week\_3\_assignment\_JimmyNg

Jimmy Ng

2/15/2019

library(tidyverse)

# 3.1) Use the tools of this chapter to rearrange the vector so that all elements conform to the standard first\_name last\_name.

# raw.data  
raw.data <- "555-1239Moe Szyslak(636) 555-0113Burns, C. Montgomery555-6542Rev. Timothy Lovejoy555 8904Ned Flanders636-555-3226Simpson, Homer5553642Dr. Julius Hibbert"  
  
# name.transform1 : extract all the names (no numbers)  
raw.data %>%  
 str\_extract\_all(., pattern = "[[:alpha:]., ]{2,}") %>%  
 unlist -> name.transform1  
  
# remove titles manually : where are these "Rev." or "Dr."  
# which(str\_detect(name.transform1, "Rev."))   
# which(str\_detect(name.transform1, "Dr."))   
  
# name.transform2 : remove titles, then split the names into "first" and "last" and lastly convert it into a data.frame  
name.transform1 %>% gsub("(Rev\\. )|(Dr\\. )", replace = "", .) %>%  
 str\_split(., pattern = ", | ", n = 2) %>%  
 plyr::ldply(., rbind) -> name.transform2  
  
# change column names, and data type  
names(name.transform2) <- c("first", "last")  
name.transform2 <- name.transform2 %>%  
 dplyr::mutate(first = as.character(first),  
 last = as.character(last))  
  
# name.transform3 : manually arrange first and last name in correct order  
name.transform3 <- name.transform2   
name.transform3[2, 1] <- name.transform2[2, 2]  
name.transform3[2, 2] <- name.transform2[2, 1]  
name.transform3[5, 1] <- name.transform2[5, 2]  
name.transform3[5, 2] <- name.transform2[5, 1]  
  
# create title column, then insert it back to the data.frame  
title <- c(rep("na", nrow(name.transform3)))  
title[which(str\_detect(name.transform1, "Rev."))] <- "Rev."  
title[which(str\_detect(name.transform1, "Dr."))] <- "Dr."  
  
name.transform3 <- name.transform3 %>%  
 dplyr::mutate(title = title) %>%  
 select(title, first, last)  
  
# final data.frame  
name.transform3

## title first last  
## 1 na Moe Szyslak  
## 2 na C. Montgomery Burns  
## 3 Rev. Timothy Lovejoy  
## 4 na Ned Flanders  
## 5 na Homer Simpson  
## 6 Dr. Julius Hibbert

A little fun fact: although “C. Montgomery” is treated as first name in above data.frame, “Montgomery” is actually a middle name. “C” is the first name and it represents “Charles”.

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

name.transform4 <- name.transform3 %>%  
 dplyr::mutate(full = ifelse(title == "na", "", title) %>%  
 paste(., first, last, sep = " "),  
 title\_flag = str\_detect(full,   
 pattern = "(Rev\\.|Dr\\.)"))  
name.transform4

## title first last full title\_flag  
## 1 na Moe Szyslak Moe Szyslak FALSE  
## 2 na C. Montgomery Burns C. Montgomery Burns FALSE  
## 3 Rev. Timothy Lovejoy Rev. Timothy Lovejoy TRUE  
## 4 na Ned Flanders Ned Flanders FALSE  
## 5 na Homer Simpson Homer Simpson FALSE  
## 6 Dr. Julius Hibbert Dr. Julius Hibbert TRUE

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

name.transform5 <- name.transform4 %>%  
 dplyr::mutate( middle\_flag = ifelse(  
 str\_count(first, pattern = "\\S+") >1, # counting all sequences on non-space characters   
 TRUE, FALSE) )  
name.transform5

## title first last full title\_flag middle\_flag  
## 1 na Moe Szyslak Moe Szyslak FALSE FALSE  
## 2 na C. Montgomery Burns C. Montgomery Burns FALSE TRUE  
## 3 Rev. Timothy Lovejoy Rev. Timothy Lovejoy TRUE FALSE  
## 4 na Ned Flanders Ned Flanders FALSE FALSE  
## 5 na Homer Simpson Homer Simpson FALSE FALSE  
## 6 Dr. Julius Hibbert Dr. Julius Hibbert TRUE FALSE

I am guessing the question is asking which character is having a middle name. In this case, that’s Mr. Burns.

# 4) Describe the types of strings that conform to the following regular expressions and construct an example that is matched by the regular expression.

# 1. [0-9]+\\$  
# this is referred to 1 or more digits followed by a dollar sign  
str\_extract\_all(c("44$", "4j$"),   
 pattern = "[0-9]+\\$") %>% unlist

## [1] "44$"

# 2. \\b[a-z]{1,4}\\b  
# this is referred to any word (in lower case) inside a string that has a boundary (word edge)  
# AND it must be at least 1 and up to 4 letters  
str\_extract\_all(c("good", "not so good", "hi goodbye",  
 "heyheyhey", "NOT SO GOOD", "88lu"),   
 pattern = "\\b[a-z]{1,4}\\b") %>% unlist

## [1] "good" "not" "so" "good" "hi"

# 3. .\*?\\.txt$  
# this is referred to anything that ends with .txt  
# before .txt, anything can be optional, including nothing (.\*?)  
str\_extract\_all(c("not so good.txt", ".txt",  
 "homework.txt.csv", "assignment\_txt"),   
 pattern = ".\*?\\.txt$") %>% unlist

## [1] "not so good.txt" ".txt"

# 4. \\d{2}/\\d{2}/\\d{4}  
# this is referred to a combo of digits separated by a slash /  
# specifically 2 digits separated by slash and then 2 digits, separated by slash again and followed by 4 digits  
# however, the beginning don't necessarily to be just 2 digits, e.g. 1111/88/9999 would still return true  
# likewise, the end can be more than 4, e.g. 1111/88/999999999 would still return true  
str\_extract\_all(c("haha 11/12/2001 oh yes", "06/14/9999", "1111/88/999999999",   
 "h9/88/9800", "randombirthdaywishes!:-)", "11/17a/2018"),  
 pattern = "\\d{2}/\\d{2}/\\d{4}") %>% unlist

## [1] "11/12/2001" "06/14/9999" "11/88/9999"

# 5. <(.+?)>.+?</\\1>  
# this is referred to something similar in html format, something may look like this  
# <SomethingInBetween>followedBySomethingInBetween</RepeatedWhatIsShowedatFirst>  
# \\1 is backreferencing, whereas .+? means optional appearance  
str\_extract\_all(c("<hihi>dllm</hihi>", "<hihi> </hihi>", "< > </ >",  
 "<hi>dllm</hihi>", "</hihi> </hihi>", "<> </>"),  
 pattern = "<(.+?)>.+?</\\1>") %>% unlist

## [1] "<hihi>dllm</hihi>" "<hihi> </hihi>" "< > </ >"

# 9) Bonus Question

secret\_code <- "clcopCow1zmstc0d87wnkig7OvdicpNuggvhryn92Gjuwczi8hqrfpRxs5Aj5dwpn0TanwoUwisdij7Lj8kpf03AT5Idr3coc0bt7yczjatOaootj55t3Nj3ne6c4Sfek.r1w1YwwojigOd6vrfUrbz2.2bkAnbhzgv4R9i05zEcrop.wAgnb.SqoU65fPa1otfb7wEm24k6t3sR9zqe5fy89n6Nd5t9kc4fE905gmc4Rgxo5nhDk!gr"  
  
secret\_code %>%  
 str\_extract\_all(., pattern = "[[A-Z]|[:punct:]]") %>% unlist %>%  
 cat

## C O N G R A T U L A T I O N S . Y O U . A R E . A . S U P E R N E R D !