

How can I use `strsplit()` with multiple delimiters in order to rewrite a string?

Authored by
stats writer

June 24, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use `strsplit()` with multiple delimiters in order to rewrite a string?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=151224>

Strsplit() is a function in programming that allows for the splitting of a string into smaller substrings. By using multiple delimiters, it is possible to rewrite a string in a specific format. These delimiters act as separators, defining the boundaries between the substrings. This allows for a more precise and controlled way of manipulating a string, as opposed to simply splitting it at every character. By using multiple delimiters in strsplit(), it is possible to create a more organized and structured output, making it easier to work with and analyze the data contained in the string.

R: Use strsplit() with Multiple Delimiters

You can use the following basic syntax with the strsplit() function in R to split a string into pieces based on multiple delimiters:

```
strsplit(my_string , '+')
```

This particular example splits the string called my_string whenever it encounters one of the following three delimiters:

**A comma (,)
An ampersand (&)
A space**

Note that the characters inside the brackets indicate which delimiters to look for and the + sign indicates that there may be multiple delimiters in a row (e.g. there may be multiple spaces in a row).

The following example shows how to use this syntax in practice.

Example: Use strsplit() with Multiple Delimiters in R

Suppose we have the following string in R:

```
#create string  
my_string <- 'this is a, string & with seven words'
```

If we use the strsplit() function to split the string whenever a space is encountered, this will produce the following output:

```
#split string based on spaces  
strsplit(my_string , ' ' )  
  
]  
"this" "is" "a," "string" "&" "with" "" ""  
"seven" "words"
```

The strsplit() function splits the string whenever a space is encountered, but it is unable to handle commas, the ampersand, and multiple spaces.

To split the string based on each of these delimiters, we can use the following syntax:

```
#split string based on multiple delimiters
```

```
strsplit(my_string , '+')
```

```
]
```

```
"this" "is" "a" "string" "with" "seven" "words"
```

This function is able to split the string based on three different delimiters and correctly returns only the words in the string that we're interested in.

Note that in this example we included three delimiters within the brackets in the strsplit() function but you can specify as many delimiters as you'd like.