How to replace values in R with examples?

Authored by

stats writer

June 24, 2024

Replacing values in R is a common task in data analysis or programming. This involves replacing existing values in a dataset or object with new values. To do this, we use the "replace" function in R. The syntax for this function is "replace(x, list, values)" where x is the dataset or object, list is the list of values to be replaced, and values is the new value to be inserted. For example, if we have a vector "x" with values 1, 2, 3, and we want to replace the value 2 with 4, we would use the code "replace(x, 2, 4)". This will replace all occurrences of 2 in the vector with the value 4. This function can also be used for replacing values in data frames or matrices. Additionally, we can use conditional statements to replace values based on certain conditions. Overall, the "replace" function in R is a useful tool for manipulating data and making changes to existing values.

There are several ways to replace/update column values in R DataFrame. In this article, I will explain how to update data frame values of single/multiple/all columns by using the R base functions/notation and dplyr package.

Let's create an R DataFrame, run these examples, and explore the output. If you already have data in CSV you can easily import CSV files to R DataFrame. Also, refer to Import Excel File into R.

```
# Create dataframe
df <- data.frame(id=c(1,2,3,NA),
address=c('Orange St','Anton Blvd','Jefferson Pkwy',''),
work_address=c('Main St',NA,'Apple Blvd','Portola Pkwy'))
```

df

Output:

```
  id       address work_address
1  1     Orange St      Main St
2  2    Anton Blvd         <NA>
3  3 Jefferson Pkwy   Apple Blvd
4 NA                 Portola Pkwy
```

Notice that the column names are: `id`, `pages`, `name`, `chapters`, and `price`.

1. Update Specified Data Frame Column Value

To replace a column value in R you can use square bracket notation `df`, By using this you can update values on a single column or all columns. To refer to a single column you can use

`df$column_name`. The following example updates `Orange St` with `Portola Pkwy` on the `address` column.

```
# Replace String with Another Stirng on a single column
df$address <- 'Portola Pkwy'
df
```

Yields below output.

```
  id        address work_address
1  1   Portola Pkwy      Main St
2  2     Anton Blvd         <NA>
3  3 Jefferson Pkwy   Apple Blvd
4 NA                Portola Pkwy
```

To update all column values use `df` notation. You can use `df <- "Orange St"` it to update all columns by replacing `'Portola Pkwy'`(wherever present in the columns of the data frame) with `'orange St'`.

```
# Replace String with Another String on All Columns
df <- "Orange St"
df
```

Output
id address work_address
1 1 Orange St Main St
2 2 Anton Blvd <NA>
3 3 Jefferson Pkwy Apple Blvd
4 NA Orange St

Alternatively, you can use the str_replace() method from `stringr` the package to update the sub-string of a string(column value) with another sub-string in R DataFrame. The following code updates substring `St` with `Street` on the `address` column.

```
# Replace String with another String
library(stringr)
df$address <- str_replace(df$address, "St", "Street")
print(df)
```

```
# Output
# id address work_address
# 1 1 Orange Street Main St
# 2 2 Anton Blvd <NA>
# 3 3 Jefferson Pkwy Apple Blvd
4 NA Orange St
```

2. Replace with Another Column Value

In R replacing a column value with another column is the most used example, let's say you want to apply some calculation on the existing column and update the result on the same column, this can be achieved with the below R example.

Here I am multiplying column id with the number 5 and assigning the result to the same id column. Similarly, you can also assign the result to another column.

```
# Create new column from existing column
df <- df * 5
df
```

```
# Output
# id address work_address
# 1 5 Orange Street Main St
# 2 10 Anton Blvd <NA>
# 3 15 Jefferson Pkwy Apple Blvd
# 4 NA Orange St
```

3. Update Based on Condition

In case you want to replace column values based on a condition, you need to check with the condition and assign the value from another column to this column when the condition matches. The below example updates the `address` column with the value of `work_address` when only if the address value is `'Orange Street'`.

```
# Replace column value with another based on condition
df$address <- df$work_address
df
```

```
# Output
# id address work_address
# 1 5 Main St Main St
# 2 10 Anton Blvd <NA>
# 3 15 Jefferson Pkwy Apple Blvd
# 4 NA Orange St
```

4. Using dplyr Package

Similarly, you can use mutate() function from dplyr package to change column values, dplyr is a third-party package hence, you need to load the library using `library("dplyr")` to use its methods. In case you don't have this package, install it using `install.packages("dplyr")`.

For bigger data sets it is best to use the methods from the `dplyr` package as they perform 30% faster. This package uses C++ code to evaluate.

```
# Using dplyr package
library(dplyr)
df <- df %>%
mutate(address = ifelse(address == '',work_address,address))
df
```

Yields below output.

```
# Output
id address work_address
1 5 Orange Street Main St
2 10 Anton Blvd <NA>
3 15 Jefferson Pkwy Apple Blvd
4 NA Orange St Orange St
```

5. Update Missing Values with Empty/Blank String

NA values are considered as missing values, to replace these missing (NA) values with empty strings use the below example. Here, the `is.na()` function checks if a column value is NA, if yes then it updates it with an empty string. Similarly, you can also replace empty string with NA values.

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

Output
id address work_address
1 5 Main St Main St
2 10 Anton Blvd
3 15 Jefferson Pkwy Apple Blvd
4 Orange St Orange St

6. Update Missing Values with 0

Replacing all missing values with an empty string is not a good approach as you may have integer values and an empty string is not the right thing to use. So to replace NA with 0 on integer columns use the below approach.

```
# Replace only numeric columns
library("tidyr")
library("dplyr")
```

df <- df %>%
mutate_if(is.numeric, ~replace_na(., 0))

7. Complete Example of Update Column Values

```
# Create dataframe
df <- data.frame(id=c(1,2,3,NA),
address=c('Orange St','Anton Blvd','Jefferson Pkwy',''),
work_address=c('Main St',NA,'Apple Blvd','Portola Pkwy'))
```

df

Replace String with Another Stirng on a single column
df$address <- 'Portola Pkwy'
df

# Replace String with Another String on All Columns
df <- "Orange St"

```
df

# Replace String with another String
library(stringr)
df$address <- str_replace(df$address, "St", "Street")
print(df)

# Create new column from existing column
df <- df * 5
df

# Replace column value with another based on condition
df$address <- df$work_address
df

# Using dplyr package
library(dplyr)
df <- df %>%
mutate(address = ifelse(address == '',work_address,address))
df

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


Frequently Asked Questions of Replace Values in R with Examples

How do I replace specific values in a vector or data frame in R?

Use the `df` notation or logical conditions to replace the specified values with new values in an R data frame. For example, `df$column_name <- 'new_col_value'`

How can I replace missing values (NA) in my data with a specific value?

Use `is.na()` to identify missing values and replace them with a desired value. For example, `df <- 0`

How is it possible to replace multiple values at once?

Using `%in%` or other logical conditions to replace multiple values simultaneously. For example, `vec <- c(1, 2, 3, 4, 5) <br/>vec %in% c(2, 4) <- 0`

8. Conclusion

In this article, you have learned how to replace/update data frame values of single/multiple/all columns by using the R base functions/notation, and the dplyr package with well-defined examples.

Related Articles

References