

Stack Overflow is a community of 4.7 million programmers, just like you, helping each other.

Join them; it only takes a minute:

Sign up

Join the Stack Overflow community to:

Ask programming questions

Answer and help your peers

Get recognized for your expertise

Replacement and non-matches with 'sub'

Months ago I ended up with a sub statement that originally worked with my input data. It has since stopped working causing me to re-examine my ugly process. I hate to share it but it accomplished several things at once:

```
active$id[grepl("CIR",active$description)] <- sub(".*CIR0*(\\d+).*", "\\1", active$description[grepl("CIR",active$description)], perl=TRUE)
```

This statement created a new id column by finding rows that had an id embedded in the description column. The sub statement would find the number following a "CIR0" and populate the id column iff there was an id within a row's description. I recognize it is inefficient with the embedded grepl subsetting either side of the assignment.

Is there a way to have a 'sub' replacement be NA or empty if the pattern does not match? I feel like I'm missing something **very** simple but ask for the community's assistance. Thank you.

Example with the results of creating an id column:

name	id	description
a	343	Here is CIR00343
b		Didn't have it
c	123	What is CIR0123
d		CIR lacks a digit
e	452	CIR452 is next

r

edited Mar 9 '12 at 21:48

asked Mar 9 '12 at 21:07

**jed**
60 6

2 It would be nice if you gave some sample data and output! That way we could verify what you need. – [nogradpes](#) Mar 9 '12 at 21:12

I convinced myself the question was general enough that sample data might be more confusing but you're right - I've added a short table. – [jed](#) Mar 9 '12 at 21:49

2 Answers

I was struggling with the same issue a few weeks ago. I ended up using the `str_match` function from the `stringr` package. It returns NA if the target string is not found. Just make sure you subset the result correctly. An example:

```
library(stringr)
str = "Little_Red_Riding_Hood"
sub(".*(Little).*", "\\1", str) # Returns 'Little'
sub(".*(Big).*", "\\1", str) # Returns 'Little_Red_Riding_Hood'
str_match(str, ".*(Little).*")[1,2] #Returns 'Little'
str_match(str, ".*(Big).*")[1,2] # Returns NA
```

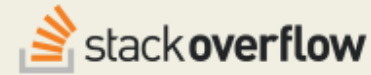
edited Mar 9 '12 at 22:02

answered Mar 9 '12 at 21:57

**Jesse Anderson**
2,580 12 26

Thanks for introducing me to the `stringr` package. Using it I've been able to clean up the process and capture a side case I hadn't anticipated. – [jed](#) Mar 9 '12 at 23:11

Work on work you love. From home.



I think in this case you could try using `ifelse()` , i.e.,

```
active$id[grepl("CIR",active$description)] <- ifelse(match, replacement, "")
```

where `match` should evaluate to true if there's a match, and `replacement` is what that element would be replaced with in that case. Likewise, if `match` evaluates to false, that element's replaced with an empty string (or `NA` if you prefer).

answered Mar 9 '12 at 21:19



[a barking spider](#)

300 1 9

Disclaimer: I haven't used the `grep/regex` functionality in R all that much, so this might not be the best solution, but I'm just throwing it out there - `ifelse()` has saved my behind in so many situations that I've lost count. — [a barking spider](#) Mar 9 '12 at 21:21

Thanks, I hadn't considered `ifelse()` in this case but tried to work out a way to leverage it after your answer. I run into difficulty capturing the pattern from the test without duplicating the process as part of the replacement. I'll give it some more thought. — [jed](#) Mar 9 '12 at 23:17
