Assignment\_3

Priya Shaji

2/14/2019

## R Character Manipulation and Date Processing

### Question 3

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

library(stringr)  
library(knitr)

Introductory Example

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

Extract vectors, Vector name stores the extracted name

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"

Replace the title and the middle name for name dataframe with blank

remove\_title\_middle <- str\_replace(name,"([[:alpha:]]{1,3}\\.\\s)", "")

Rearrange the vector so that all elements conform to the standard first\_name last\_name.

Reverse the first and last name where necessary

firstName\_lastName <- str\_replace(remove\_title\_middle, "([[:alpha:]]+), ([[:alpha:]]+)", "\\2 \\1")  
  
kable(list(data.frame(name, firstName\_lastName)), caption = "Reorder \"name\" so it conform to the standard first and last name format.")

Reorder “name” so it conform to the standard first and last name format.

|  |  |
| --- | --- |
| name | firstName\_lastName |
| Moe Szyslak | Moe Szyslak |
| Burns, C. Montgomery | Montgomery Burns |
| Rev. Timothy Lovejoy | Timothy Lovejoy |
| Ned Flanders | Ned Flanders |
| Simpson, Homer | Homer Simpson |
| Dr. Julius Hibbert | Julius Hibbert |

As we can see, all the names are separated as the standard format, firstName, lastName

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

title\_char <- str\_detect(name, "[[:alpha:]]{2,3}\\. ")  
kable(list(data.frame(name, title\_char)), caption = "Detect if names in \"name\" has a title.")

Detect if names in “name” has a title.

|  |  |
| --- | --- |
| name | title\_char |
| Moe Szyslak | FALSE |
| Burns, C. Montgomery | FALSE |
| Rev. Timothy Lovejoy | TRUE |
| Ned Flanders | FALSE |
| Simpson, Homer | FALSE |
| Dr. Julius Hibbert | TRUE |

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

second\_name <- str\_detect(name, " [[:alpha:],]{1,}")  
kable(list(data.frame(name, second\_name)), caption = "Detect if names in \"name\" has a second name.")

Detect if names in “name” has a second name.

|  |  |
| --- | --- |
| name | second\_name |
| Moe Szyslak | TRUE |
| Burns, C. Montgomery | TRUE |
| Rev. Timothy Lovejoy | TRUE |
| Ned Flanders | TRUE |
| Simpson, Homer | TRUE |
| Dr. Julius Hibbert | TRUE |

All the names in the name vector has a second name, therefore, the resulting table has TRUE for all the inputs.

### Question 4

1. Describe the types of strings that conform to the following regular expressions and construct an example that is matched by the regular expression.
2. [0-9]+\$

This regular expression is matched by a string which starts with a number/s [0-9], + sign means that the preceeding item will be matched one or more time, \$ ends with a $ sign

example:

str\_1<- c("123$", "abc$", "-1098$", "test$", 27)  
test\_1<- unlist(str\_extract\_all(str\_1, "[0-9]+\\$"))  
test\_1

## [1] "123$" "1098$"

To detect if the above regular expression works:

test\_1 <- str\_detect(str\_1, "[0-9]+\\$")  
test\_1

## [1] TRUE FALSE TRUE FALSE FALSE

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

This regular expression is matched by a string which is followed by lowe case alphabets which ranges between a to z.We are asking the function for all instances where this sequence appears at least once, but at most four times. \b indicates the word edges

str\_2<- c("aa", "bzdg", "bnewyork", "jeny")  
test\_2<- unlist(str\_extract\_all(str\_2, "\\b[a-z]{1,4}\\b"))  
test\_2

## [1] "aa" "bzdg" "jeny"

To detect if the above regular expression works:

test\_2 <- str\_detect(str\_2, "\\b[a-z]{1,4}\\b")  
test\_2

## [1] TRUE TRUE FALSE TRUE

1. .\*?\.txt$

This regular expression is matched by any string which ends by .txt

str\_3<- c("aa.txt", "bzdg", "$bnewyork", "sneha.txt")  
test\_3<- unlist(str\_extract\_all(str\_3, ".\*?\\.txt$"))  
test\_3

## [1] "aa.txt" "sneha.txt"

To detect if the above regular expression works:

test\_3 <- str\_detect(str\_3, "\\b[a-z]{1,4}\\b")  
test\_3

## [1] TRUE TRUE FALSE TRUE

1. \d{2}/\d{2}/\d{4}

This regular expression is matched by a number pattern dd/dd/dddd

str\_4<- c("12/12/2019", "11-22/2018", "11/22/2018", "sneha.txt")  
test\_4<- unlist(str\_extract\_all(str\_4, "\\d{2}/\\d{2}/\\d{4}"))  
test\_4

## [1] "12/12/2019" "11/22/2018"

To detect if the above regular expression works:

test\_4 <- str\_detect(str\_4, "\\b[a-z]{1,4}\\b")  
test\_4

## [1] FALSE FALSE FALSE TRUE

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

This regular expression is matched by a pattern with open and closed tags, like the html tags.

str\_5 <- c("<b> qwerty </b>", "<h1>priya <h1>", "<>test</tag>", "<tag> helloworld </tag>")  
test\_5 <- unlist(str\_extract\_all(str\_5, "<(.+?)>.+?</\\1>"))  
test\_5

## [1] "<b> qwerty </b>" "<tag> helloworld </tag>"

To detect if the above regular expression works:

test\_5 <- str\_detect(str\_5, "<(.+?)>.+?</\\1>")  
test\_5

## [1] TRUE FALSE FALSE TRUE

### Question 9

1. 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. clcopCow1zmstc0d87wnkig7OvdicpNuggvhryn92Gjuwczi8hqrfpRxs5Aj5dwpn0TanwoUwisdij7Lj8kpf03AT5Idr3coc0bt7yczjatOaootj55t3Nj3ne6c4Sfek.r1w1YwwojigOd6vrfUrbz2.2bkAnbhzgv4R9i05zEcrop.wAgnb.SqoU65fPa1otfb7wEm24k6t3sR9zqe5fy89n6Nd5t9kc4fE905gmc4Rgxo5nhDk!gr

str\_test <- "clcopCow1zmstc0d87wnkig7OvdicpNuggvhryn92Gjuwczi8hqrfpRxs5Aj5dwpn0TanwoUwisdij7Lj8kpf03AT5Idr3coc0bt7yczjatOaootj55t3Nj3ne6c4Sfek.r1w1YwwojigOd6vrfUrbz2.2bkAnbhzgv4R9i05zEcrop.wAgnb.SqoU65fPa1otfb7wEm24k6t3sR9zqe5fy89n6Nd5t9kc4fE905gmc4Rgxo5nhDk!gr"  
str\_extract\_all(str\_test, "[a-z]")

## [[1]]  
## [1] "c" "l" "c" "o" "p" "o" "w" "z" "m" "s" "t" "c" "d" "w" "n" "k" "i"  
## [18] "g" "v" "d" "i" "c" "p" "u" "g" "g" "v" "h" "r" "y" "n" "j" "u" "w"  
## [35] "c" "z" "i" "h" "q" "r" "f" "p" "x" "s" "j" "d" "w" "p" "n" "a" "n"  
## [52] "w" "o" "w" "i" "s" "d" "i" "j" "j" "k" "p" "f" "d" "r" "c" "o" "c"  
## [69] "b" "t" "y" "c" "z" "j" "a" "t" "a" "o" "o" "t" "j" "t" "j" "n" "e"  
## [86] "c" "f" "e" "k" "r" "w" "w" "w" "o" "j" "i" "g" "d" "v" "r" "f" "r"  
## [103] "b" "z" "b" "k" "n" "b" "h" "z" "g" "v" "i" "z" "c" "r" "o" "p" "w"  
## [120] "g" "n" "b" "q" "o" "f" "a" "o" "t" "f" "b" "w" "m" "k" "t" "s" "z"  
## [137] "q" "e" "f" "y" "n" "d" "t" "k" "c" "f" "g" "m" "c" "g" "x" "o" "n"  
## [154] "h" "k" "g" "r"

str\_extract\_all(str\_test, "[A-Z]")

## [[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"

The hidden message is revealed. “CONGRATULATIONS YOU ARE A SUPER NERD”