Hantz_Angrand_HW3

Hantz Angrand

September 16, 2018

Exercise 3

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

library(stringr)

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"
```

```r
#extract phone number
phone <- unlist(str_extract_all(raw.data, "\\(?(\\d{3})?\\)?(-| )?\\d{3}(-| )?\\d{4}"))
phone
```

```
## [1] "555-1239"       "(636) 555-0113" "555-6542"       "555 8904"
## [5] "636-555-3226"   "5553642"
```

```r
#Createw data frame
data.frame(name= name)
```

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

```r
#get first name
fname <- unlist(str_extract_all(name, "[:punct:] [[:alpha:]]{2,}$|[[:alpha:]]{2,} "))
first_name<-unlist(str_extract_all(fname, "[[:alpha:]]{2,}"))
first_name
```

```
## [1] "Moe"        "Montgomery" "Timothy"    "Ned"        "Homer"
## [6] "Julius"
```

```r
#get Last Name
lname<-unlist(str_extract_all(name, "[^[:punct:]] [[:alpha:]]{2,}$|[[:alpha:]]{2,}, "))
last_name<-unlist(str_extract_all(lname, "[[:alpha:]]{2,}"))
last_name
```

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

```r
#Update data frame
data.frame(first_name=first_name, last_name=last_name)
```

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

Construct a logical vector indicating wheter a character has a title

```
#Extract title from characters
title<-unlist(str_extract_all(name,"[[:alpha:]]{2,}\\."))
title
```

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

```
#detect title in list
title_detect<-unlist(str_detect(name, title))
title_detect
```

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

```
#Update data frame
data.frame(first_name=first_name, last_name=last_name, title_detect=title_detect)
```

```
##   first_name last_name title_detect
## 1        Moe   Szyslak        FALSE
## 2 Montgomery     Burns        FALSE
## 3    Timothy   Lovejoy         TRUE
## 4        Ned  Flanders        FALSE
## 5      Homer   Simpson        FALSE
## 6     Julius   Hibbert         TRUE
```

Construct a logical vector to indicate whether a character has a middle name

```
second_name<-unlist(str_detect(name, " [[:alpha:]]{1}\\.? [[:alpha:]]{1,}\\.?"))
second_name
```

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

```
#Update data fram
data.frame(first_name=first_name, last_name=last_name, title_detect=title_detect, second_name=second_na
```

```
##   first_name last_name title_detect second_name
## 1        Moe   Szyslak        FALSE       FALSE
## 2 Montgomery     Burns        FALSE        TRUE
## 3    Timothy   Lovejoy         TRUE       FALSE
## 4        Ned  Flanders        FALSE       FALSE
## 5      Homer   Simpson        FALSE       FALSE
## 6     Julius   Hibbert         TRUE       FALSE
```

Describe the types of strings that conform to the following regular expressions and

construct an example that is matched by the regular expression.

```r
library(stringr)
pat<-"[0-9]+\\$"
#Meaning one or more number followes by a number
ex<-c("56$", "bill67$balloon", "12345$")
nber_detect<-str_detect(ex,pat)
nber_detect
```

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

```r
pattern="\\b[a-z]{1,4}\\b"
#Meaning a lowercase word of minimum 1 letter and maximum 4 letters
ex<-c("h","sde", "wxyz")
letter_detect<-unlist(str_detect(ex,pattern))
letter_detect
```

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

```r
pat=".*?\\.txt$"
#Meaning ending by .txt
ex<-c("file.txt", "abc.txt","456abc.txt")
ending_detect<-str_detect(ex,pat)
ending_detect
```

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

```r
patrn = "\\d{2}/\\d{2}/\\d{4}"
#Meaning number in the format of nn/nn/nnnn
ex<-c("12/08/3456","30/30/3000", "02/09/2019 Excellent")
format_detect<-str_detect(ex,patrn)
format_detect
```

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

```r
pattern="<(.+?)>.+?</\\1>"
#Meaning one or more element between brackets follow by element and follow by element between bracket
ex<-c("<tag>Text</tag>","<html>Hello world</html>")
el_detect<-str_detect(ex,pattern)
el_detect
```

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

Decode the secret message

```r
mes<-"clcopCow1zmstc0d87wnkig7OvdicpNuggvhryn92Gjuwczi8hqrfpRxs5Aj5dwpn0Tanwo
Uwisdij7Lj8kpf03AT5Idr3coc0bt7yczjatOaootj55t3Nj3ne6c4Sfek.r1w1YwwojigO
d6vrfUrbz2.2bkAnbhzgv4R9i05zEcrop.wAgnb.SqoU65fPa1otfb7wEm24k6t3sR9zqe5
fy89n6Nd5t9kc4fE905gmc4Rgxo5nhDk!gr"

decode<-str_extract_all(mes,pattern="[[:upper:]]")
```

```
decode
```

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