8 Processing Text Homework

James Young

7/17/2019

There are six exercises below. You are required to provide solutions for at least four of the six. You are required to solve at least one exercise in R, and at least one in SAS. You are required to provide five solutions, each solution will be worth 10 points. Thus, you may choose to provide both R and SAS solutions for a single exercise, or you may solve five of the sixth problems, mixing the languages as you wish.

If you choose SAS for an exercise, you may use IML, DATA operations or PROC SQL at your discretion.

*Warning* I will continue restricting the use of external libraries in R, particularly tidyverse libraries. You may choose to use ggplot2, but take care that the plots you produce are at least as readable as the equivalent plots in base R. You will be allowed to use whatever libraries tickle your fancy in the midterm and final projects.

## Reuse

For many of these exercises, you may be able to reuse functions written in prior homework. Define those functions here.

#To turn matrix into csv compatible string for question 1  
stringmaker <- function(x){  
 csvnat <- do.call(paste, c(as.list(colnames(x)), sep=",", do.call(paste,c(x, sep=",")) ))  
 n <- dim(x)[2]  
 rep1 <- strrep("[^,]+,", n)  
   
 rep1 <- gsub(",$","",rep1)  
  
 rep1 <- paste0("(",rep1,"),")  
   
 o = gsub(rep1, "\\1\n", paste(csvnat, collapse = ","))  
   
   
 return(o)  
}

# Exercise 1 Please Grade.

Write a loop or a function to convert a matrix to a CSV compatible string. Given a matrix of the form

|  |  |  |
| --- | --- | --- |
| C1 | C2 | C3 |
| a | b | c |
| d | e | f |
| g | h | i |

produce a string of the form

a,b,c\nd,e,f\ng,h,i

where \n is the newline character.

You are only required to convert a matrix to CSV format, but you may choose to write code to convert data tables to CSV; in this case, include column names in the output string. Use NATR332.DAT as a test case.

NATR332.DAT <- data.frame(  
 Y1 = c(146,141,135,142,140,143,138,137,142,136),  
 Y2 = c(141,143,139,139,140,141,138,140,142,138)  
)  
  
  
stringmaker(NATR332.DAT)

## [1] "Y1,Y2\n146,141\n141,143\n135,139\n142,139\n140,140\n143,141\n138,138\n137,140\n142,142\n136,138"

If you choose SAS, I’ve include the NATR332 data table and framework code for IML in the template. I used the CATX function in IML. I found I could do this in one line in R, with judicious use of apply, but I haven’t found the equivalent in IML. Instead, I used a pair of nested loops to “accumulate” an increasingly longer string.

# Exercise 3 Please Grade.

Calculate MSW, MSB, and for the data from Wansink Table 1 (Homework 4, Exercise 6) where

Start with the strings:

Means <- "268.1 271.1 280.9 294.7 285.6 288.6 384.4"  
StandardDeviations <- "124.8 124.2 116.2 117.7 118.3 122.0 168.3"  
SampleSizes <- "18 18 18 18 18 18 18"

Tokenize the strings, then convert the tokens to a create vectors of numeric values. Use these vectors to compute and print , , and .

Mean1 = as.numeric(strsplit(Means,split=" ")[[1]])  
StandardDeviations1 = as.numeric(strsplit(StandardDeviations, split=" ")[[1]])  
SampleSizes1 = as.numeric(strsplit(SampleSizes, split=" ")[[1]][1])  
  
  
ms.len = length(Mean1)  
msb.nu = 0  
msw.nu = 0  
for(i in 1:length(Mean1)){  
 msb.nu = (SampleSizes1 \* ((Mean1[[i]]-mean(Mean1))^2)) + msb.nu  
 msw.nu = ((SampleSizes1 - 1) \* StandardDeviations1[[i]]^2)+msw.nu  
   
}  
msb = msb.nu/(ms.len-1)  
msw = msw.nu/(length(StandardDeviations1)\*SampleSizes1 - length(StandardDeviations1))  
  
F\_ratio = msb/msw  
  
  
p\_value = pf(F\_ratio, (length(Mean1)-1), (length(StandardDeviations1) \* SampleSizes1 - length(StandardDeviations1)), lower.tail=FALSE)  
  
  
msb

## [1] 28815.96

msw

## [1] 16508.6

F\_ratio

## [1] 1.745512

p\_value

## [1] 0.1163133

# Exercise 5 Please Grade.

Use the file openmat2015.csv from D2L. These data are from <https://news.theopenmat.com/high-school-wrestling/high-school-wrestling-rankings/final-2015-clinch-gear-national-high-school-wrestling-individual-rankings/57136>. This is a list of top-ranked high school wrestlers in 2015, their high School, Weight class and in some cases the College where they expected to enroll and compete after high school.

We wish to know how many went on to compete in the national championship in 2019, so we will merge this table with the data from Homework 7, ncaa2019.csv. The openmat2015.csv data contains only a single column, Name. You will need to split the text in this column to create the columns First and Last required to merge with ncaa2019.csv.

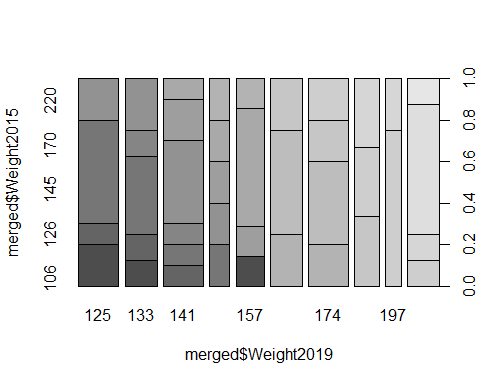
**Do not print these tables in the submitted work** Instead, print a contingency table comparing Weight for 2015 and Weight for 2019. What is the relationship between high school and college weight classes? You may instead produce a scatter plot or box-whisker plot, using high school weight class as the independent variable.

If you do this in SAS, use the openmat2015SAS.csv file, it will import College correctly.

r <- read.csv("C://data/ncaa2019.csv")  
d <- read.csv("C://data/openmat2015.csv")  
  
d$Name <- as.character(d$Name)  
  
First = sapply(strsplit(d$Name, ' '), function(x) x[1])  
Last = sapply(strsplit(d$Name, ' '), function(x) x[length(x)])  
r$Last <- as.character(r$Last)  
r$First <- as.character(r$First)  
d$Weight2015 <- as.numeric(d$Weight)  
r$Weight2019 <- as.numeric(r$Weight)  
open2015 <- cbind(First, Last, d$Weight2015)  
ncaa2019 <- cbind(r$First, r$Last, r$Weight2019)  
open2015 <-as.data.frame(open2015)  
ncaa2019 <-as.data.frame(ncaa2019)  
open2015$Weight2015 <-open2015$V3  
ncaa2019$Weight2019 <-ncaa2019$V3  
ncaa2019$First <- ncaa2019$V1  
ncaa2019$Last <- ncaa2019$V2  
ncaa2019$V1 <- NULL  
ncaa2019$V2 <- NULL  
open2015$V3 <- NULL  
ncaa2019$V3 <- NULL  
#makes merged table with first name, last name, and weights from 2015 and 2019  
merged = merge(open2015, ncaa2019, all =T)  
merged = na.omit(merged)  
#contingency table for condensed information  
contingency <- ftable(merged$Weight2015 ~ merged$Weight2019)  
contingency

## merged$Weight2015 106 113 120 126 132 138 145 152 160 170 182 195 220 285  
## merged$Weight2019   
## 125 2 1 5 0 2 0 0 0 0 0 0 0 0 0  
## 133 1 1 3 1 2 0 0 0 0 0 0 0 0 0  
## 141 0 1 1 1 4 2 1 0 0 0 0 0 0 0  
## 149 0 0 1 0 1 1 1 1 0 0 0 0 0 0  
## 157 1 0 0 0 0 1 4 1 0 0 0 0 0 0  
## 165 0 0 0 0 0 0 0 2 4 2 0 0 0 0  
## 174 0 0 0 0 0 0 0 2 4 2 2 0 0 0  
## 184 0 0 0 0 0 0 0 0 0 2 2 2 0 0  
## 197 0 0 0 0 0 0 0 0 0 0 3 1 0 0  
## 285 0 0 0 0 0 0 0 0 0 0 1 1 5 1

plot(merged$Weight2015 ~ merged$Weight2019)



# Exercise 6 Please Grade.

Use file openmat2015.csv from Exercise 6, and use partial text matching to answer these questions. To show your results, print only the rows from the table that match the described text patterns, but to save space, print only Name, School and College. Each of these can be answered in a single step.

d <- read.csv("C://data/openmat2015.csv")

* Which wrestlers come from a School with a name starting with St.?

d.schoolST = d[grepl("St.", d$School),]  
d.schoolST

## Weight Rank Name Year School State  
## 1 106 1 Cade Olivas Fr. St. John Bosco CA  
## 17 106 17 John Tropea Jr. St. Joseph Montvale NJ  
## 30 113 10 Mitch Moore Fr. St. Paris Graham OH  
## 37 113 17 Joey Prata Jr. St. Christopher's VA  
## 50 120 10 Eli Stickley Sr. St. Paris Graham OH  
## 64 126 4 Mitchell McKee Jr. St. Michael-Albertville MN  
## 67 126 7 Eli Seipel Sr. St. Paris Graham OH  
## 76 126 16 Ben Lamantia Sr. St. Anthonys NY  
## 82 132 2 Kaid Brock Sr. Stillwater OK  
## 94 132 14 Austin O'Connor So. St. Rita IL  
## 99 132 19 Hunter Ladnier Jr. St. Edward OH  
## 128 145 8 Brent Moore Jr. St. Paris Graham OH  
## 134 145 14 Tristan Moran Sr. Stillwater OK  
## 153 152 13 Kyle Lawson So. St. Paris Graham OH  
## 161 160 1 Alex Marinelli Jr. St. Paris Graham OH  
## 182 170 2 Anthony Valencia Sr. St. John Bosco CA  
## 183 170 3 Logan Massa Sr. St. Johns MI  
## 185 170 5 Joe Smith Sr. Stillwater OK  
## 201 182 1 Zahid Valencia Sr. St. John Bosco CA  
## 217 182 17 Jordan Joseph Sr. St. Michael-Albertville MN  
## 251 220 11 Christian Colucci Sr. St. Peter's Prep NJ  
## 255 220 15 Ian Butterbrodt Sr. St. Johns Prep MA  
## College Previous  
## 1 1  
## 17 9  
## 30 12  
## 37 HM  
## 50 Wisconsin 10  
## 64 Minnesota '16 3  
## 67 Pittsburgh 10  
## 76 Michigan 18  
## 82 Oklahoma State 2  
## 94 14  
## 99 16  
## 128 10  
## 134 Oklahoma State 13  
## 153 14  
## 161 Iowa '16 1  
## 182 Arizona State 2  
## 183 Michigan 3  
## 185 Oklahoma State 4(160)  
## 201 Arizona State 1  
## 217 18  
## 251 Lehigh 13  
## 255 Brown 17(285)

* Which wrestlers were intending to attend an Iowa College?

d.CollegeIowa = d[grepl("Iowa", d$College),]  
d.CollegeIowa

## Weight Rank Name Year School State  
## 21 113 1 Justin Mejia So. Clovis CA  
## 24 113 4 Jason Renteria So. Oak Park-River Forest IL  
## 65 126 5 Markus Simmons Sr. Broken Arrow OK  
## 121 145 1 Michael Kemerer Sr. Franklin Regional PA  
## 122 145 2 Max Thomsen Sr. Union Community IA  
## 155 152 15 Kaleb Young Jr. Punxsutawney PA  
## 161 160 1 Alex Marinelli Jr. St. Paris Graham OH  
## 166 160 6 Bryce Steiert Sr. Waverly-Shell Rock IA  
## 176 160 16 Paden Moore Sr. Jackson County Central MN  
## 194 170 14 Isaiah Patton Sr. Dowling Catholic IA  
## 196 170 16 Jacob Holschlag Sr. Union IA  
## 197 170 17 Colston DiBlasi Sr. Park Hill MO  
## 204 182 4 Taylor Lujan Sr. Carrollton GA  
## 233 195 13 Cash Wilcke Sr. OA-BCIG IA  
## 244 220 4 Ryan Parmely Sr. Maquoketa Valley IA  
## College Previous  
## 21 Iowa '17 2  
## 24 Iowa '17 5  
## 65 Iowa State 6  
## 121 Iowa 3  
## 122 Northern Iowa 1  
## 155 Iowa '16 9  
## 161 Iowa '16 1  
## 166 Northern Iowa 8  
## 176 Northern Iowa HM  
## 194 Northern Iowa 16  
## 196 Northern Iowa 18  
## 197 Iowa State 12  
## 204 Northern Iowa 4  
## 233 Iowa 11  
## 244 Upper Iowa 5

* Which wrestlers were intending to start College in 2016 or 2017 (College will end with 16 or 17)?

d.p = d[grepl(" '16", d$College),]  
d.p2 = d[grepl(" '17", d$College),]  
d.p3 = rbind(d.p2, d.p)  
d.p3

## Weight Rank Name Year School State  
## 21 113 1 Justin Mejia So. Clovis CA  
## 24 113 4 Jason Renteria So. Oak Park-River Forest IL  
## 45 120 5 Kyle Norstrem Jr. Brandon FL  
## 46 120 6 Jack Mueller Jr. Wyoming Seminary PA  
## 51 120 11 Ty Agaisse Jr. Delbarton NJ  
## 64 126 4 Mitchell McKee Jr. St. Michael-Albertville MN  
## 126 145 6 Hayden Hidlay Jr. Mifflin County PA  
## 145 152 5 Jake Wentzel Jr. South Park PA  
## 155 152 15 Kaleb Young Jr. Punxsutawney PA  
## 161 160 1 Alex Marinelli Jr. St. Paris Graham OH  
## 186 170 6 Nick Reenan Jr. Wyoming Seminary PA  
## College Previous  
## 21 Iowa '17 2  
## 24 Iowa '17 5  
## 45 Virginia Tech '16 5  
## 46 Virginia '16 6  
## 51 Princeton '16 11  
## 64 Minnesota '16 3  
## 126 NC State '16 6  
## 145 Pitt '16 5  
## 155 Iowa '16 9  
## 161 Iowa '16 1  
## 186 Northwestern '16 5

* Which wrestlers are intending compete in a sport other than wrestling? (look for a sport in parenthesis in the College column. Note - ( is a special character in regular expressions, so to match the exact character, it needs to be preceded by the escape character \. However, \ in strings is a special character, so itself must be preceded by the escape character.

dt = d[grepl("\\(", d$College),]  
dt

## Weight Rank Name Year School State  
## 218 182 18 Chase Osborn Sr. Penn IN  
## 225 195 5 Tevis Barlett Sr. Cheyenne East WY  
## 230 195 10 Jan Johnson Sr. Governor Mifflin PA  
## 261 285 1 Michael Johnson Sr. Montini Catholic IL  
## 264 285 4 Gage Cervenka Sr. Emerald SC  
## 267 285 7 Jake Marnin Sr. Southeast Polk IA  
## 277 285 17 Que Overton Sr. Jenks OK  
## 279 285 19 Norman Oglesby Sr. Benjamin Davis IN  
## College Previous  
## 218 Minnesota (Baseball) 19  
## 225 Washington (FB) 8  
## 230 Akron(FB) 3  
## 261 Yale (Football) 1  
## 264 Clemson (Football) 4  
## 267 Southern Illinois (Football) 7  
## 277 Oklahoma (Football) HM  
## 279 Cincinnati (Football) NR

Done