STA 141A Homework 4
Janet Loyola

The codes and results derived by using these codes constitute my own work. I have consulted the following resources regarding this assignment: Dhanya Chandrasekhar, Stephanie Lam, and Juliana Noronha

Q1.

CarAdvert.zip file consists of 1531 posts on advertisements of used vehicles by several vendors. We were asked to extract some relevant information and create a data frame in R to represent the information. Some of the information that I extracted were (i) Model (Year and Make) of the car (ii) Vehicle Identification Number (VIN) (iii) Price (iv) Mileage (v) Color (Interior and Exterior) (vi) Transmission (vii) Engine displacement (in liters) (viii) Name of company selling the car (ix) Street address of the company (x) Phone number of the company (xi) Website of the company. Through R, we see that some of the information is not provided. I dealt with it by providing those spaces empty.

Some of the interpretations that I make after looking at the .csv file are as follows:

- ➔ Many of these cars are black both interior as well as exterior.
- ➔ Most of these cars are automatic.
- ➔ The addresses of these car companies are in a few states. They are MA, NY, NJ, CA, IL, CO, and NV. This makes me wonder why this dataset has chosen only these few states. Is it possibly because these are major car producing states? Can it also be because the main transportation in these states is by car?
- ➔ Depending on the brand of the car and the year, the prices are made. For example: Nissan Altima 2.5 (2008 model) is being sold for $1,999 and Nissan Altima 2.5 (2010 model) is being sold for $4,499. Why such a difference in price? Are there any other factors that can influence the change in prices?
- ➔ Mileage is an important thing that people take into consideration before buying a used car. Between two same cars (same year, model, make, color), the seller with the more car mileage usually sells his car for a cheaper price.
- ➔ One interesting thing that I found is depending on the place, people sell cars. For example: Jamaica has a mountainous terrain. Hence, most of the cars sold here are jeeps, SUV’s, or cars with really powerful engines like Audi cars.
- ➔ Why is the VIN important? VIN cloning is a scam where sellers replace the VIN of a stolen car with one that is legally registered. This type of fraud can easily be avoided by decoding the VIN of the vehicle.
- ➔ The make and the model of the car help people do a bit of detective work on industry and consumer reviews to uncover possible defects or even common problems.
- ➔ To ensure that you are being charged a fair price, make sure to compare prices for the same make, model and year with several sources. Checking the Blue Book values and dealer prices can easily be done online. Even though condition and mileage will play a role in price, you can still get a good ballpark figure of the going price.

From CarAdvert.csv, we see that the extracted information is really important for people to buy a used car. This information helps people do more research on the car that they wish to buy as mentioned above.

How I went about writing the code for this?

Initially, I just played around with the gsub() function. This helped me extract all the required information per text file. But I had to extract the information for all the text files. The next step was to create the gettoken() function. This function has 3 parameters- exp, lines and extra=",". Its function is to check for the required information line by line and store it in a variable v given the pattern was matched. Then I created the parselines() function that included the gettoken() function. This function basically extracted all the required information for all the text files together. Finally, I just wrote all the information in the form of a .csv file.

Q2.

Using the XML package and text processing, we extracted required information from the webpage http://anson.ucdavis.edu/~mueller/cveng13.html. Our task was to create an R data frame containing the following information for each of the journal publications:
(i) Year of publication.
(ii) Authors.
(iii) Title of publication.
(iv) Journal title.
(v) Journal volume.
(vi) URL for the publication.

We also did a basic statistical summary of the dataframe including information such as number of publications per year, number of co-authors, number of publications in different journals, etc.

After getting the dataframe, we tried figuring out some basic summaries. We found the general summary of the overall publication from the dataframe. Now, we wanted to get into a little more depth and see the summary of each information/factor like Year of publication.

From the summary for years, we see that the maximum numbers of his publications (13) were done in 2005 and least publication (1) was done in 1986.

From the summary on authors, we see the number of publications written by the authors separately and with others. For example: Müller, H.G. and Stadtmüller, U wrote one publication-Detecting dependencies in smooth regression models in 1988.

From the summary on the URL for the publications, we see that each publication has its own separate URL.

From the summary on the title of publication, we see that each title of publication is unique.

From the summary on the journal title, we see that some of the journals have more than one same journal title. For example: Change-point Problems and smoothing techniques for curve estimation both have two publications under the same journal title. Similarly, Annals of Human Biology have three publications under the same journal title.

From the summary on journal volume, we see that each journal volume has in between one and five publications.

How I went about writing the code for this?

Initially, I viewed the webpage by its page source and tried to find a pattern for each of the information. For example: Between <em> and </em>, you have the journal title. Similarly, I tried finding out the patterns for the rest too. After that, I played around with the gsub() function like in question 1 to extract the relevant information. As the task is to create a dataframe, I created an empty dataframe with the names of the columns as year, authors, URL, publish, journal, journal_volume. So, it basically created an empty dataframe of 6 columns and zero rows. Then I created the actual dataframe inside a for loop with all the required information. I also wrote cases for when the data entries are NULL inside the for loop. That's how I created the dataframe. Then, I just provided basic statistical summaries based on the dataframe.

CODE APPENDIX:

```
#Q1
```

```
#library(foreign)
#setwd("/Users/janetloyola/Downloads/CarAdvert")
#files <-list.files()
#data <- 0


#for (f in files) {
#  tempData = scan( f, what="character")
# data <- c(data,tempData)
#}

#sentence = paste(data,collapse=" ")
#sentence

#fileName="1.txt"
#con=file(fileName,open="r")
#line=readLines(con)
#long=length(line)
#for (i in line){
# print(i)
#}
#close(con)
setwd("/Users/janetloyola/Downloads")

#just practicing extracting parts that I need
line[1] #Model of the car
gsub('VIN: (.*)Stock.*','\\1',line[3],perl=TRUE)  ##(.*) acts like a buffer
gsub('VIN: (.*)Stock.*UsedMileage: (.*)MPG.*','\\1,\\2',line[3],perl=TRUE)
gsub('VIN: (.*)Stock.*UsedMileage: (.*)MPG.*Exterior:
(.*)Interior.*','\\1,\\2,\\3',line[3],perl=TRUE)
gsub('VIN: (.*)Stock.*UsedMileage: (.*)MPG.*Exterior: (.*)Interior:
(.*)Body.*','\\1,\\2,\\3,\\4',line[3],perl=TRUE)
gsub('.*\\?\\?\\?.*\\?\\?\\? (.*)','\\1',line[2])
gsub('.*CoupeEngine: (.*L) .*','\\1',line[3]) #Engine
#Name of company selling the car
gsub('Offered by: (.*) \\?\\?\\?.*\\?\\?\\?.*','\\1',line[2])
#Street address of the company
gsub('.*Address: (.*)Phone.*','\\1',line[32])
#Phone number of the company
gsub('.* \\?\\?\\? (.*) \\?\\?\\?.*','\\1',line[2])
#Website of the company
gsub('Website: (.*)/Address.*','\\1',line[32])

#Model of the car
model=line[1]
#Vehicle Identification Number (VIN)
vin=gsub('VIN: (.*)Stock.*','\\1',line[3],perl=TRUE)
#Price
price=gsub('.*\\?\\?\\?.*\\?\\?\\? (.*)','\\1',line[2])
```

```
#Mileage
mileage=gsub('.*UsedMileage: (.*)MPG.*','\\1',line[3],perl=TRUE)
#Color (Interior and Exterior)
color=gsub('.*Exterior: (.*)Interior: (.*)Body.*','\\1,\\2',line[3],perl=TRUE)

#Engine displacement (in liters)
engine=gsub('.*CoupeEngine: (.*)L.*','\\1',line[3])
#Name of company selling the car
name=gsub('Offered by: (.*) \\?\\?\\?.*\\?\\?\\?.*','\\1',line[2])
#Street address of the company
address=gsub('.*Address: (.*)Phone.*','\\1',line[32])
#Phone number of the company
number=gsub('.* \\?\\?\\? (.*) \\?\\?\\?.*','\\1',line[2])
#Website of the company
website=gsub('Website: (.*)/Address.*','\\1',line[32]
###EXTRA:::
interior=gsub('VIN: (.*)Stock.*UsedMileage: (.*)MPG.*Exterior: (.*)Interior:
(.*)Body.*','\\4',line[3],perl=TRUE)
#interior
#[1] " Black"
##Creating a function to use inside another function parsellines.
gettoken=function(exp,lines,extra=",") {
 v=""
 for(line in lines){
  p=regexpr(exp, line) #searches for matches to a pattern.
  if(p[1]>0){ ##Then matched
   v=(gsub(exp,'\\1',line))
   break
  }
 }
 cat(c('"',v,'"'),file=CarAdvert)
 cat(extra,file=CarAdvert)
 return(v)
}
##Extracting the parts from the lines that we need
parselines = function(lines) {
 year=gettoken('([^ ]+) .*',lines)
 #print(year)
 model=gettoken('[^ ]+ (.*)',lines)
 #print(model)
 vin=gettoken('VIN: (.*)Stock.*',lines)
 #print(vin)
 price=gettoken('.*\\?\\?\\?.*\\?\\?\\? (.*)',lines)
 #print(price)
 mileage=gettoken('.*UsedMileage: (.*)MPG.*',lines)
 #print(mileage)
 ecolor=gettoken('.*Exterior: (.*)Interior: .*Body.*',lines)
 icolor=gettoken('.*Exterior: .*Interior: (.*)Body.*',lines)
 #print(ecolor)
 #print(icolor)
```

```
  engine=gettoken('.*Engine: (.*)L.*',lines)
  #print(engine)
  name=gettoken('Offered by: (.*) \\?\\?\\?.*\\?\\?\\?.*',lines)
  #print(name)
  address=gettoken('.*Address: (.*)Phone.*',lines)
  #print(address)
  number=gettoken('.* \\?\\?\\? (.*) \\?\\?\\?.*',lines)
  #print(number)
  website=gettoken('Website: (.*)Address.*',lines,extra="\n")
  #print(website)
}
##Writing the information in the form of a .csv file. Tried it above for the case of 1.txt file
CarAdvert=file("CarAdvert.csv", "w")
zipname="CarAdvert.zip"
zipfiles=unzip(zipname,list=TRUE)[,1]
for (i in 2:length(zipfiles)) {
  f=zipfiles[i]
  con=unz(zipname,f)
  lines=readLines(con)
  close(con)
  #cat(f,file=CarAdvert)
  #cat(",",file=CarAdvert)

  parselines(lines)
}
close(CarAdvert)

#Q2
install.packages("XML")
library(XML)
hurl = 'http://anson.ucdavis.edu/~mueller/cveng13.html'
hnode = htmlTreeParse ( hurl , useInternalNodes = TRUE , trim=TRUE )

allem=getNodeSet(hnode,'//*/p/em')  ##Want everything from p to em for all publications

df=data.frame(matrix(NA, nrow = 0, ncol = 6)) ##creating empty dataframe of 6 columns
with their names written below.
colnames(df)=c("year","authors","URL","publish","journal", "journal_volume")

#length(allem)=239. So runs the loop from 1 to 239
for (i in 1:length(allem)) {
  em=allem[[i]]
  #print(em)
  par=xmlParent(em)
  authtext=xpathSApply(par, "text()[1]", xmlValue)[[1]]
  ##extracting the required part (year,authors,etc)
  year=gsub('.*\\((....)\\).*', '\\1', authtext)
  authors=gsub('(.*)\\(....\\).*', '\\1', authtext)
  anc=getNodeSet(par,'a')
  ## few entries dont have publish URL
  if (length(anc)>0)
    puburl=xmlGetAttr(anc[[1]], "href")
  else
    puburl=""
  ##pub stands for publication title
  pub=gsub('.*\\(....\\)(.*)', '\\1', authtext)
  jtitle=xpathSApply(em, "text()", xmlValue)
  if (is.null(jtitle))  ##telling what to do if the journal title has null values
    jtitle=""
  jvolume=xpathSApply(par, "strong/text()", xmlValue)
  if (is.null(jvolume))
    jvolume=""
  df=rbind(df, data.frame(year=year, authors=authors, puburl=puburl, pub=pub, jtitle=jtitle,
jvolume=jvolume))
  ##creating a dataframe
}
#General analysis from the coded dataframe are written below.
summary(df)
##It gives a general summary of the overall publication dataframe
table(df$year)
##From this, we can see that maximum number of his publications (13) were done in 2005
and least publication (1) was done
#in 1986.
mean(table(df$year))
#We see that on average he does 7 publications per year. However, this seems to be a
wrong estimation because of the
#many outliers.
table(df$authors)
##It tells the number of publications written by the authors seperately and with others.
#For ex:Müller, H.G. and Stadtmüller, U (1988) wrote one publication-Detecting
dependencies in smooth regression models.
table(df$puburl)
#From this, we can see that each publication has its own seperate URL.
table(df$pub)
#Each title of publication is unique.
table(df$jtitle)
#Some of the journals have more than one same journal title.EX:Change-point Problems and
Smoothing techniques for curve
#estimation both have 2 publications under the same journal title.Similarly, Annals of
Human Biology have 3
#publications under the same journal title
table(df$jvolume)
#We can see that a journal volume has in between 1 and 5 publications under the same
journal volume.
```