Data 607 Week 3

Yuen Chun Wong

September 17, 2017

Week 3

Given a string

```
library(stringr)

raw.data <-"555-1239Moe Szyslak(636) 555-0113Burns, C. Montgomery555-6542Rev. Timothy Lovejoy555 8904Ne

print( raw.data )
```

```
## [1] "555-1239Moe Szyslak(636) 555-0113Burns, C. Montgomery555-6542Rev. Timothy Lovejoy555 8904Ned Fla
```

3.1) rearrange the vector so that all element conform to the standard first_name lastname

```
raw.data.fullname <- unlist(str_extract_all(raw.data, "[[:alpha:][:space:].,]{2,}"))
raw.data.fullname
```

```
## [1] "Moe Szyslak"          "Burns, C. Montgomery" "Rev. Timothy Lovejoy"
## [4] "Ned Flanders"         "Simpson, Homer"       "Dr. Julius Hibbert"
```

```
raw.data.lastname <- unlist(str_extract_all(raw.data.fullname, "[[:alpha:]]{2,},"))

raw.data.lastname
```

```
## [1] "Burns,"   "Simpson,"
```

```
raw.data.lastname2 <- unlist(str_extract_all(str_extract_all(raw.data.fullname, "[^,][:alpha:][:space:]
raw.data.lastname2
```

```
## [1] " Szyslak"  " Lovejoy"  " Flanders" " Hibbert"
```

```
raw.data.firstname <- unlist(str_extract_all(raw.data.fullname, "[,][.[:space:][:alpha:]]{2,}"))

raw.data.firstname
```

```
## [1] ", C. Montgomery" ", Homer"
```

```
raw.data.firstname2 <- unlist(str_extract_all(raw.data.fullname, "[^,][:alpha:]{2,}[:space:]"))
raw.data.firstname2
```

```
## [1] "Moe "      " Timothy " "Ned "      " Julius "
```

```
raw.data.firstname3 <- c(raw.data.firstname, raw.data.firstname2)
raw.data.lastname3 <- c(raw.data.lastname, raw.data.lastname2)

#use str_replace_all()
raw.data.firstname3 <- str_replace_all(raw.data.firstname3,",","")
raw.data.lastname3 <- str_replace_all(raw.data.lastname3,",","")

df <- data.frame(
      firstname = raw.data.firstname3,
      lastname = raw.data.lastname3
                 )
```

```
print(df)
```

```
##        firstname  lastname
## 1  C. Montgomery     Burns
## 2          Homer   Simpson
## 3            Moe    Szyslak
## 4       Timothy    Lovejoy
## 5           Ned   Flanders
## 6         Julius    Hibbert
```

3.2) Construct a logical vector indicating whether a character has a title (ie. Rev. And Dr)

Try to construct regular expression to detect title which beginning with no space, contain a-z, A-Z and follow by "."

```
title_regex <- "[[:alpha:]]{2,}[.]"
WithTitle <- unlist(str_extract_all(raw.data.fullname,title_regex))
print(WithTitle)
```

```
## [1] "Rev." "Dr."
```

Build a table to show the result

```
chk_title <- data.frame(
        fullname <- unlist(raw.data.fullname),
        HasTitle <- str_detect(raw.data.fullname, title_regex)
        )

print(chk_title)
```

```
##   fullname....unlist.raw.data.fullname.
## 1                            Moe Szyslak
## 2                  Burns, C. Montgomery
## 3                  Rev. Timothy Lovejoy
## 4                          Ned Flanders
## 5                       Simpson, Homer
## 6                    Dr. Julius Hibbert
##   HasTitle....str_detect.raw.data.fullname..title_regex.
## 1                                                  FALSE
## 2                                                  FALSE
## 3                                                   TRUE
## 4                                                  FALSE
## 5                                                  FALSE
## 6                                                   TRUE
```

3.3) Construct a logical vector indicating whether a character has a second name

construct a logic to detech second name that has "space" in the name between "words"

```
secondname_regex <- "[[:alpha:].]{2,}[[:space:]][[:alpha:]]{2,}"
WithsecondName <- unlist(str_extract_all(raw.data.firstname3,secondname_regex))
print(WithsecondName)
```

```
## [1] "C. Montgomery"
```

Build a table to show the result

```
chk_secondname <- data.frame(
        firstname <- unlist(raw.data.firstname3),
```

```
        HasTitle <- str_detect(raw.data.firstname3, secondname_regex)
        )

print(chk_secondname)
```

```
##   firstname....unlist.raw.data.firstname3.
## 1                              C. Montgomery
## 2                                      Homer
## 3                                        Moe
## 4                                    Timothy
## 5                                        Ned
## 6                                     Julius
##   HasTitle....str_detect.raw.data.firstname3..secondname_regex.
## 1                                                          TRUE
## 2                                                         FALSE
## 3                                                         FALSE
## 4                                                         FALSE
## 5                                                         FALSE
## 6                                                         FALSE
```

4)Describe the types of strings that conform to the following regular expressions and construct an example that is matched by the regular expression. (1) [0-9]+\$ Ans: match all numberic number, at least one digit, and then end of the line. "[[:digit:]]+"

(2) \b[a-z]{1,4}\b Ans: with word bounds at the begining and the end of the word, word with lower case alphabetic, at least one characters, max 4 characters. "\b[[:lower:]]{1,4}\b"

(3) .*?\.txt$ Ans: match anything from zero to many times follow by ".txt" "[[:alnum:]]*.txt"

(4) \d{2}/\d{2}/\d{4}

Ans: It is the date format mm/dd/yyyy format.
"[[:digit:]]{2}/[[:digit:]]{2}/[[:digit:]]{4}""

(5) <(.+?)>.+?</\1> Ans: match anyting "<[[:alnum:]]{1,}>[[:alnum:]]{1,}</\1>"