

# Homework 7

201982188 통계학과 박현주

2020-11-12

## Your turn - HW1.

### 1-1.

Replace all forward slashes in a string with backslashes. (Change “past/present/future” into “past”)

```
prob1 <- 'past/present/future'
cat(str_replace_all(prob1, '\\/', '\\\\\\\\'))

## past\\present\\future
```

### 1-2.

Implement a simple version of `str_to_lower()` using `replace_all()`. Use data words.

```
pattern <- as.vector(letters)
names(pattern) <- LETTERS
new_words <- str_to_upper(words) #words 모두 대문자로 바꾸기
cat(str_to_upper(words)[1:5], '=>', str_replace_all(new_words, pattern = pattern))

## A ABLE ABOUT ABSOLUTE ACCEPT => a able about absolute accept
```

### 1-3.

Switch the first and last letters in words. Which of those strings are still words?

```
x <- 'icecream'

word_func <- function(x){
  split_word <- str_split(x, '')[[1]]
  word_len <- split_word %>% length()
  new_word <- split_word[word_len:1] %>% paste(., collapse = '', sep='')
  return(new_word)
}

new_word <- words %>% lapply(., word_func) %>% unlist()
words[which(words == new_word)] %>% cat()

## a dad eye level non refer
```

## Your turn - HW2.

## 2-1.

Split up a string like “apples, pears, and bananas” into individual components.

```
prob2 <- 'apples, pears, and bananas'
#1.
str_split(prob2, "\\s(and\\s)?")[[1]] %>% cat('1. ', .., '\n')

## 1.  apples pears bananas

#2.
str_split(prob2, ', (and )?')[[1]] %>% cat('2. ', .., '\n')

## 2.  apples pears bananas
```

## 2-2.

Why is it better to split up by boundary(“word”) than “ ”? Use the following sentence

```
sentence <- "The quick (\"brown\") fox can't jump 32.3 feet, right?"
str_split(sentence, " ")[[1]]

## [1] "The"          "quick"        "(\"brown\")"  "fox"          "can't"
## [6] "jump"         "32.3"         "feet,"       "right?"

str_split(sentence, boundary("word"))[[1]]

## [1] "The"      "quick"    "brown"    "fox"      "can't"    "jump"     "32.3"     "feet"
## [9] "right"
```


boundary(“word”)를 사용하면 특수 기호는 제외하고 단어만 뽑아낸다. “ ”를 사용하면 space를 기준으로 분리하기 때문에 특수 기호도 포함된다.

## 2-3.

What does splitting with an empty string ("" ) do? Experiment, and then read the documentation.

```
str_split("What does splitting with an empty string ('')", "")[[1]]

## [1] "W" "h" "a" "t" " " "d" "o" "e" "s" " " "s" "p" "l" "i" "t" "t" "i" "n"
## [20] " " "w" "i" "t" "h" " " "a" "n" " " "e" "m" "p" "t" "y" " " "s" "t" "r"
## [39] "n" "g" " " " (" "" "" "" " )"
```



”: 글자 하나하나 분리한다.