KATHLEEN ZHEN

999210972

STA 141A

HOMEWORK 4

The codes and results derived by using these codes constitute my own work. I have consulted the following resources regarding this assignment:"
http://stackoverflow.com/questions/9068397/import-text-file-as-single-character-string
PIAZZA

NUMBER 1

Please see Cars.csv file for all the data on the cars. The table below explains the content of the csv.

Column #	
1	The number location of the file from my loop
2	The full model string
3	The year of the car
4	The make of the car
5	The VIN of the car
6	The prices of car with $
7	The mileages on the car
8	The exterior color of the car
9	The interior color of the car
10	The transmission
11	The engine that ends in L
12	The company name
13	The address of the car company
14	The phone number of the car company
15	The website of the car company

When there's no detected fields, my code checked for that and replaced it with the string "Not Available".

NUMBER 2

Please see "webscrape.rda" for the data.frame of information from the website.

My data is named "newmueller". The first column contains the date of the publication, the second column are the authors, the third column is the name of publication, the fourth column is the journal in which the publications are in, the fifth column is the volume of the journal, and lastly, there is a link to publication.

Some statistical summary I did were finding the number of publications per year. I didn't include all the years here since there were a total of 34. The rest can be found in the variable datenum.

Year	2010	2011	2012	2013	2014	2015
Number of publication	10	7	8	4	6	3

The average number of co-authors for each publication is 2.
The number of unique groups of authors are 165.

Below are some of the output from finding how many publications were in the specific journals. The rest can be found in the variable numPub.

Bioinformatics, 3
Biometrical Journal, 1
Biometrics, 6
Biometrika, 16
Biostatistics, 2

I also found the number of NA's in Journal (6), URL (120), and volume (29).

APPENDIX

```
##sets path and working direction to the CarAdvert Folder
setwd("~/Desktop/Lectures/CarAdvert/")
path = ("~/Desktop/Lectures/CarAdvert/")

#gets all the file names
file.names <- dir(path, pattern =".txt")

#for loop to read all the files into a dataframe
singleString1 = NULL
for(i in 1:length(file.names)){
  singleString <- paste(readLines(file.names[i]), collapse="\n")
  singleString = rbind(singleString1, singleString)
  singleString1 = singleString
}

#sapply to go through all the files in the dataframe to extract relevant info
allyearmodel = sapply(1:1531, function (i) {
  #gets a single file
  file <- singleString[i]
  newfile = unlist(strsplit(singleString[i],split="\n"))

  #extracts the all the information of the car
  yearmodel = newfile[1]
  # gets just the year
  year = str_extract(yearmodel, "[0-9]{4}")
  # gets just the make of the car
  make = str_extract(yearmodel, "[A-Za-z]+")

  #finds locations of vin
  vin_stock = regexpr('[A-Z0-9]{13}[0-9]{4}', newfile)
  vin = regmatches(newfile, vin_stock) #grabs vin from locations

  #gets price
  test = regexpr('(\\$)[0-9]*\\,[0-9]{3}', newfile)
  price = regmatches(newfile, test)
  if (length(price) == 0) {
    price = 'Not Available' #if no price available, say not available
  }

  #finds the words mileages and gets the miles
  extractmiles = regexpr('Mileage: ([0-9])+\\,[0-9]{3}', newfile)
  miles = regmatches(newfile, extractmiles)
```

```
mileage = unlist(strsplit(miles, split = 'Mileage: '))[2] #detach mileage from the string
if (length(mileage) == 0) {
  mileage = 'Not Available'
}

#gets exterior color
extractex = regexpr('Exterior: [A-Za-z ]*(Interior|Body)', newfile)
ex_color = regmatches(newfile, extractex)
#splits string to just get color
exteriorcolor = unlist(strsplit(ex_color, split = "Exterior: ", fixed = TRUE))[2]
if (length(exteriorcolor) == 0) {
  exteriorcolor = 'Not Available'
}
excolor = unlist(strsplit(exteriorcolor, split = "Interior", fixed = TRUE))
excolor = unlist(strsplit(excolor, split = "Body", fixed = TRUE))

#gets interior color
interiorex = regexpr('Interior: [A-Za-z \\/]*(Body)?', newfile)
in_color = regmatches(newfile, interiorex)
interiorcolor = unlist(strsplit(in_color, split = "Interior: ", fixed = TRUE))[2]
if (length(interiorcolor) == 0) {
  interiorcolor = 'Not Available'
}
incolor = unlist(strsplit(interiorcolor, split = "Body", fixed = TRUE))
incolor = unlist(strsplit(incolor, split = "Transmission", fixed = TRUE))

#gets transmission number
trans_num = regexpr('Transmission: [A-Za-z 0-9-]*', newfile)
trans_phrase = regmatches(newfile, trans_num)
transm = unlist(strsplit(trans_phrase, split = "Transmission: ", fixed = TRUE))[2]
if (length(transm) == 0) {
  transm = 'Not Available'
}
transmi = unlist(strsplit(transm, split = "Engine", fixed = TRUE))

#gets engine type.
#this only gets the liters as stated in the instructions
engined = regexpr('[0-9]\\.[0-9]L', newfile)
engine_dis = regmatches(newfile, engined)
if (length(engine_dis) == 0) {
  engine_dis = 'Not Available'
}

#gets company
```

```
 comp = regexpr('Offered by: [A-Za-z &0-9\']* [?]', newfile)
 compnum = regmatches(newfile, comp)
 firstsplit = unlist(strsplit(compnum, split = "Offered by: "))[2]
 if (length(firstsplit) == 0) {

  firstsplit = 'Not Available'
 }
 compname = unlist(strsplit(firstsplit, split = " ?", fixed = TRUE))

 #gets address
 address = regexpr('[0-9A-Za-z, ]+ [0-9]{5}', newfile)
 addy = regmatches(newfile, address)
 if (length(addy) == 0) {
  addy = 'Not Available'
 }

 #gets phone number
 phonenum = regexpr('[\\(\\)0-9]{5} [0-9-]{8}', file)
 phoney = regmatches(file, phonenum)
 if (length(phoney) == 0) {
  phoney = 'Not Available'
 }

 #finds website that ends in .com
 webby = regexpr('[A-Za-z.0-9\\]*\\.com', file)
 website = regmatches(file, webby)
 if (length(website) == 0) {
  website = 'Not Available'
 }

 #binds all the lists together
 return (cbind(yearmodel, year, make, vin, price, mileage, excolor, incolor, transmi, engine_dis,
compname, addy, phoney, website))
})

#converts list to dataframe
ym = sapply(allyearmodel, '[', 1)
yeaar = sapply(allyearmodel, '[', 2)
maake = sapply(allyearmodel, '[', 3)
v = sapply(allyearmodel, '[', 4)
p = sapply(allyearmodel, '[', 5)
m = sapply(allyearmodel, '[', 6)
eC = sapply(allyearmodel, '[', 7)
iC = sapply(allyearmodel, '[', 8)
```

```
t = sapply(allyearmodel, '[', 9)
d = sapply(allyearmodel, '[', 10)
n = sapply(allyearmodel, '[', 11)
a = sapply(allyearmodel, '[', 12)
pp = sapply(allyearmodel, '[', 13)
w = sapply(allyearmodel, '[', 14)
xx = data.frame(Model = ym, Year = yeaar, Make = maake, VIN = v, Price = p, Mileage = m,
ExteriorColor = eC,
           InteriorColor = iC, Transmission = t, Engine = d, Company = n, Address = a, Phone = pp,
Web = w)

#writes to csv
write.csv(xx, file = "Cars.csv")

#NUMBER 2

library("XML")
library(RCurl)
library('tidyverse')
library('xml2')
library(stringr)

#read html of the link
link = read_html("http://anson.ucdavis.edu/~mueller/cveng13.html")
#find all the p tags
p_tags = xml_find_all(link, "//p")
#subset until the first row of publications
newp_tags = p_tags[12:272]
#gets the url href tags
url = xml2::xml_find_first(newp_tags, ".//a/@href")
#translate to text
url = xml_text(url)
#all journels are italized and start with the em tag
journal = xml_find_first(newp_tags, ".//em")
#translate to text
journal = xml_text(journal)
#all volume is bolded
volume = xml_find_first(newp_tags, ".//strong")
#translate to text
volume = xml_text(volume)
#convert the ptags to text for extractions without tags
newp_tags = xml_text(newp_tags)
```

```
webscape = sapply(1:261, function (i) {

  #gets the locations of the years of publications
  datelocations = regexpr('[0-9]{4}', newp_tags[i])
  #extract publication dates from its locations
  date = regmatches(newp_tags[i], datelocations)

  #gets the arthors until ( which the start of the dates
  ar = str_extract(newp_tags[i], "[A-Za-z.,*üéöŞ\\- ]*\\(")
  #replace ( with nothing)
  arthor = gsub(" \\(", "", ar)

  #gets the title of publication through characters, numbers, and characters until a period
  title = str_extract(newp_tags[i], "\\).[éA-Za-z0-9- \\\n(),;:\"\'.?!%&~ ]*\\.")
  #replace period with nothing
  title1 = gsub("\\).[ \\\n]*", "", title)

  #returns as lists
  return (cbind(date, arthor, title1))
})

#convert to dataframe
d = sapply(webscape, '[', 1)
a = sapply(webscape, '[', 2)
t = sapply(webscape, '[', 3)
xx = data.frame(Date = d, Authors = a, Title = t)

#binds the journal, volume, url together
test = cbind(Journal = journal, Volume = volume, URL = url)
#creates data frame
tester = as.data.frame(test)
#bind all together
mueller = cbind(xx, tester)

newmueller = mueller[-c(14, 15, 51, 52, 109, 110, 151, 152, 153, 188, 189, 220, 221, 222, 260,
261), ]

totalNAURL = sum(is.na(newmueller$URL)) #total NA for URL
totalNAVolume = sum(is.na(newmueller$Volume)) #total NA for Volume
totalNAJournal = sum(is.na(newmueller$Journal))

#gets all dates
alldates = factor(levels(newmueller$Date))
numperdate = sapply(1:length(alldates), function(i){
```

```
  num = length(which(newmueller$Date == factor(levels(newmueller$Date))[i]))
  #gets length when a date in the dataframe is equal to dates and counts them
  return (num)
})
#binds the dates and num of publications
datenum = rbind(alldates, numperdate)

num_author = sapply(1:245, function(i){
  numa = str_count(newmueller$Authors[i], '\\,') #counts all the commas
  num_coauth = ceiling(numa/2) #divides by 2 and get the ceiling because each name has 2
commas
  #ceiling is used because solo names or last person only has 1 comma.
  #ceiling will count them as 1 rather than 0.5

  return (num_coauth)
})
average_author = mean(num_author) #average num of coauthors

#finds unique group of Authors
unique_groupsauth = length(unique(newmueller$Authors))

#finds the number of publications in these journals
numPub = table(newmueller$Journal)
write.csv(newmueller, file = "muellerweb.csv")

save(newmueller, file = "webscrape.rda")
```