title: “DATA 607 Assignment 3”
author: “Md Jalal Uddin”
date: “September 18, 2016”

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

```
library(stringr)
raw.data <- "555-1239Moe Szyslak(636) 555-0113Burns, C. Montgomery555
-6542Rev. Timothy Lovejoy555 8904Ned Flanders636-555-3226Simpson, Homer5553642Dr. Julius Hibbert"

name <- unlist(str_extract_all(raw.data, "[[:alpha:]., ]{2,}")) #Source:Automated Data Collecton wit R,
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.

```
name1 <- sub(" [A-z]{1}\\. "," ",name)  # remove initials
name1
```

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

```
# Source:http://stackoverflow.com/questions/33826650/last-name-first-name-to-first-name-last-name
name2 <- sub("(\\w+),\\s(\\w+)","\\2 \\1", name1) # switch last,first to first last

name2
```

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

```
name3 <- sub("[A-z]{2,3}\\. ","",name2)
#remove titles
name3
```

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

(b) Construct a logical vector indicating whether a character has a title.

```
Show_title<-str_detect(name,"[[:alpha:]]{2,}\\.") #Source:Automated Data Collecton wit R, section 8.1.2
title <- data.frame(name,Show_title)
title
```

```
##                     name Show_title
## 1          Moe Szyslak      FALSE
## 2 Burns, C. Montgomery      FALSE
## 3 Rev. Timothy Lovejoy       TRUE
## 4          Ned Flanders      FALSE
## 5        Simpson, Homer      FALSE
## 6    Dr. Julius Hibbert       TRUE
```

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

```
Second_name <- str_detect(name," [A-z]\\.{1}")
Second <- data.frame(name,Second_name)
Second
```

```
##                    name Second_name
## 1           Moe Szyslak       FALSE
## 2 Burns, C. Montgomery         TRUE
## 3 Rev. Timothy Lovejoy        FALSE
## 4          Ned Flanders       FALSE
## 5        Simpson, Homer       FALSE
## 6    Dr. Julius Hibbert       FALSE
```

```
#Source:Automated Data Collecton wit R, section 8.2.1, page 209
```

Problem 4:Describe the types of strings that conform to the following regular expressions and construct an example that is matched by the regular expression. (a) [0-9]+\$ (b) \b[a-z]{1,4}\b (c) .*?\.txt$ (d) \d{2}/\d{2}/\d{4} (e) <(.+?)>.+?</\1>

(a) [0-9]+\$

```
raw<- "I love new york.56967$ NY- 11/34/7453, 587932$, 30032$"
str_extract(raw, "[0-9]+\\$")
```

```
## [1] "56967$"
```

```
str_extract_all(raw, "[0-9]+\\$")
```

```
## [[1]]
## [1] "56967$"  "587932$" "30032$"
```

Description: take any digit from zero to nine and any number of digit with dollar sign. the example is given above.

(b) \b[a-z]{1,4}\b

```
raw<- "Ryan love New Youk. I love New york.56967$ NY-567456, born- 11/34/7453, 587932$, bercelona4b"

str_extract(raw, "\\b[a-z]{1,4}\\b")
```

```
## [1] "love"
```

```r
str_extract_all(raw, "\\b[a-z]{1,4}\\b")
```

```
## [[1]]
## [1] "love" "love" "york" "born"
```

Description: It will give us any letter with at least 1 but not more than 4 letter word with all small letter.

(c) .*?\.txt$

```r
raw<- " Ryan love New Youk. I love New york.56967$ NY-567456,  lock_dplyr_fake.rmt.txt"

str_extract(raw, ".*?\\.txt$")
```

```
## [1] " Ryan love New Youk. I love New york.56967$ NY-567456,  lock_dplyr_fake.rmt.txt"
```

```r
str_extract_all(raw, "\\.txt$")
```

```
## [[1]]
## [1] ".txt"
```

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

```r
raw<- "I love new york.56967$ NY-11-32-4567, born- 11/08/1993, 587932$, 30032$"
str_extract(raw, "\\d{2}/\\d{2}/\\d{4}")
```

```
## [1] "11/08/1993"
```

```r
str_extract_all(raw, "\\d{2}/\\d{2}/\\d{4}")
```

```
## [[1]]
## [1] "11/08/1993"
```

Description: the above expression give the date with the following format 11/11/1111

(e) <(.+?)>.+?</\1>

```r
raw<- " Ryan love a. I love New york.56967$ NY-567456, born- 11/34/7453, 5.6/9 587932$, 0.123$  abc201

str_extract(raw, "<(.+?)>.+?</\\1>")
```

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

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

clcopCow1zmstc0d87wnkig7OvdicpNuggvhryn92Gjuwczi8hqrfpRxs5Aj5dwpn0TanwoUwisdij7Lj8kpf03AT5
Idr3coc0bt7yczjatOaootj55t3Nj3ne6c4Sfek.r1w1YwwojigOd6vrfUrbz2.2bkAnbhzgv4R9i05zEcrop.wAgn
b.SqoU65fPa1otfb7wEm24k6t3sR9zqe5fy89n6Nd5t9kc4fE905gmc4Rgxo5nhDk!gr

```
secret <- "clcopCow1zmstc0d87wnkig7OvdicpNuggvhryn92Gjuwczi8hqrfpRxs5Aj5dwpn0TanwoUwisdijLj8kpf03AT5Idr
```

Solution:

```
#Seperating all the capital letter.
Pattern <- unlist(str_extract_all(secret, "[[:upper:].]")) #Source:Automated Data Collecton wit R, Tabl
Pattern
```

```
##  [1] "C" "O" "N" "G" "R" "A" "T" "U" "L" "A" "T" "I" "O" "N" "S" "." "Y"
## [18] "O" "U" "." "A" "R" "E" "." "A" "." "S" "U" "P" "E" "R" "N" "E" "R"
## [35] "D"
```

```
Pattern <- paste(Pattern, sep=" ", collapse="")
# remove all the ""
Pattern
```

```
## [1] "CONGRATULATIONS.YOU.ARE.A.SUPERNERD"
```