How can I rename a column in R?

Authored by

stats writer

June 23, 2024

To rename a column in R, the user can use the "colnames" function and specify the old column name and the new desired name. This will change the name of the specified column in the data frame. Alternatively, the user can also use the "rename" function from the "dplyr" package to rename columns in a data frame. This function allows for multiple columns to be renamed at once and also allows for renaming based on specific conditions. Both methods provide a simple and efficient way to rename columns in R.

How to rename a column in the R data frame? You can use `colnames()` and `names()` methods to rename a single column using column index or name and use `rename()` function from the dplyr package along with `"%>%"` operator to rename multiple columns.

In this article, I will explain how to rename columns/variables in the R data frame by using the following approaches. Note that in R the data frame columns are called variables and rows are called observations.

1. Quick Examples of Rename Column

Following are quick examples to rename columns/variables of the R data frame (change/update old column name with new column name).

```
# Below are the quick examples

# Example 1 - Rename second column to c2
colnames(my_dataframe) ="c2"

# Example 2 - Rename fifth column to c5
names(my_dataframe) ="c5"

# Example 3 - Rename the column name - id to c1
colnames(my_dataframe) ="c1"

# Example 4 - Load the library
library("dplyr")
# Using rename()
my_dataframe <- my_dataframe %>%
rename("id" = "c1")

# Example 5 - Rename multiple columns using rename()
my_dataframe <- my_dataframe %>%
rename("id" = "c1",
"pages" = "c2",
```

"name" = "c3")

# Example 6 - Rename column by index
my_dataframe <- my_dataframe %>%
rename(col1 = 1, col2 = 2)

# Example 7 - Using select()
my_dataframe <- my_dataframe %>%
select(col1 = 1, everything())

# Example 8 - Using rename_with()
my_dataframe <- my_dataframe %>%
rename_with(.cols = 1, ~"col1")

# Example 9 - Load library data.table
library(data.table)
# Rename multiple columns for old to new
setnames(my_dataframe, old = c('c1','c2','c3'),
new = c('id','pages','name'))

Let's create an R data frame,

```
# Create data frame
my_dataframe=data.frame(id=c(11,22,33,44,55),
pages=c(32,45,33,22,56),
name=c("spark","python","R","java","jsp"),
chapters=c(76,86,11,15,7),
price=c(144,553,321,567,890))
```

# Display the data frame
print(my_dataframe)

Yields below output.

```
  id pages   name chapters price
1 11    32  spark       76   144
2 22    45 python       86   553
3 33    33      R       11   321
4 44    22   java       15   567
5 55    56    jsp        7   890
```

Notice that the column names are: `id`, `pages`, `name`, `chapters`, and `price`. You can rename the column name of a single R data frame column using several methods.

2. R Rename Column using colnames()

The R base `colnames()` function R to rename single/multiple/all columns present in the data frame. By using this you can rename a column by column index /column name.

Let's use this method to rename all the columns presented in the R data frame with new column names at a time. for example,

```
# Rename all old column names to new column names
colnames(my_dataframe) <- c("c1", "c2", "c3", "c4", "c5")
print(my_dataframe)
```

Yields below output.

```
  c1 c2     c3 c4  c5
1 11 32  spark 76 144
2 22 45 python 86 553
3 33 33      R 11 321
4 44 22   java 15 567
5 55 56    jsp  7 890
```

As we can see from the above code has returned the dataframe with all new column names. Now we will see how to rename the specified column name by its index. For example,

Alternatively, you can also use the `name()` method to rename the column of the R data frame. Note that indexing starts with 1 in R, not zero like in other languages.

Syntax:

```
# Syntax using colnames()
colnames(my_dataframe) = "new_column_name"
```

```
# Syntax using names()
names(my_dataframe) = "new_column_name"
```

Here, `index_pos` indicates the column index. In this example, we will rename the second column, `pages`, to `c2`, and the fifth column, `price`, to `c5`.

```
# Change second column pages to c2 using colnames()
colnames(my_dataframe) = "c2"
```

```
# Change fifth column to c5 using names()
names(my_dataframe) = "c5"

# Display the dataframe
print(my_dataframe)
```

Yields below output.

```
  id c2   name chapters  c5
1 11 32  spark       76 144
2 22 45 python       86 553
3 33 33      R       11 321
4 44 22   java       15 567
5 55 56    jsp        7 890
```

We can see that the second and fifth column names are changed.

3. Rename the Column by Name in R

Sometimes you would be required to rename a column by column name in R, when you do by name you don't have to know the index of the column you would like to change. To do this, you have to specify a condition. We will check by comparing it with the old column name and assigning

a new column name to it.

Syntax:

```
# Syntax rename with condition
colnames(my_dataframe) ="new_column_name"
```

Let's run an example.

```
# Change the column name - id to c1
colnames(my_dataframe) = "c1"
# (or)
# Change the column name - id to c1
names(my_dataframe) = "c1"
```

# Display the dataframe
print(my_dataframe)

Yields the below output.

```
# Output:
c1 c2 name chapters c5
1 11 32 spark 76 144
2 22 45 python 86 553
3 33 33 R 11 321
4 44 22 java 15 567
5 55 56 jsp 7 890
```

4. Rename DataFrame Column in R using rename()

rename() function in the dplyr package, is used to modify the particular column name present in the data frame. The operator `%>%` sends the new column names to the data frame. Moreover, you can use this function to rename the multiple column names. After using the %>% operator, the subsequent functions are applied to the data frame on the left.

If you want to use `dplyr` functions in the R environment first, install the dplyr package like `install. packages('dplyr')`. After completing the installation, you can load the `dplyr` library by `library("dplyr")`.

Syntax:

```
#Load the library
library("dplyr")
# Use rename()
my_dataframe %>% rename(new_column_name = old_column_name)
```

Here,

`my_dataframe` is given data frame`new_column_name` is the new column name that replaces the `old_column_name`

Let's rename a column from `c1` to `id` by using `rename()`.

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

Change the column name - c2 to pages
my_dataframe %>%
rename("id" = "c1")

Yields below output. This changes the col name from `c1` to `id`.

```
# Output:
id c2 name chapters c5
1 11 32 spark 76 144
2 22 45 python 86 553
3 33 33 R 11 321
4 44 22 java 15 567
5 55 56 jsp 7 890
```

Note that the above example doesn't change the column on the existing data frame. print() of this data frame results in unchanged data. To update assign this statement to the existing data frame.

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

#Update the column on existing dataframe

```
my_dataframe <- my_dataframe %>%
rename("id" = "c1")
print(my_dataframe)
```

Yields the same output as above, but updates the column names on `my_dataframe`.

5. Change Multiple Columns using rename()

Similarly, you can also rename multiple columns by name using `rename()` function on the R data frame, here all you need is to know your old and new column names.

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

```
# Rename multiple columns
my_dataframe <- my_dataframe %>%
rename("pages" = "c2",
"price" = "c5")
print(my_dataframe)
```

Yields below output

```
# Output
id pages name chapters price
1 11 32 spark 76 144
2 22 45 python 86 553
3 33 33 R 11 321
4 44 22 java 15 567
5 55 56 jsp 7 890
```

6. Change All Columns by rename_with()

Alternatively, you can use the `rename_with()` function to rename all columns in a data frame. In the below example, we will use this function to convert all column names to uppercase.

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

```
# Rename using rename_with()
my_dataframe <- my_dataframe %>%
rename_with(toupper)

print(my_dataframe)
```

Yields below output.

```
# Output
ID PAGES NAME CHAPTERS PRICE
1 11 32 spark 76 144
2 22 45 python 86 553
3 33 33 R 11 321
4 44 22 java 15 567
5 55 56 jsp 7 890
```

7. By using setnames()

You can use `setnames()` function from data.table library to change columns with list. `data.table` is also a third-party library hence, you need to first install it by using `install.packages('data.table')`. Once installation is completed, load the `data.table` library using `library("data.table")`.

```
# Rename all columns for old to new
# Rename columns from list
setnames(my_dataframe, new = c('c1','c2','c3','c4','c5'),
old = c('ID','PAGES','NAME','CHAPTERS','PRICE'))
```

Frequently Asked Questions on Rename Column in R

How can I rename a column in a data frame?

You can use the `colnames()` or `names()` function to rename a specific column in a data frame. Here's an example, `colnames(df) <- "new_name"`

How do I rename multiple columns at once?

You can use the `rename()` function of `dplyr` package with multiple arguments to rename the multiple columns of data frame at once. For example, `df <- df %>% rename(new_name1 =`

```
old_name1, new_name2 = old_name2, ...)
```

What if I want to rename columns based on a vector of new names?

Simply, you can use the `colnames()` function to rename columns based on a vector of new names. For example, `colnames(df) <- c("new_name1", "new_name2", "new_name3",....)`

How can I remove spaces from column names?

You can use the `make.names()` function to remove spaces and make column names syntactically valid. Here's an example, `colnames(df) <- make.names(colnames(df))`

8. Conclusion

In this article, you have learned how to rename a single column/variable name, multiple and all columns of the R data frame (data.frame) using the colnames(), names() function through column index and conditionally. Also, we discussed using rename() and setnames().

Related Articles:

References