selected with a predicate function, here `is.numeric` is used as a predicate to replace values only on numeric columns. Since we have all numeric columns, it updates all columns with NA for value empty string.

```
#Example 6 - Replace only on all Character columns
library(dplyr)
df <- df %>% mutate_if(is.character, ~na_if(., ''))
print(df)
```

#Output
id name gender
#1 2 ram <NA>
#2 1 <NA> m
#3 3 chrisa <NA>

Yields the same output as above.

8. Replace Blank String with NA Only on Selected Columns

`mutate_at()` affects variables selected with a character `vector` or `vars()`. Here we update values only on `pages` column.

```
#Example 7 - Replace only on selected columns
```

```
library(dplyr)
df <- df %>% mutate_at(c('name'), ~na_if(., ''))
print(df)
```

#Output
id name gender
#1 2 ram
#2 1 <NA> m
#3 3 chrisa

9. Replace Blank String with NA on Selected Column Indexs

If you pass a vector with index position to `mutate_at()`, it replaces all blank values with NA on selected index position columns in R dataframe. This updates index 2 which is `name` column. Note that in R the index starts from 1.

```
#Example 8 - Replace only on selected column index
library(dplyr)
df <- df %>% mutate_at(c(2), ~na_if(., ''))
print(df)
```

#Output
id name gender
#1 2 ram
#2 1 <NA> m
#3 3 chrisa

10. Conclusion

In this article, I have covered 10 ways to replace the empty or blank string with NA value in an R data frame. Also, I have covered how to replace it on a single column, multiple columns, and columns with index position using the R base function and `dplyr` package methods.

Related Articles

References