How can I remove a column in R?

Authored by

stats writer

June 24, 2024

Removing a column in R can be achieved by using the "subset" function. This function allows you to select and remove specific columns from a data frame. The syntax for removing a column is subset(dataframe, select = -column_name). This will create a new data frame without the specified column. Alternatively, you can also use the "select" function and specify the columns you want to keep, leaving out the one you want to remove. These methods are useful for data manipulation and cleaning in R.

To remove a single column or multiple columns in R DataFrame use square bracket notation or use functions from third-party packages like dplyr. There are several ways to remove columns or variables from the R DataFrame (data.frame).

1. Prepare the Data

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(11,22),
pages=c(32,45),
name=c("spark","python"),
chapters=c(76,86),
price=c(144,553))

# Display the DataFrame
print(df)
```

Yields below output.

```
  id pages   name chapters price
1 11    32  spark       76   144
2 22    45 python       86   553
```

2. Remove Column using R Base Functions

Using R base function subset() or square bracket notation you can remove single or multiple columns by index/name from the R DataFrame.

2.1 Remove Column by Index

First, let's use the R base bracket notation `df` to remove the column by Index. This notation takes syntax `df` to select columns in R, and removes them using the - (negative) operator.

The following example removes the second column by Index from the R DataFrame.

```
# Remove Columns by Index
df2 <- df
df2
```

Yields below output.

```
  id   name chapters price
1 11  spark       76   144
2 22 python       86   553
```

2.2 Remove Range of Columns

This notation also supports selecting columns by the range and using the negative operator to remove columns by range. In the following example, remove all rows between 2 and 4 indexes, which ideally remove columns `pages`, `names`, and `chapters`.

```
# Remove specified range of columns
df2 <- df
df2
```

Output
id price
1 11 144
2 22 553

2.3 Remove Multiple Columns

You can use a vector to specify the indexes of the columns that you want to remove from a DataFrame in R. The following example removes multiple columns with indexes 2 and 3.

```
# Remove Multiple columns
df2 <- df
df2
```

Output
id chapters price
1 11 76 144
2 22 86 553

2.4 Remove Columns using name() function

You can also use the column names from the list to remove them from the R DataFrame. Here I am using the `names(df)` function that returns all column names and using %in% c("id", "name", "chapters") to check if the column names("id", "name", "chapters") are presented in the specified vector. Then you can use the `!` operator to select columns NOT in the specified vector.

As a result, `df2` will return only the columns that are NOT `"id"` , `"name"`, or `"chapters"` from the original DataFrame `df`. In this case, it will return only the `"pages"` and `"price"`.

```
# Remove Columns using names()
df2 <- df
df2
```

Output:
pages price
1 32 144
2 45 553

2.5 By using subset() Function

You can use the R base function subset() to remove columns by name from the data frame. This function takes the data frame object as an argument and the columns you want to remove.

```
# Remove columns using subset()
df2 <- subset(df, select = -c(id, name, chapters))
df2
```

Yields the same output as above.

3. Remove Columns by using dplyr Functions

In this section, I will use functions from the dplyr package to remove columns in the R DataFrame. dplyr is an R package that provides a grammar of data manipulation and provides a most used set of verbs that helps data science analysts to solve the most common data manipulation. In order to use this, you have to install it first using `install.packages('dplyr')` and load it using `library(dplyr)`.

3.1 Remove Column by Matching

dplyr select() function is used to select the column and by using negation of this to remove columns. All verbs in the dplyr package are taken `data.frame` as a first argument. When we use `dplyr` package, we mostly use the infix operator `%>%` from `magrittr`, it passes the left-hand side of the operator to the first argument of the right-hand side of the operator.

For example, `x %>% f(y)` converted into `f(x, y)` so the result from the left-hand side is then "piped" into the right-hand side. This pipe can be used to write multiple operations that you can read from left to right.

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

```
# Remove columns using select()
df2 <- df %>% select(-c(id, name, chapters))
```

```
# Output
# pages price
# 1 32 144
# 2 45 553
```

3.2 Remove Variables By Name Range

The same function can also be used to remove variables by name range.

```
# Remove columns by Range
df2 <-df %>% select(-(id:chapters))
df2
```

```
# Output
# price
```

1 144
2 553

3.3 Remove Variables using contains

You can use `-contains()` to ignore columns that contain text. The following example removes the column `chapters` as it contains text `apt`. This function also takes a list of values to check contains.

```
# Remove columns contains character
df2 <-df %>% select(-contains('apt'))
df2
```

Output
id pages name price
1 11 32 spark 144
2 22 45 python 553

3.4 Remove Column starts with

Similarly, you can use `-starts_with()` to ignore columns that start with a text. The following example removes the column `chapters` as it starts with character c.

```
# Remove columns starts with
df2 <-df %>% select(-starts_with('c'))
df2
```

Output
id pages name price
1 11 32 spark 144
2 22 45 python 553

3.5 Remove Column -ends_with()

Alternatively, you can use `-ends_with()` to remove variables that end with a text, the following examples remove `name` and `price` columns as they end with the letter e.

```
# Remove columns ends with
df2 <-df %>% select(-ends_with('e'))
```

```
df2
```

Output
id pages chapters
1 11 32 76
2 22 45 86

3.6 Remove Columns if it exists

Finally, you can use the `one_of()` function to check if the column exists and then remove it from the DataFrame only when it exists. If a column is not found, it returns a warning.

```
df2 <- df %>%
select(-one_of("name", "marks"))
```

4. Complete Example of Remove Columns in R

The following is a complete example of how to remove a single column/variable or several columns/variables from the R DataFrame (data.frame)

```
# 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))
```

# Display the dataframe
print(df)

# Remove Columns by Index
df2 <- df

# Remove Columns by Range
df2 <- df

# Remove Multiple columns
df2 <- df

Remove Columns in List

```
df2 <- df

# Remove using subset
df2 <- subset(df, select = -c(id, name, chapters))

# Load the dplyr package
library("dplyr")

# Remove columns using select()
df2 <- df %>% select(-c(id, name, chapters))

# Remove columns by Range
df2 <- df %>% select(-(id:chapters))

# Remove columns contains character
df2 <- df %>% select(-contains('apt'))

# Remove columns starts with
df2 <- df %>% select(-starts_with('c'))

# Remove columns ends with
df2 <- df %>% select(-ends_with('e'))

# Remove columns using within()
df2 <- within(df, rm(id, name, chapters))
```


Frequently Asked Questions on Remove Columns in R

How can I remove a specific column from a data frame?

You can use the R base notation to remove a specific column by its index. For example, to remove the second column of DataFrame you can use this syntax `df2 <- df`.

How do I remove multiple columns at once?

To remove multiple columns from DataFrame you can use the above approach. For example, you can use a vector to specify the indexes of columns which we want to remove from DataFrame. For example, `df2 <- df`.

How can I use dplyr package for removing columns?

The `dplyr` package provides various packages to remove columns from DataFrame. For example, using the `select()` function to remove specific columns and apply the negation on it to remove

those columns. For example, `library(dplyr)`

```
df2 <- subset(df, select = -c(id, name, chapters))
```

How do I remove columns and store the result in a new data frame?

To keep the original data frame unchanged and store the result in a new one, you can create a new data frame. For example, `new_df <- df`

Conclusion

In this article, you have learned different ways to remove a single column/variable and several columns/variables in the R DataFrame. The example includes removing columns by name, index, and from the list based on conditions, etc. Also learned how to use a select() function from the dplyr package.

Related Articles

Reference