

How can I drop columns from a dataset if the column name contains a specific string using R?

Authored by
stats writer

June 25, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I drop columns from a dataset if the column name contains a specific string using R?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=151578>

The process of dropping columns from a dataset in R can be achieved by using the "subset" function and specifying the condition that the column name must contain a specific string. This function allows for the selection of specific columns based on their names, and by using the "!" operator with the "grepl" function, the columns that do not contain the specified string can be dropped. This allows for a more efficient and targeted way of removing unwanted columns from a dataset.

R: Drop Columns if Name Contains Specific String

You can use the following methods to drop columns from a data frame in R whose name contains specific strings:

Method 1: Drop Columns if Name Contains Specific String

```
library(dplyr)
```

```
df_new <- df %>% select(-contains('this_string'))
```

Method 2: Drop Columns if Name Contains One of Several Specific Strings

```
library(dplyr)
```

```
df_new <- df %>% select(-contains(c('string1', 'string2', 'string3')))
```

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_name=c('A', 'B', 'C', 'D', 'E', 'F'),  
team_location=c('AU', 'AU', 'EU', 'EU', 'AU', 'EU'),  
player_name=c('Andy', 'Bob', 'Chad', 'Dan', 'Ed', 'Fran'),  
points=c(22, 29, 35, 30, 18, 12))
```

```
#view data frame
```

```
df
```

```
team_name team_location player_name points  
1 A AU Andy 22  
2 B AU Bob 29  
3 C EU Chad 35  
4 D EU Dan 30  
5 E AU Ed 18  
6 F EU Fran 12
```

Example 1: Drop Columns if Name Contains Specific String

We can use the following syntax to drop all columns in the data frame that contain 'team' anywhere in the column name:

```
library(dplyr)
```

```
#drop columns that contain 'team'
```

```
df_new <- df %>% select(-contains('team'))
```

```
#view new data frame
```

```
df_new
```

```
player_name points
```

```
1 Andy 22
```

```
2 Bob 29
```

```
3 Chad 35
```

```
4 Dan 30
```

```
5 Ed 18
```

```
6 Fran 12
```

Notice that both columns that contained 'team' in the name have been dropped from the data frame.

Example 2: Drop Columns if Name Contains One of Several Specific Strings

We can use the following syntax to drop all columns in the data frame that contain 'player' or 'points' anywhere in the column name:

```
#drop columns whose name contains 'player' or 'points'  
df_new <- df %>% select(-contains(c('player', 'points')))
```

```
#view new data frame
```

```
df
```

```
team_name team_location
```

```
1 A AU
```

```
2 B AU
```

```
3 C EU
```

```
4 D EU
```

```
5 E AU
```

```
6 F EU
```

Notice that both columns that contained either 'player' or 'points' in the name have been dropped from the data frame.

Note: You can find the complete documentation for the `dplyr select()` function .