DATA607_ASSIGNMENT3

Krishna Rajan

2/17/2018

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 8904Ne
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 thestandardfirst_name last_name.

```
name
```

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

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

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

```
fst_lst_name_final <- sub("(\\w+),\\s(\\w+)","\\2 \\1", sub("[A-z]{2,3}\\. ","",fst_lst_name))
fst_lst_name_final
```

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

```
data.frame(fst_lst_name_final)
```

```
##   fst_lst_name_final
## 1        Moe Szyslak
## 2   Burns,Montgomery
## 3    Timothy Lovejoy
## 4       Ned Flanders
## 5      Homer Simpson
## 6     Julius Hibbert
```

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

```
str_detect(name, "[[:alpha:]]{2,3}[.]")
```

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

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

```
fst_lst_name_final
```

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

```
grepl ( ",",str_trim(fst_lst_name_final))
```

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

(a) [0-9]+\$

```
##$a. contains continous numbers followed by $ sign
ninenines <-"999999999$"
unlist(str_extract_all(ninenines, "[0-9]+\\$"))
```

```
## [1] "999999999$"
```

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

```
##b. expression denotes a word with 1 to 4 lowercase alphabets
SNTCES <- ("Mary had a little lamb whose name was John ")
unlist(str_extract_all(SNTCES, "\\b[a-z]{1,4}\\b"))
```

```
## [1] "had"  "a"    "lamb" "name" "was"
```

(c) .*?\.txt$

```
##c. A word preceding period and ending with .txt
newsntcs <- ("mary had a little lamb whose name was john.txt")
unlist(str_extract_all(newsntcs, ".*?\\.txt$"))
```

```
## [1] "mary had a little lamb whose name was john.txt"
```

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

```
##d.Numbers in the pattern "2 numbers/2 numbers/4 numbers"
nmbrpttrn <- ("11/22/333 34/56/4444 55/66/99878")
unlist(str_extract_all(nmbrpttrn, "\\d{2}/\\d{2}/\\d{4}"))
```

```
## [1] "34/56/4444" "55/66/9987"
```

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

```
##e.evaluate start and end markup tags which are well formed
tags <- c(" <MAry>had a little/>","<lamb>whose</fleece>","<was>white</as snow>")
tags <- str_extract(tags, "<(.+?)>.+?</\\1>")
tags
```

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

9.9. The following code hides a secret message. Crack it with R and regular expressions. Hint: Some of the characters are more revealing than others! The code snippet is also available in the materials at www.r-datacollection.com.clcopCow1zmstc0d87wnkig7OvdicpNuggvhryn92Gjuwczi8hqrfpRxs5Aj5d d6vrfUrbz2.2bkAnbhzgv4R9i05zEcrop.wAgnb.SqoU65fPa1otfb7wEm24k6t3sR9zqe5 fy89n6Nd5t9kc4fE905gmc4Rgxo5nhDk!gr

```
## observe the capital letters that pop up every where in the sentence what if we extract only capital l
decode <-("clcopCow1zmstc0d87wnkig7OvdicpNuggvhryn92Gjuwczi8hqrfpRxs5Aj5dwpn0Tanwo
     Uwisdij7Lj8kpf03AT5Idr3coc0bt7yczjat0aootj55t3Nj3ne6c4Sfek.r1w1YwwojigO
     d6vrfUrbz2.2bkAnbhzgv4R9i05zEcrop.wAgnb.SqoU65fPa1otfb7wEm24k6t3sR9zqe5
     fy89n6Nd5t9kc4fE905gmc4Rgxo5nhDk!gr")
decode_1 <-unlist(str_extract_all(decode,"[[:upper:].]{1,}"))

decode_2 <- str_replace_all(paste(decode_1, collapse = ''), "[.]", " ")
decode_2
```

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