607 Homework 2

Oluwakemi Omotunde

September 13, 2016

```
library(stringr)
```

3 Copy the introductory example. The vector name stores the extracted names.

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

raw.data
```

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

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

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

#a) Use the tools of this chapter to rearrange the vector so that all elements conform to the standard first_name_ last_name.

```
  name.initial <- sub(" [A-z]{1}\\. "," ",name)
  name.initial
```

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

#this removes the initials that are present.

```
  name.prefix <- sub("[A-z]{2,3}\\. "," ",name.initial)
  name.prefix
```

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

#this removes the prefixes that are present.

```
  name.switch <- sub("(\\w+),\\s(\\w+)","\\2 \\1", name.prefix)
  name.switch
```

```
## [1] "Moe Szyslak"      "Montgomery Burns" " Timothy Lovejoy"
## [4] "Ned Flanders"     "Homer Simpson"    " Julius Hibbert"
```

#this will switch the first and last names. this helped me realize how important the correct spacing is.

#citation for help: http://stackoverflow.com/questions/33826650/last-name-first-name-to-first-name-last-name

#b) Construct a logical vector indicating whether a character has a title (i.e Rec and Dr.).

```
  title.name <-str_detect(name, "[A-z]{2,3}\\.") #alpha characters, length 2 and 3, and periods
  title.name
```

```
## [1] FALSE FALSE  TRUE FALSE FALSE  TRUE
```

#displays TRUE for entries with prefixes.

#c) Construct a logical vector indicating whether a character has a second name.

```
  second.name <- str_detect(name, " [A-z]{1}\\. ") #alpha characters, length 1, periods.
  second.name
```

```
## [1] FALSE  TRUE FALSE FALSE FALSE FALSE
```

#this is another one that took me quite a while to work out because of a spacing discrepancy

4 Describe the types of strings that confrom to the following regular expressions and construct and example that is matched by the expression.

#a) [0-9]+\$ #digits zero through nine followed by the dollar sign. The + tells us that the numbers will be matched one or more times

```
    example.a <- c('251$abc', '0141$', '$123', 'notit','589$')
    str_detect(example.a, "[0-9]+\\$")
```

```
## [1]  TRUE  TRUE FALSE FALSE  TRUE
```

#b) \b[a-z]{1,4}\b #lower case word of length one to four located at the end(word edge)

```
    example.b <- c('MATH', 'math', 'mathematics', 'i', 'be', 'can')
    str_detect(example.b, "\\b[a-z]{1,4}\\b")
```

```
## [1] FALSE  TRUE FALSE  TRUE  TRUE  TRUE
```

#c) .*?\.txt$ #items ending in .txt

```
    example.c <- c('bmcc.txt', 'change.txt', '.txt', 'not.txt.working')
    str_detect(example.c, ".*?\\.txt$")
```

```
## [1]  TRUE  TRUE  TRUE FALSE
```

#d) \d{2}/\d{2}/\d{4} #digits of length 2, 2, 4 (01/11/1999)

```
    example.d <- c('01/11/1988', '1999/01/11', '25-36-8585', '25/36/8585')
    str_detect(example.d, "\\d{2}/\\d{2}/\\d{4}")
```

```
## [1]  TRUE FALSE FALSE  TRUE
```

#e) <(.+?)>.+?</\1> #This one was a little difficult for me. I think it will return items in the format random

```
    example.e <- c('<text>random</text>', '<your>book</your>', '<your>book<mine>')
    str_detect(example.e, "<(.+?)>.+?</\\1>")
```

```
## [1]  TRUE  TRUE FALSE
```

9 The following code hides a secret message. Crack it with R and regular expressions.

I will continue to work on this particular question after I submit.