

How can I drop columns by name in R, and what are some examples of doing so?

Authored by
stats writer

June 25, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I drop columns by name in R, and what are some examples of doing so?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=152863>

In R, dropping columns by name refers to the process of removing specific columns from a data frame or matrix. This can be done using the `subset` function, which allows for the selection of columns based on their names. For example, if a data frame has columns named "age", "gender", and "income", the following code can be used to drop the "age" and "income" columns:

```
subset(data_frame, select = -c(age, income))
```

Another way to drop columns by name is by using the `select` function from the `dplyr` package. This function allows for more flexibility in selecting columns, as it allows for the use of logical operators and wildcards. For instance, the following code will drop all columns that start with the letter "a" in a data frame:

```
select(data_frame, -starts_with("a"))
```

Additionally, the `subset` function can also be used in conjunction with logical operators to drop columns based on specific conditions. For example, the following code will drop all columns where the values are less than 10:

```
subset(data_frame, select = -(col1
```

Drop Columns by Name in R (With Examples)

There are three common ways to drop columns from a data frame in R by name:

Method 1: Use Base R

```
#drop col2 and col4 from data frame
```

```
df_new <- subset(df, select = -c(col2, col4))
```

Method 2: Use dplyr

```
library(dplyr)
```

```
#drop col2 and col4 from data frame  
df_new <- df %>% select(-c(col2, col4))
```

Method 3: Use data.table

```
library(data.table)
```

```
#convert data frame to data table
```

```
dt <- setDT(df)
```

```
#drop col2 and col4 from data frame
```

```
dt
```

The following examples show how to use each method in practice with the following data frame in R:

```
#create data frame
```

```
df <- data.frame(team=c('A', 'A', 'B', 'B', 'C', 'C', 'C', 'D'),  
points=c(12, 15, 22, 29, 35, 24, 11, 24),  
rebounds=c(10, 4, 4, 15, 14, 9, 12, 8),  
assists=c(7, 7, 5, 8, 19, 14, 11, 10))
```

```
#view data frame
```

```
df
```

team points rebounds assists

1 A 12 10 7

2 A 15 4 7

3 B 22 4 5

4 B 29 15 8

5 C 35 14 19

6 C 24 9 14

7 C 11 12 11

8 D 24 8 10

Example 1: Drop Columns by Name Using Base R

The following code shows how to drop the points and assists columns from the data frame by using the subset() function in base R:

```
#create new data frame by dropping points and assists  
columns
```

```
df_new <- subset(df, select = -c(points, assists))
```

```
#view new data frame
```

```
df_new
```

team rebounds

1 A 10

2 A 4

3 B 4

4 B 15

5 C 14

6 C 9

7 C 12

8 D 8

Notice that the points and assists columns have both been dropped from the new data frame.

Example 2: Drop Columns by Name Using dplyr

The following code shows how to drop the points and assists columns from the data frame by using the select() function in the dplyr package:

```
library(dplyr)
```

```
#create new data frame by dropping points and assists  
columns
```

```
df_new <- df %>% select(-c(points, assists))
```

```
#view new data frame
```

```
df_new
```

team rebounds

1 A 10

2 A 4

3 B 4

4 B 15

5 C 14

6 C 9

7 C 12

8 D 8

Notice that the points and assists columns have both been dropped from the new data frame.

Example 3: Drop Columns by Name Using data.table

The following code shows how to drop the points and assists columns from the data frame by setting both columns equal to NULL using the data.table package:

```
library(data.table)
```

```
#convert data frame to data table
```

```
dt <- setDT(df)
```

```
#drop points and assists columns
```

```
dt
```

```
#view updated data table
```

```
dt
```

```
team rebounds
```

```
1: A 10
```

```
2: A 4
```

```
3: B 4
```

```
4: B 15
```

```
5: C 14
```

```
6: C 9
```

```
7: C 12
```

```
8: D 8
```

Notice that the points and assists columns have both been dropped from the new data table.

Note: All three methods produce the same result, but the dplyr and data.table methods will tend to be faster when working with extremely large datasets.