How can I rename a column in R based on its index position?

Authored by

stats writer

June 24, 2024

In R, columns in a data frame can be renamed based on their index position by using the "colnames" function and specifying the desired index number followed by the new name within the brackets. This allows for a quick and efficient way to rename columns without having to manually specify each column name. This method is particularly useful when dealing with large datasets that have a large number of columns. By using the index position, the column name can be changed without the need to know the specific name beforehand.

How to rename column by index in the R data frame? R provides base function `colnames()` and `names()` function to change column name by index position. Besides these, use `dplyrrename()`, `select()` and `rename_with()` to rename/change a DataFrame (data.frame) column.

In a separate article, I have covered different ways to rename columns in R if you have time, I would recommend reading it.

1. Quick Example of Rename Column Name by Index Position

Following are quick examples of renaming data frame columns by Index position in R.

```
#Example 1 - Change second column to c2 using colnames()
colnames(df) = "c2"
```

```
#Example 2 - Change second column to c2 using names()
names(df) = "c2"

# Example 3 - load library
library(dplyr)
# Rename column by index
df <- df %>%
rename(col1 = 1, col2 = 2)

# Example 4 - You can also write as
df <- rename(df, col1 = 1)

# Example 5 - Using select()
df <- df %>%
select(col1 = 1, everything() )

# Example 6 - Using rename_with()
df <- df %>%
rename_with(.cols = 1, ~"col1")
```

Let's create an R data frame (data.frame) and execute the examples above, then verify the results.

```
# Create dataframe
df=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))

# Print dataframe
print(df)
```

Yields below data frame 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
```

2. Rename Column By Index by colnames()

In the R programming language, the colnames() function allows to rename of a single column name within the data frame. Renaming a single column can be done through the column index. We have to specify the index of the column. In R, indexing starts with 1.

Syntax:

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

Example: In this example, we will modify the `identifier` of the second column to `c2` and the `price` of the fifth column to `c5`.

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

#Change fifth column to c5
colnames(df) = "c5"

#Display the dataframe
print(df)

Output:

```
#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 Column by Index using names()

Alternatively, you can also use the `name()` method from the R base to rename. This method exactly behaves the same as `colnames()`.

```
#Change second column to c2 using names()
names(df) = "c2"
```

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

#Display the dataframe
print(df)

Yields the same output as above.

4. Use rename()

Use rename() from R dplyr library to change columns by index in the data frame. This is the best approach as it is easily readable and the code is more organized. Note that dplyr doesn't change the current data frame instead, it returns the new data frame after the rename.

```
#Syntax of rename()
df %>%
rename(new_col_name = col_index)
```

`dplyr` is a third-party library hence, to use the dplyr library, you need to first install it by using install.packages('dplyr'). Once installation is completed, load the `dplyr` library using `library("dplyr")`. If you have `tidyverse` packages installed, load the `tidyverse` packages, which include `dplyr`.

```
# Using rename() from dplyr library

# load library
library(dplyr)

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

# print dataframe
print(df)
```

Yields below output. When you use the `%>%` operator, the functions we use after this will be applied to the data frame to the left of the operator.

```
# Output
col1 col2 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
```

You can also simply write the statement below to rename multiple columns by index.

```
# You can also write as
df <- rename(df, col1 = 1, col2 = 2)
```

5. Use select() Function

Alternatively, you can also use `select()` to change the column name of the data frame by position. select() also understands column indices, so if you're renaming the first column, you can simply do as following.

```
# Using select()
df <- df %>%
select(col1 = 1, everything() )
print(df)
```

Note that using `select()` changes the order of the columns on the resultant data frame when you rename the column in the middle of the R data frame.

6. Using rename_with()

Finally, use the `rename_with()` method to rename column names by Index Position in the R data frame.

```
# Using rename_with()
df <- df %>%
rename_with(.cols = 1, ~"col1")
print(df)
```

Conclusion

In this article, I have explained the usage of `rename()`,`select()`, and `rename_with()` methods to change a data frame column by index position in R. Besides these R also provides a base function `colnames()` and `names()` function to change column name by position.

Related Articles