# HW5

2019150445/Shin Baek Rok

2020 12 4

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## Warning: package 'tidyr' was built under R version 4.0.3
```

```
## Warning: package 'readr' was built under R version 4.0.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(stringr)
```

## 2.

### a)

**1)** Only one backslash: Escape the next character in R strings. Two backslash: Escape the next character in R regex. Three backslash: First two backslash means just backslash in regex, and third backslash escapes the next character.

**2)** "’\\ will work since first backslash works as escape symbol to escape ’, similarly second backslash works as escape symbol to escape ", and last four backslashes indicate just one backslash.

**3)** In regex, dot means any character. To match the character ‘.’, we have to use . . Thus ...... means dot,any character,dot,any character,dot,any character. For example, .x.y.z

### b)

```
#1.
’^[aeiouAEIOU]’
```

**1**

```
## [1] "^[aeiouAEIOU]"
```

```
#2.
’[^aeiouAEIOU]’
```

```
## [1] "[^aeiouAEIOU]"
```

```
#3.
’[^e]ed$’
```

```
## [1] "[^e]ed$"
```

```
#4.
’(ing|ise)$’
```

```
## [1] "(ing|ise)$"
```

```
str_subset(words, "(cei|[^c]ie)")
```

**2)**

```
##  [1] "achieve"    "believe"    "brief"      "client"     "die"
##  [6] "experience" "field"      "friend"     "lie"        "piece"
## [11] "quiet"      "receive"    "tie"        "view"
```

```
str_subset(words, "[^c]ei")#exception
```

```
## [1] "weigh"
```

```
str_subset(words,’qu’)
```

**3)**

```
##  [1] "equal"    "quality"  "quarter"  "question" "quick"    "quid"
##  [7] "quiet"    "quite"    "require"  "square"
```

```
str_subset(words,'q[^u]') #None
```

```
## character(0)
```

**4)** For example, word summarise is the British English definition of summarize. We can match this figure by ‘ise$’

**5)** ‘^010\-?\d{4}\-?\d{4}$’ will work well.

```
s<-c('010-7173-2932','010-234-gsd9','010-3232-233','010-23232-2333','01017382928')
str_subset(s,'^010\\-?\\d{4}\\-?\\d{4}$')
```

```
## [1] "010-7173-2932" "01017382928"
```

**c)**

**1)** ? means 0 or 1, that is {0,1} + means 1 or more, that is {1,} * means 0 or more, that is {0,}

```

# 1. ^.*$ will match any string since . means any character and * means 0 or more.

# 2. '\\{.+\\}' will match at least one character that is enclosed in parentheses

# 3. \d{4}-\d{2}-\d{2} will match 0000-00-00 where 0 can be replaced in 0 to 9

# 4. "\\\\{4}" will match \\\\\\\\ since \\ means \ and {4} means repeat 4 times.
```

**2)**

**d)**

**1)**

1.

```
pattern<-"^x|x$"
str_subset(words,pattern)
```

```
## [1] "box" "sex" "six" "tax"
```

```
#or
words[str_detect(words,pattern)]
```

```
## [1] "box" "sex" "six" "tax"
```

2.

```
pattern<-"^[aeiouAEIOU]|[^aeiouAEIOU]$"
str_subset(words,pattern) %>% head()
```

```
## [1] "a"        "able"     "about"    "absolute" "accept"   "account"
```

3.

```
pattern<-"([aeiouAEIOU])"
str_subset(words,pattern) %>% str_replace_all("[aeiouAEIOU]","") %>% str_subset(’[aeiouAEIOU]’) #None
```

```
## character(0)
```

## 3.

```
library(gutenbergr)
```

```
## Warning: package ’gutenbergr’ was built under R version 4.0.3
```

### 1)

```
x<-gutenberg_metadata
x$gutenberg_id[x$title %>% str_detect(’Pride and Prejudice’) ] %>% na.omit() %>% .[1:6]
```

```
## [1]  1342 20686 20687 26301 37431 42671
```

### 2)

```
gutenberg_works(languages=’en’)$gutenberg_id[gutenberg_works(languages=’en’)$title %>% str_detect(’^Pri
```

```
## [1]   NA 1342
```

### 3)

```
book<-gutenberg_download(1342)
```

```
## Determining mirror for Project Gutenberg from http://www.gutenberg.org/robot/harvest
```

```
## Using mirror http://aleph.gutenberg.org
```

### 4)

```
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 4.0.3
```

```
words<-book %>% unnest_tokens(word,text)
words %>% head()
```

```
## # A tibble: 6 x 2
##   gutenberg_id word
##          <int> <chr>
## 1         1342 there
## 2         1342 is
## 3         1342 an
## 4         1342 illustrated
## 5         1342 edition
## 6         1342 of
```

### 5)

```
words<-words %>% mutate(location=1:nrow(words))
words %>% head()
```

```
## # A tibble: 6 x 3
##   gutenberg_id word        location
##          <int> <chr>          <int>
## 1         1342 there              1
## 2         1342 is                 2
## 3         1342 an                 3
## 4         1342 illustrated        4
## 5         1342 edition            5
## 6         1342 of                 6
```

### 6)

```
words<-words %>% anti_join(stop_words,by='word')
words %>% head()
```

```
## # A tibble: 6 x 3
##   gutenberg_id word        location
##          <int> <chr>          <int>
## 1         1342 illustrated        4
## 2         1342 edition            5
## 3         1342 title              8
## 4         1342 viewed            11
## 5         1342 ebook             13
## 6         1342 42671             14
```

**7)**

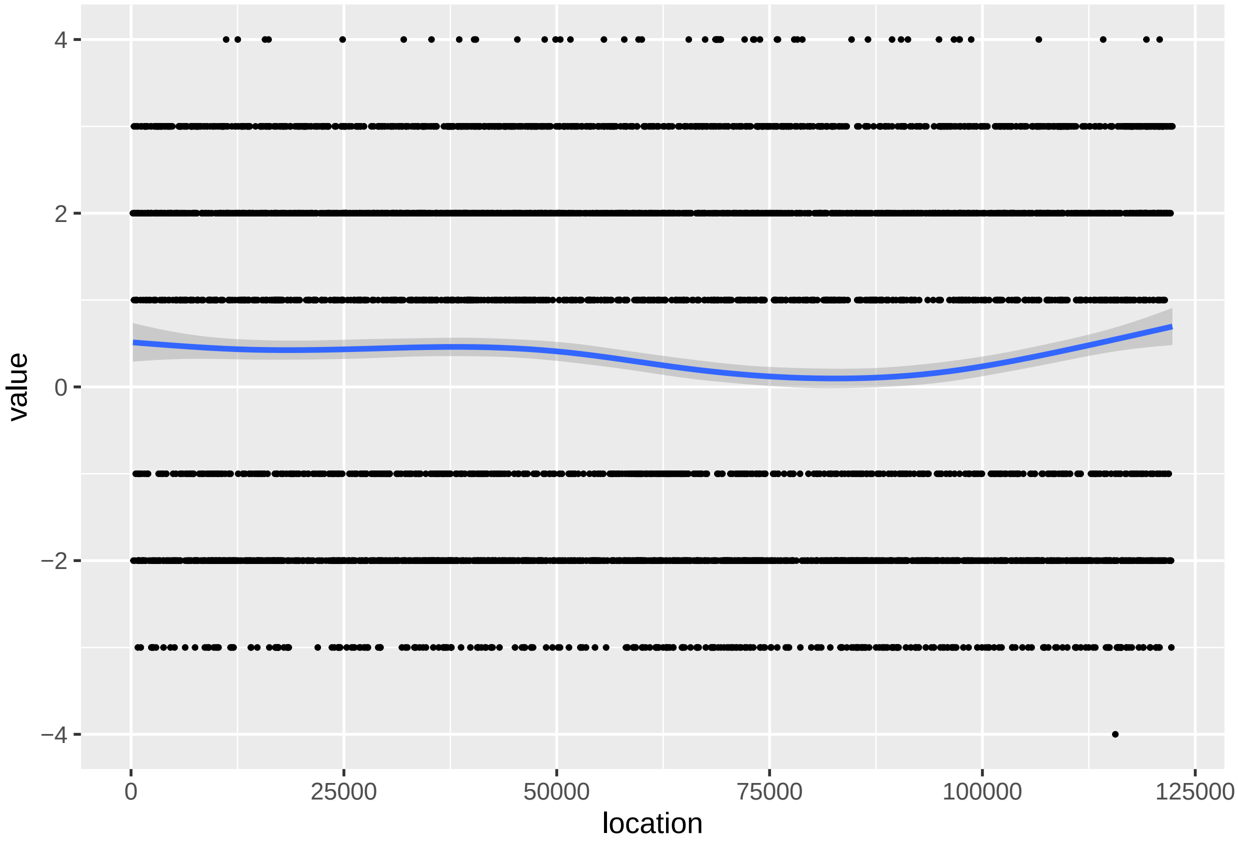
```
words<-words %>% inner_join(get_sentiments('afinn'),by='word')
words %>% head()
```

```
## # A tibble: 6 x 4
##   gutenberg_id word      location value
##          <int> <chr>        <int> <dbl>
## 1         1342 dear           218     2
## 2         1342 cried          279    -2
## 3         1342 dear           302     2
## 4         1342 delighted      344     3
## 5         1342 agreed         349     1
## 6         1342 dear           392     2
```

**8)**

```
words %>% ggplot(aes(location,value))+geom_point(size=.5)+geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```


value
location