```
############################################
### TextMining Lecture 05              #####
### Subject: Text Preprocessing I      #####
### Developed by. KKIM                 #####
############################################

library(dplyr)
library(magrittr)
library(stringr)

women %>% head(1)
women %>%
  head(1) %>%      nrow

str_view_all("Text-Mining", "ing",
              html=TRUE)

### Regular Expression I
# ^: starting point of text
text.1 <- "Pep Guardiola expressed his admiration for Erling
Haaland’ s attitude after winning Man of the Match performance
against Manchester United"
vector.1 <- c("I like to drink Pepsi","I love Manchester
United",
              "I have iphone","iphone is very good")

str_view_all(text.1, "^")
str_view_all(text.1, "^p")
str_view_all(text.1, "^pep")
str_view_all(text.1, "^Pep")

str_view_all(vector.1, "^")
str_view_all(vector.1, "^I")
str_view_all(vector.1, "^i")

# $: end point of text
str_view_all(text.1, "$")

str_view_all(text.1, "d$")
str_view_all(vector.1, "d$")

# .: Only one letter to the left and right
str_view_all(text.1, ".pres.")
str_view_all(vector.1, ".p.")
str_view_all(vector.1, ".pho.")

### Regular Expression II
car <- c('car','cr','caar','caaar','caaaar')
apple <- c('apple','applee','aple','appple','apleee')

# ?: match-zero-or-one
str_view_all(car, "ca?r")
str_view_all(apple, "ap?le")
str_view_all('ale', "ap?le")

# *: match-zero-or-more
str_view_all(car, "ca*r",
             html=TRUE)
str_view_all(car, "ch*r",
             html=TRUE)
str_view_all(apple, "ap*le",
             html=TRUE)

# +: match-one-or-more
str_view_all(car, "ca+r",
             html=TRUE)
str_view_all(apple, "ap+le",
             html=TRUE)
str_view_all('ale', "ap+le",
             html=TRUE)

### Regular Expression III
# []
str_view_all(text.1, "[a-e]")
str_view_all(text.1, "a-e")

str_view_all(vector.1, "[a-e]")
str_view_all(vector.1, "a-e")

str_view_all(text.1, "[a-e]")
str_view_all(text.1, "[a|e]")
str_view_all(text.1, "[ae]")

str_view_all(vector.1, "[a-e]",
             html=TRUE)
str_view_all(vector.1, "[a|e]",
             html=TRUE)

# [] + ^
str_view_all(text.1, "[^a-z]",
             html=TRUE)
str_view_all(text.1, "[^a-zA-Z]",
             html=TRUE)

str_view_all(vector.1, "[A-Z]",
             html=TRUE)
str_view_all(vector.1, "[^A-Z]",
             html=TRUE)

##### Exercise
sample <- "abc ABC 123.!?₩₩(){}₩n abcde aaa bacad .a.aa.aaa
abbaab ababcbabcdcbabcde"
sample.vec <- c("abc ABC 123₩t.!?₩₩(){}₩n abcde",
                "aaa bacad .a.aa.aaa",
                "abbaab ababcbabcdcbabcde")

str_view_all(sample, "ab?c",
             html=TRUE)
# str_view_all(sample.vec, "ab?c")

str_view_all(sample, "a|c",
             html=TRUE)
str_view_all(sample, "[a|c]",
             html=TRUE)

str_view_all(sample, "[a-c]",
             html=TRUE)

# match-one-or-more
str_view_all(sample, "ac+",
             html=TRUE)

# match-zero-or-more
str_view_all(sample, "ac*",
             html=TRUE)

str_view_all(sample, "[ab]",
             html=TRUE)
str_view_all(sample, "[a|b]",
             html=TRUE)

str_view_all(sample, "[ab]c",
             html=TRUE)
str_view_all(sample, "[a|b]c",
             html=TRUE)

str_view_all(sample, "[^ab]",
             html=TRUE)

str_view_all(sample, "a{1,2}",
             html=TRUE)

### Regular Expression IV
vector.1 %>%
  str_view_all("[:lower:]",
               html=TRUE)
vector.1 %>%
  str_view_all("[[:lower:]]",
               html=TRUE)

text.1 %>%
  str_view_all("[:lower:]",
               html=TRUE)
text.1 %>%
  str_view_all("[[:lower:]]",
               html=TRUE)

vector.1 %>%
  str_view_all("[:upper:]",
               html=TRUE)
vector.1 %>%
  str_view_all("[[:upper:]]",
               html=TRUE)

vector.1 %>%
  str_view_all("[:alpha:]",
               html=TRUE)
vector.1 %>%
  str_view_all("[[:alpha:]]",
               html=TRUE)

c("one 2 3 four 5 six") %>%
  str_view_all("[:digit:]",
               html=TRUE)
c("one 2 3 four 5 six") %>%
  str_view_all("[:xdigit:]",
               html=TRUE)

c("one 2 3 four 5     six
  7 delapan nine 10") %>%
  str_view_all("[:blank:]",
               html=TRUE)
c("one 2 3 four 5     six
  7 delapan nine 10") %>%
  str_view_all("[:space:]",
               html=TRUE)

c("'I love you.' heheh!@#") %>%
  str_view_all("[:punct:]",
               html=TRUE)
c("'I love you.' heheh!@#") %>%
  str_view_all("[:alnum:]",
               html=TRUE)
c("'I love you.' heheh!@#") %>%
  str_view_all("[:print:]",
               html=TRUE)
c("'I love you.' heheh!@#") %>%
  str_view_all("[:graph:]",
               html=TRUE)

### Regular Expression V
c("'I love you.' heheh!@#") %>%
  str_view_all("[:punct:]{1}",
               html=TRUE)
c("'I love you.' heheh!@#") %>%
  str_view_all("[:punct:]",
               html=TRUE)
c("'I love you.' heheh!@#") %>%
  str_view_all("[:punct:]{2}",
               html=TRUE)
c("'I love you.' heheh!@#") %>%
  str_view_all("[:punct:]{3}",
               html=TRUE)

c("'I love you.' heheh!@#") %>%
  str_view_all("[:alnum:]{1,}",
               html=TRUE)
c("'I love you.' heheh!@#") %>%
  str_view_all("[:alnum:]{3,}",
               html=TRUE)
c("'I love you.' heheh!@#") %>%
  str_view_all("[:alnum:]{4,}",
               html=TRUE)

c("'I love you.' heheh!@#") %>%
  str_view_all("[:alnum:]{1,1}",
               html=TRUE)
c("'I love you.' heheh!@#") %>%
  str_view_all("[:alnum:]{1,3}",
               html=TRUE)
c("'I love you.' heheh!@#") %>%
  str_view_all("[:alnum:]{2,2}",
               html=TRUE)

c("'I love you.' heheh!@#") %>%
  str_view_all("₩₩s",
               html=TRUE)
c("'I love you.' heheh!@#") %>%
  str_view_all("₩₩S",
               html=TRUE)
c("'I love you.' heheh!@#") %>%
  str_view_all("₩₩w",
               html=TRUE)
c("'I love you.' heheh!@#") %>%
  str_view_all("₩₩W",
               html=TRUE)
c("'I love you.' heheh!@#") %>%
  str_view_all("₩₩d",
               html=TRUE)
c("'I love you.' heheh!@#") %>%
  str_view_all("₩₩D",
               html=TRUE)



### Exercise
library("officer")
trump.speech <-
  read_docx("R file/R file_LEC04/Trump_2021_final_speech.docx")
```

```
trump.speech.sum <-
  trump.speech %>%
  docx_summary

# From Trump’ s speech, find texts that are not spoken by the
person
# [Crowd chants: “We love you” ], [Melania Trump], [Donald
Trump]
trump.speech.sum$text %>%
  str_extract('₩₩[[:print:]+₩₩]')
trump.speech.sum$text %>%
  str_extract('₩₩[[[:print:]]+₩₩]')

# Steps
trump.speech.sum$text %>%
  str_extract('₩₩[')
trump.speech.sum$text %>%
  str_extract('₩₩[[:alpha:]')
trump.speech.sum$text %>%
  str_extract('₩₩[[:alpha:]+')
trump.speech.sum$text %>%
  str_extract('₩₩[[:alpha:]{1,}')

@ Question
"[Donald Trump]" %>%
  str_extract('₩₩[[:print:]+₩₩]')
"[Donald Trump]" %>%
  str_extract('₩₩[+₩₩]')

"[[[[[[[Donald Trump]]]]]]]" %>%
  str_extract('₩₩[+')
"[[[[[[[Donald Trump]]]]]]]" %>%
  str_extract('₩₩]+')

c("caaaabr", "caar","ca!r") %>%
  str_extract('ca*r')
c("caaaabr", "caar","ca!r") %>%
  str_extract('ca*!')

# Write down the desired result
c("[DonaldTrump]","[Donald Trump]","[!@#Donald Trump]") %>%
  str_extract('[:punct:]+')
c("[DonaldTrump]","[Donald Trump]","[!@#Donald Trump]") %>%
  str_extract('[:punct:]{1,}')

c("[DonaldTrump]","[Donald Trump]","[!@#Donald Trump]") %>%
  str_extract('₩₩[[:punct:]+')
c("[DonaldTrump]","[Donald Trump]","[!@#Donald Trump]") %>%
  str_extract('₩₩[[:punct:]{1,}')

c("[DonaldTrump]","[Donald Trump]","[!@#Donald Trump]") %>%
  str_extract('₩₩[[:alpha:]+₩₩]')

c("[DonaldTrump]","[Donald Trump]","[!@#Donald Trump]") %>%
  str_extract('₩₩[[:print:]+₩₩]')


### Load text data with pdf
# https://github.com/ujjwalkarn/DataScienceR

library('pdftools')

mat10 <-
  pdf_text("R data/Matthew10.pdf")
mat10
length(mat10)

### Text Mining Web Data
# Load text data from web
library("rvest")
url <- 'https://news.naver.com/'
html <- read_html(url)
html

html %>%
  html_nodes("div.cjs_t")

html %>%
  html_nodes("div.cjs_t") %>%
  html_text()

html %>%
  html_nodes("div.cjs_age_name")

html %>%
html_nodes("div.cjs_age_name") %>%
html_text()
```