How can I use the str_replace() function to replace matched patterns in a string with a new value?

Authored by

stats writer

June 24, 2024

The str_replace() function is a useful tool for replacing specific patterns within a string with a new value. By specifying the pattern to be replaced and the desired replacement value, this function allows for efficient and accurate manipulation of strings. This can be especially helpful in tasks such as data cleaning or formatting text. By utilizing the str_replace() function, users can easily make changes to their strings without having to manually search and replace each instance. This function is available in many programming languages and can be a valuable tool for any developer or user working with strings.

R `str_replace()` and `str_replace_all()` are used to replace values of a string column based on matched patterns (pattern matching with regex - regular expression), also used to replace with a specific string or character value. This function can be used on both DataFrame columns and a vector.

In order to use this `str_replace()` method, first, you need to load its library using `library("stringr")`. In case you don't have this package, install it using `install.packages("stringr")`. The `stringr` package provides a set of functions to work with strings as easily as possible.

1. str_replace() Syntax

Following is a syntax of `str_replace()` method from `stringr` package.

```
# Syntax of str_replace()
str_replace(string, pattern, replacement)
str_replace_all(string, pattern, replacement)
```

Let's create an R dataframe and explore some examples using `str_replace()` method.

```
# Create DataFrame
df <- data.frame(id=c(1,2,3,4),
address=c('Orange St','Anton Blvd','Jefferson Pkwy','Main St'))
print(df)
```

#Output
id address
#1 1 Orange St
#2 2 Anton Blvd
#3 3 Jefferson Pkwy
#4 4 Main St

2. Use str_replace() to Replace Part of String with Another String

`str_replace()` method from `stringr` package is used to replace a part of a string on column with another string or replace column with pattern matching. The following example replaces string `St` with `Street` on column `address`. Here, `df$address` is a vector. Note that every column in a DataFrame is a vector.

If you wanted to replace NA with Empty String in R dataframe use methods from `dplyr` package.

```
# Replace String with anotehr String
library(stringr)
df$address <- str_replace(df$address, "St", "Street")
print(df)
```

Yields below output.

```
# Output
id address
1 1 Orange Street
2 2 Anton Blvd
3 3 Jefferson Pkwy
4 4 Main Street
```

3. str_replace_all() to Match on Multiple Strings

Use `str_replace_all()` method of `stringr` package to replace multiple string values at a time on a single column. The following example takes vector `c()` with mapping of values to be replaced on `address` column.

```
# Replace multiple strings at a time
rep_str = c('St'='Street','Blvd'='Boulevard','Pkwy'='Parkway')
df$address <- str_replace_all(df$address, rep_str)
print(df)
```

Yields below output

```
# Output
```

```
id address
1 1 Orange Street
2 2 Anton Boulevard
3 3 Jefferson Parkway
4 4 Main Street
```

4. Replace With Pattern Matching (regex)

Since these methods are used on vector, let's create a R vector and replace values in it with pattern matching. Since every column in a DataFrame is a vector, you can also use pattern matching on DataFrame columns.

```
# Create Vector
numbers <- c("one", "two", "three")
num <- str_replace(numbers, "", "-")
print(num)
```

Output
"-ne" "tw-" "thr-e"

Now let's use str_replace_all() on the same example.

```
# Create Vector
numbers <- c("one", "two", "three")
num2 <- str_replace_all(numbers, "", "-")
print(num2)
```

Output
"-n-" "tw-" "thr--"

5. Conclusion

In this article, you have learned str_replace() and str_replace_all() methods from sringr package are used to replace part of a string from a column to another string. Also, these methods are used to replace the string with another string based on pattern matching.

Related Articles

References