Replacing Matches

Lauren Kapraun

10/03/18

Document Setup

```
knitr::opts_chunk$set(echo = TRUE)
# The original characters.csv file can be found here:
# https://github.com/efekarakus/potter-network/tree/master/data
potter = read.csv(file = "characters.csv", header = TRUE, stringsAsFactors=FALSE)
library(tidyverse)
library(stringr)
library(tibble)
```

Replace Matched Patterns in a String

`str_replace()` and `str_replace_all()` allow you to replace matches with new strings.

`str_replace()` replaces the first matched pattern in each string, while `str_replace_all()` replaces all matched patterns in each string.

Arguments

The structure inside the parentheses for `str_replace()` **AND** `str_replace_all()` is: **(string, "[pattern]", "replacement")**

The **string** argument is either a character vector or something that can be formatted as a character vector.

The **pattern** argument includes letters that need to be found within a string.

The **replacement** argument is a character vector of replacements.

```
# Display Original Text
potter[1,3]

# Replace first occurance
str_replace(potter[1,3], "[e]", "*")

# Replace all occurances
str_replace_all(potter[1,3], "[eE]", "*")
```

```
## [1] "Brother of Sirius. Used to be a Death Eater but defected."
```

```
## [1] "Broth*r of Sirius. Used to be a Death Eater but defected."
```

```
## [1] "Broth*r of Sirius. Us*d to b* a D*ath *at*r but d*f*ct*d."
```

Multiple Replacements using str_replace_all()

With `str_replace_all()` you can perform multiple replacements by supplying a named vector. In the example below, the `str_replace_all()` is combined with the `c` function to list out all of the text to be replaced.

The structure for this command is: `str_replace_all(string, c("pattern" = "replacement"))`

```
# Display original text
potter[56,3]

# Replacement values
str_replace_all(potter[56,3], c("Marries" = "Ginny marries",
                                "and" = "&",
                                "& only" = "& is the only",
                                "Molly & Arthur" = "Molly & Arthur Weasley"))
```

```
## [1] "Marries Harry Potter and only daughter of Molly and Arthur."
```

```
## [1] "Ginny marries Harry Potter & is the only daughter of Molly & Arthur Weasley."
```

Backreferences

Instead of replacing with a fixed string you can use backreferences to insert components of the match. In the following code, I flipped the order of the first three words:

```
# Display original text
potter [1, 3]
potter [2, 3]
potter [3, 3]

# Updated text
potter [, 3] %>%
  str_replace("([^ ]+) ([^ ]+) ([^ ]+)", "\\3 \\2 \\1") %>%
  head(3)
```

```
## [1] "Brother of Sirius. Used to be a Death Eater but defected."
```

```
## [1] "Best friend of James Potter and godfather of Harry."
```

```
## [1] "Killed by a werewolf. She was a gryffindor student who dated Ron. "
```

```
## [1] "Sirius. of Brother Used to be a Death Eater but defected."
## [2] "of friend Best James Potter and godfather of Harry."
## [3] "a by Killed werewolf. She was a gryffindor student who dated Ron. "
```