

How can `gsub()` in R be used to replace multiple patterns simultaneously?

Authored by
stats writer

July 1, 2024

RECOMMENDED CITATION

stats writer (2024). *How can `gsub()` in R be used to replace multiple patterns simultaneously?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=164862>

The `gsub()` function in R can be used to replace multiple patterns simultaneously by specifying a vector of patterns and a vector of replacement values. This allows for efficient and streamlined replacement of multiple patterns in a single step. Additionally, the use of regular expressions in the patterns vector provides flexibility in targeting specific patterns to be replaced. This functionality of `gsub()` makes it a powerful tool for data manipulation and text processing in R.

Use `gsub()` in R to Replace Multiple Patterns

The function in R can be used to replace all occurrences of a certain pattern within a string in R.

To replace multiple patterns at once, you can use a nested `gsub()` statement:

```
df$col1 <- gsub('old1', 'new1',  
gsub('old2', 'new2',  
gsub('old3', 'new3', df$col1)))
```

However, a much faster method is the `stri_replace_all_regex()` function from the `stringi` package, which uses the following syntax:

```
library(stringi)
```

```
df$col1 <- stri_replace_all_regex(df$col1,  
pattern=c('old1', 'old2', 'old3'),  
replacement=c('new1', 'new2', 'new3'),
```

vectorize=FALSE)

The following examples show how to use each method in practice.

Method 1: Replace Multiple Patterns with Nested gsub()

Suppose we have the following data frame in R:

```
#create data frame
```

```
df <- data.frame(name=c('A', 'B', 'B', 'C', 'D', 'D'),  
points=c(24, 26, 28, 14, 19, 12))
```

```
#view data frame
```

```
df
```

```
name points
```

```
1 A 24
```

```
2 B 26
```

```
3 B 28
```

```
4 C 14
```

```
5 D 19
```

```
6 D 12
```

We can use a nested gsub() statement to replace multiple patterns in the name column:

#replace multiple patterns in name column

```
df$name <- gsub('A', 'Andy',  
gsub('B', 'Bob',  
gsub('C', 'Chad', df$name)))
```

#view updated data frame

df

name points

1 Andy 24

2 Bob 26

3 Bob 28

4 Chad 14

5 D 19

6 D 12

Notice that A, B, and C in the name column have all been replaced with new values.

Method 2: Replace Multiple Patterns with stringi

A much faster way to replace multiple patterns is by using the `stri_replace_all_regex()` function from the `stringi` package.

The following code shows how to use this function:

library(stringi)

```
#replace multiple patterns in name column  
df$name <- stri_replace_all_regex(df$name,  
pattern=c('A', 'B', 'C'),  
replacement=c('Andy', 'Bob', 'Chad'),  
vectorize=FALSE)
```

```
#view updated data frame
```

```
df
```

```
name points
```

```
1 Andy 24
```

```
2 Bob 26
```

```
3 Bob 28
```

```
4 Chad 14
```

```
5 D 19
```

```
6 D 12
```

Notice that the resulting data frame matches the one from the previous example.

Additional Resources