Ch9. String manipulation

Jongrak Jeong

'2023-01-18

```
setwd("~/Library/Mobile Documents/com~apple~CloudDocs/Study/2_Data Science/Practice/R Programming by He
```

1. grep() and nchar()

grep(pattern, x): find where there is the pattern

```
lyrics <- scan("yesterday.txt", what = "")
str(lyrics)
```

```
##  chr [1:126] "Yesterday," "all" "my" "troubles" "seemed" "so" "far" "away." ...
```

```
grep("Yesterday", lyrics)
```

```
## [1]   1  63 105
```

```
grep("yesterday", lyrics)
```

```
## [1]  22  40  62  84 104 126
```

```
grep("\\?", lyrics)
```

```
## [1] 47 89
```

```
lyrics[47]
```

```
## [1] "go?"
```

```
lyrics[89]
```

```
## [1] "go?"
```

nchar(x): length of string x

```
nchar("yesterday")
```

```
## [1] 9
```

```
nchar("John Lennon")
```

```
## [1] 11
```

```
nchar(lyrics)
```

```
##   [1] 10  3  2  8  6  2  3  5  3  2  5  2  6  7  4  2  5  3  1  7  2 10  9  3  3
##  [26]  4  3  3  1  4  2  3  7  1  6  7  4  3  3  9  4  9  3  3  3  2  3  1  5  5
##  [51]  3  8  4  1  4  9  6  3  1  4  3 10  9  4  3  4  2  4  4  2  5  3  1  4  1
##  [76]  5  2  4  5  3  1  7  2 10  3  3  3  2  3  1  5  5  3  8  4  1  4  9  6  3
```

```
## [101]  1  4  3 10  9  4  3  4  2  4  4  2  5  3  1  4  1  5  2  4  5  3  1  7  2
## [126] 10
```

2. paste(), substr(), and strsplit()

paste(..., sep = " ")

```
paste("John", "Lennon", sep = " ")
```

```
## [1] "John Lennon"
```

```
paste("John", "Lennon", sep = "")
```

```
## [1] "JohnLennon"
```

```
paste(2016, 01, 19, sep = "-")
```

```
## [1] "2016-1-19"
```

```
paste(2016, "01", 19, sep = "-")
```

```
## [1] "2016-01-19"
```

substr(x, star, stop): extract strings from 'start' to 'stop'

```
substr("20160119", 7, 8)
```

```
## [1] "19"
```

```
substr("20160119", 5, 8)
```

```
## [1] "0119"
```

```
substr("19Jan2016", 3, 5)
```

```
## [1] "Jan"
```

strsplit(x, split)

```
strsplit("2016-01-19", "-")
```

```
## [[1]]
## [1] "2016" "01"   "19"
```

```
unlist(strsplit("2016-01-19", "-"))
```

```
## [1] "2016" "01"   "19"
```

```
strsplit(c("2016-01-19", "2016-04-03"), "-")
```

```
## [[1]]
## [1] "2016" "01"   "19"
##
## [[2]]
## [1] "2016" "04"   "03"
```

3. gregexpr(), gsub(), LETTERS, letters

gregexpr()

gregexpr(pattern, x): find all the place the pattern is found

```
gregexpr("-", "2016-01-19")
```

```
## [[1]]
## [1] 5 8
## attr(,"match.length")
## [1] 1 1
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
```

```
unlist(gregexpr("-", "2016-01-19"))
```

```
## [1] 5 8
```

gsub()

gsub(pattern, replace, x): pattern of x > replace it with another pattern

```
gsub("-", ".", "2016-01-19")
```

```
## [1] "2016.01.19"
```

```
head(gsub("Yesterday", "yesterday", lyrics), 5)
```

```
## [1] "yesterday," "all"        "my"         "troubles"   "seemed"
```

```
head(gsub(",", "", lyrics), 5)
```

```
## [1] "Yesterday" "all"       "my"        "troubles"  "seemed"
```

LETTERS and letters

```
LETTERS
```

```
##  [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S"
## [20] "T" "U" "V" "W" "X" "Y" "Z"
```

```
letters
```

```
##  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
## [20] "t" "u" "v" "w" "x" "y" "z"
```

tolower(): upper > lower and toupper(): lower > upper

```
tolower("Yesterday")
```

```
## [1] "yesterday"
```

```
tolower("John Lennon")
```

```
## [1] "john lennon"
```

```
toupper("yesterday")
```

```
## [1] "YESTERDAY"
```

```
toupper("John Lennon")
```

```
## [1] "JOHN LENNON"
```

4. regular expression

"." means arbitary character

| ( ) [{ ^ $ * + ?

ex1.

```
grep("y", lyrics, ignore.case = T) # "my", "Why" "easy" "Why" "easy"
```

```
##  [1]   1   3   8  14  17  22  23  40  42  43  53  62  63  68  71  79  84  85  95
## [20] 104 105 110 113 121 126
```

```
grep("y.", lyrics, ignore.case = T)
```

```
##  [1]   1   8  14  17  22  23  40  42  53  62  63  71  79  84  95 104 105 113 121
## [20] 126
```

```
A <- grep("y", lyrics, ignore.case = T)
A.1 <- grep("y.", lyrics, ignore.case = T)

A %in% A.1
```

```
##  [1]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
## [13]  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
## [25]  TRUE
```

```
! (A %in% A.1)
```

```
##  [1] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
## [13] FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
## [25] FALSE
```

```
A[! (A %in% A.1)]
```

```
## [1]   3  43  68  85 110
```

```
index <- rep(F, length(lyrics))
index[A[! (A %in% A.1)]] <- T
subset(lyrics, index)
```

```
## [1] "my"   "Why"  "easy" "Why"  "easy"
```

ex2. find ".r" files

```
filelist <- c("yesterday.txt", "yesterday.r", "exercise.R", "graph.jpg", "output.dat")
strsplit(filelist, ".", fixed = T)
```

```
## [[1]]
## [1] "yesterday" "txt"
```

```
##
## [[2]]
## [1] "yesterday" "r"
##
## [[3]]
## [1] "exercise" "R"
##
## [[4]]
## [1] "graph" "jpg"
##
## [[5]]
## [1] "output" "dat"
```

```
sapply(strsplit(filelist, ".", fixed = T), "[", 2)
```

```
## [1] "txt" "r"   "R"   "jpg" "dat"
```

```
grep("r", sapply(strsplit(filelist, ".", fixed = T), "[", 2))
```

```
## [1] 2
```

```
grep("r", sapply(strsplit(filelist, ".", fixed = T), "[", 2), ignore.case = T)
```

```
## [1] 2 3
```

ex3. find the files whose names have some digits

```
filelist.2 <- c("survey_1.txt", "exam final.hwp", "records 21.sav", "graph.jpg", "20160405")
grep("\\d", filelist.2)
```

```
## [1] 1 3 5
```

ex4. find the files whose names have some spaces

```
grep("\\s", filelist.2)
```

```
## [1] 2 3
```

ex5. find the files whose names have some underlines

```
grep("\\u", filelist.2)
```

```
## [1] 1
```

5. application: text visuallization - word cloud

load the text data

```
lyrics <- scan("yesterday.txt", what = "character")
str(lyrics)
```

```
##  chr [1:126] "Yesterday," "all" "my" "troubles" "seemed" "so" "far" "away." ...
```

```
head(lyrics)
```

```
## [1] "Yesterday," "all"        "my"         "troubles"   "seemed"
```

```
## [6] "so"
```

remove seperators

```
lyrics.1 <- gsub(",", "", lyrics)
lyrics.1 <- gsub("\\.", "", lyrics.1)
lyrics.1 <- gsub("\\!", "", lyrics.1)
lyrics.1 <- gsub("\\?", "", lyrics.1)
```

convert upper letters to lower letters

```
# for
for (j in 1:26)
  lyrics.1 <- gsub(LETTERS[j], letters[j], lyrics.1)

# letters
lyrics.1 <- tolower(lyrics.1)
```

check

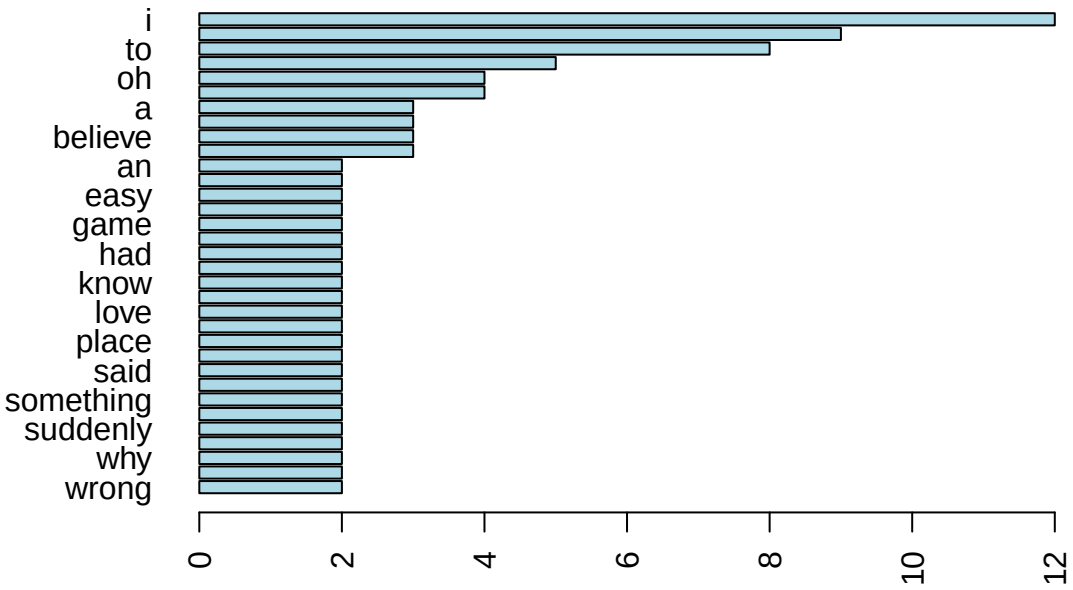
```
head(cbind(lyrics, lyrics.1), 9)
```

```
##       lyrics       lyrics.1
##  [1,] "Yesterday," "yesterday"
##  [2,] "all"        "all"
##  [3,] "my"         "my"
##  [4,] "troubles"   "troubles"
##  [5,] "seemed"     "seemed"
##  [6,] "so"         "so"
##  [7,] "far"        "far"
##  [8,] "away."      "away"
##  [9,] "Now"        "now"
```

frequency and bar plot

```
tab.1 <- table(lyrics.1)
tab.2 <- sort(tab.1, decreasing = TRUE)
tab.2a <- tab.2[tab.2 > 1]
```

```
par(mar = c(4, 6, 5, 4))
barplot(rev(tab.2a), horiz = TRUE, las = 2,
        main = "Beatles' Yesterday", col = "lightblue")
```



word cloud

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
par(mar = c(2, 2, 2, 2))
set.seed(12345)
wordcloud(words = names(tab.1), freq = tab.1, scale = c(5, 0.5),
          min.freq = 1, colors = rainbow(10), random.color = FALSE,
          random.order = FALSE, rot.per = 0.25)
```