How can I replace values in an R data frame based on a certain condition?

Authored by

stats writer

June 24, 2024

To replace values in an R data frame based on a certain condition, one can use the "ifelse" function. This function allows users to specify a condition and provide the replacement value if the condition is met. This can be useful for updating or correcting data within a data frame, ensuring that it meets specific criteria. Additionally, the "replace" function can be used to directly replace values within a specific column of the data frame. Both of these methods allow for efficient and accurate manipulation of data within an R data frame.

There are multiple ways to replace column values based on conditions in an R DataFrame. Conditionally updating columns is a very basic thing we do all the time while manipulating data.

In this article, I will explain how to replace values based on single/multiple logical conditions, and conditions on numeric and character columns in the R dataframe

First, Let's create an R DataFrame.

```
# Create dataframe with numeric columns
df = data.frame(id=c(25,40,30,30),
name=c('Chris','Scott','Anna','Ramana'),
gender=c('m','m','f','m'),
marks1=c(99,30,50,NA),
marks2=c(80,99,60,45))
df
```

Yields below output.

```
# Output
id name gender marks1 marks2
1 25 Chris m 99 80
2 40 Scott m 30 99
3 30 Anna f 50 60
4 30 Ramana m NA 45
```

1. Replace Values Based on Condition in R

Replace column values based on checking logical conditions in R DataFrame is pretty straightforward. All you need to do is select the column vector you want to update and use the condition within .

The following example demonstrates how to update DataFrame column values by checking

conditions on a numeric column. It updates column `id` to `55` when its value is equal to `40`.

```
# Replace Values Based on Condition
df$id <- 55
df
```

Yields below output. You can also use this approach to replace NA with 0 or replace NA with an empty string in R.

```
# Output
id name gender marks1 marks2
1 25 Chris m 99 80
2 55 Scott m 30 99
3 30 Anna f 50 60
4 30 Ramana m NA 45
```

2. Check the Condition of the Character Column

Similarly, you can also update the column value by checking the condition of the character column. The following example replaces the `name` column with the `Jeni` string when it finds the `name` value is equal to `Chris`.

```
# Check Condition on Character Column
df$name <- "Jeni"
df
```

Yields below output.

```
# Output
id name gender marks1 marks2
1 25 Jeni m 99 80
2 55 Scott m 30 99
3 30 Anna f 50 60
4 30 Ramana m NA 45
```

3. Replace Values in Column Based on Multiple Conditions

Now, let's see how to replace column values by checking multiple conditions in R. The following example demonstrates using & operator with two conditions. It updates column `id` with value `60` when `id` is equal to `55` and `gender` is equal to `'m'`.

```
# Replace by Checking Multiple Conditions
df$id <- "60"
df
```

Yields below output.

```
# Output
id name gender marks1 marks2
1 25 Jeni m 99 80
2 60 Scott m 30 99
3 30 Anna f 50 60
4 30 Ramana m NA 45
```

Replace All DataFrame Columns Conditionally

The below example updates all column values in a DataFrame to `95` when the existing value is `99`. Here, `marks1` and `marks2` have `99` value hence, these two values are updated with `95`.

```
# Replace all columns by condition
df <- 95
df
```

Yields below output.

```
# Output
id name gender marks1 marks2
1 25 Jeni m 95 80
2 60 Scott m 30 95
3 30 Anna f 50 60
4 30 Ramana m NA 45
```

4. Using data.table to Replace Values Conditionally

If you have `data.table`, then use the following approach to replace values Conditionally. This performs much faster than the traditional approach.

First, you need to load the library using `library("data.table")`. In case you don't have this package, install it using `install.packages("data.table")`.

```
#Load dplyr package
library("data.table")
```

```
# Replace conditionally using data.table.
df2 = as.data.table(df)
df2
df2
```

Yields below output.

```
# Output
id name gender marks1 marks2
1: 25 Chris m 99 80
2: 40 Scott m 30 99
3: 60 Anna f 50 60
4: 60 Ramana m NA 45
```

5. Replace Column Based on Condition Using dplyr Package

To use this mutate() method, first, you need to load its library using `library("dplyr")`. In case you don't have this package, install it using `install.packages("dplyr")`. The dplyr package provides a set of functions to work with strings as easily as possible.

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 the `dplyr` package as they perform 30% faster. This package uses C++ code to evaluate.

Let's see how we can write the above examples using `dplyr::mutate()`

```
#Load dplyr package
```

```
library(dplyr)

# Create dataframe with numeric columns
df=data.frame(id=c(25,40,30,30,45,40),
marks1=c(99,30,50,NA,40,50),
marks2=c(80,99,60,45,NA,60))
df

# Output
# id marks1 marks2
#1 25 99 80
#2 40 30 99
#3 30 50 60

# Replace using mutate() function and checking condition
# Replaces when id==30
df <- mutate(df, id = case_when(
id == 30 ~ 40,
TRUE ~ id
))
df

#Output
# id marks1 marks2
#1 25 99 80
#2 40 30 99
#3 40 50 60
```

6. Complete Examples of Replace Values Based on Condition

Following is a complete example of how to replace column values based on conditions in R DataFrame.

```
# Create dataframe with numeric columns
df = data.frame(id=c(25,40,30,30),
name=c('Chris','Scott','Anna','Ramana'),
gender=c('m','m','f','m'),
marks1=c(99,30,50,NA),
marks2=c(80,99,60,45))
df
```

```
# Example 1 - Replace Column Value Based on Condition
df$id <- 55
df

# Example 2 - Replace by Checking Condition on Character Column
df$name <- "Jeni"
df

# Example 3 - Replace Column Value by Checking Multiple Conditions
df$id <- "60"
df

# Example 4 - Replace all DataFrame columns by condition
df <- 95
df

# Example 5 - Using data.table
library('data.table')
df2 = as.data.table(df)
df2
df2

# Create dataframe with numeric columns
df=data.frame(id=c(25,40,30,30,45,40),
marks1=c(99,30,50,NA,40,50),
marks2=c(80,99,60,45,NA,60))
df

# Example 6 - Using dplyr
# Using this only on numeric columns df.
library('dplyr')
df <- mutate(df, id = case_when(
id == 30 ~ 40,
TRUE ~ id
))
df
```

Conclusion

In this article, I have explained how to replace values based on a single logical condition, multiple conditions, conditions on numeric and character columns etc. Also covered using the data.table

and dplyr packages.

Related Articles

References