homework two

Yi (Chris) Chen

October 2, 2017

homework two

Name: Yi Chen

UNI:yc3356

Email:yc3356@columbia.com

Goals: regular expressions, character functions in R, and web scraping. In this assignment, we're going to scrape the 2017-2018 Brooklyn Nets Regular Season Schedule (they're a basketball team from Brooklyn that plays in the NBA). We will take the regular season schedule from http://www.espn.com/nba/ and reassemble the game listings in an R data frame for computational use. To do this, perform the following tasks:

i.Use the readLines() command we studied in class to load the NetsSchedule.html file into a character vector in R. Call the vector nets1718.

```
setwd("C:/Users/cheny/Desktop/study/statistical computing and intro to data
science/homework/homework two")
nets1718 <- readLines("NetsSchedule.html",warn = FALSE)
# test whether the file has been read.
head(nets1718,3)

## [1] "<!DOCTYPE html>"
## [2] "<html xmlns:fb=\"http://www.facebook.com/2008/fbml\">"
## [3] "<head><script
src=\"http://cdn.espn.com/sports/optimizely.js\"></script><meta
charset=\"iso-8859-1\">"

tail(nets1718,3)

## [1] "  //]]>"          "</script></body>" "</html>"
```

a. How many lines are in the NetsSchedule.html file?

```
# there are two method to calculate the number of lines in this file
summary_nets1718_1 <- summary(nets1718)
summary_nets1718_2 <- length(nets1718)
#test
summary_nets1718_1[1] == summary_nets1718_2

## Length
##   TRUE

cat('there are',summary_nets1718_1[1],'lines in NetSchedule.html file' )
```

```
## there are 828 lines in NetSchedule.html file
```

b. What is the total number of characters in the file?

```r
# split the whole file in order to calculate the number of characters
total_characters <- strsplit(nets1718, split = "")
number_of_each_line <- vector()
# go through every line in the file and calculate the number of characters in each line
for ( i in 1:length(total_characters)){
        number_of_this_line <- length(total_characters[[i]])
        number_of_each_line <- c(number_of_each_line,number_of_this_line)
}
total_characters_number_1 <- sum(number_of_each_line)

# this problem can also be solved in this way
total_characters_number_2 <- sum(nchar(nets1718))
#test
total_characters_number_1 == total_characters_number_2
```

```
## [1] TRUE
```

```r
cat('there are total',total_characters_number_1,'characters in this file.')
```

```
## there are total 129188 characters in this file.
```

c. What is the maximum number of characters in a single line of the file?

```r
# find the maxmum number of characters
maximum_number <- max(number_of_each_line)
# find in which line that have the maximum number
which_line <- which(number_of_each_line == maximum_number)
cat("the maximum number of characters in a single line is: ",
maximum_number," in the", which_line,"-th line of the file.")
```

```
## the maximum number of characters in a single line is:  9736  in the 485 -
th line of the file.
```

ii. Open NetsSchedule.html as a webpage. This should happen if you simply click on the file. You should see a table listing all the games scheduled for the 2017-2018 NBA season. There are a total of 82 regular season games scheduled. Who and when are they playing first? Who and when are they playing last?

```r
# all the information is just get through wesite
cat("They will have their first game on Oct 18(Wed,7:00 PM) against Indiana
Pacers.")
```

```
## They will have their first game on Oct 18(Wed,7:00 PM) against Indiana
Pacers.
```

```r
cat("They will have their first game on Apr 11(Wed,8:00 PM) against Boston
Celtics.")
```

```
## They will have their first game on Apr 11(Wed,8:00 PM) against Boston
Celtics.
```

iii. Now, open NetsSchedule.html using a text editor. To do this you may need to rightclickon the file and tell your computer to use a text editor to open the file. Whatline in the file holds information about the first game of the regular season (date, time,opponent)? What line provides the date, time, and opponent for the final game? It may be helpful to use CTRL-F or COMMAND-F here and also work between the file in R and in the text editor.

```
# search for the date
date_search <- grep('Apr 11',nets1718)

# search for the time
time_search <- grep('8:00 PM',nets1718)

# search for the opponent
opponent_search <- grep('boston-celtics',nets1718)

# give the line that provide all of these information:
final_game <- intersect(date_search,intersect(time_search,opponent_search))

cat('all the infroamtion about the final game is in the:', final_game,'line')

## all the infroamtion about the final game is in the: 402 line

nets1718[final_game]

## [1] "<td colspan=\"4\"><a name=\"&lpos=nba:team:schedule:tickets\"
href=\"https://www.vividseats.com/nba-basketball/brooklyn-nets-tickets/nets-
vs-bulls-4-9-2431213.html?wsUser=717\">3,446 available from
$24</a></td></tr><tr class=\"evenrow team-46-2\"><td>Wed, Apr 11</td><td><ul
class=\"game-schedule\"><li class=\"game-status\">@</li><li class=\"team-
logo-small logo-nba-small\"><a
href=\"http://www.espn.com/nba/team/_/name/bos/boston-celtics\"><img
src=\"http://a.espncdn.com/combiner/i?img=/i/teamlogos/nba/500/scoreboard/Bos
.png&h=80&w=80\"></a></li><li class=\"team-name\"><a
href=\"http://www.espn.com/nba/team/_/name/bos/boston-
celtics\">Boston</a></li></ul></td><td>8:00 PM</td><td style=\"text-
align:center;\">&nbsp;</td>"
```

iv. Write a regular expression that will capture the date of the game. Then using the grep() function nd the lines in the file that correspond to the games. Make sure that grep() finds 82 lines, and the first and last locations grep() finds match the first and last games you found in (ii).

```
exp <-"><td>[MTWFS][a-z]+\\,\\s[A-Z][a-z]+\\s[0-9]+</td><td>"
lines_number <- grep(exp,nets1718)
#test
length(lines_number) == 82

## [1] TRUE
```

```
grepl('Wed, Oct 18',nets1718[lines_number[1]]) & grepl("Wed, Apr
11",nets1718[lines_number[length(lines_number)]])

## [1] TRUE
```

v. Using the expression you wrote in (v) along with the functions regexp() and regmatches(),

extract the dates from the text file. Store this information in a vector called date to save to use below. HINT: We did something like this in class.

```
date_exp <- regexpr("><td>[MTWFS][a-z]+\\,\\s[A-Z][a-z]+\\s[0-9]+</td><td>",
nets1718)
date <- regmatches(nets1718,date_exp)
date <- substring(date, 6, nchar(date) - 9)
head(date,5)

## [1] "Wed, Oct 18" "Fri, Oct 20" "Sun, Oct 22" "Tue, Oct 24" "Wed, Oct 25"

#test
length(date)

## [1] 82

head(date,1) == "Wed, Oct 18" & tail(date,1) == "Wed, Apr 11"

## [1] TRUE
```

vi. Use the same strategy as in (v) and (vi) to create a time vector that stores the time of the game.

```
time_exp <- regexpr("</td><td>[0-9][0-9]?\\:[0-9][0-9]\\s[AP][M]</td>",
nets1718)
time <- regmatches(nets1718,time_exp)
time <- substring(time, 10, nchar(time) - 5)
head(time,5)

## [1] "7:00 PM" "7:30 PM" "3:30 PM" "7:00 PM" "7:30 PM"

#test
length(time) == 82

## [1] TRUE

head(time,1) == "7:00 PM" & tail(time,1) == "8:00 PM"

## [1] TRUE
```

vii. We would now like to gather information about whether the game is home or away. This information is indicated in the schedule by either an @ or a vs in front of the opponent. If the Nets are playing @ their opponent's court, the game is away. If the Nets are playing vs the opponent, the game is at home. Capture this information using a regular expression. You may want to use the HTML code around these values to guide your search. Then extract this information and use it to create a vector called home which takes the value 1 if the game is played at home or 0 if it is away.

```
# create the regular express for the home and away information
home_exp <- '>\\@</li><li'
away_exp <- '>vs</li><li'
# replace the way to indecate home or away information
nets1718 <- gsub(home_exp,'>1</li><li', nets1718)
nets1718 <- gsub(away_exp,'>0</li><li', nets1718)
# search for the home and away infroamtion and create the vector home
home_exp <- regexpr(">[01]</li><li", nets1718)
home <- regmatches(nets1718,home_exp)
home <- substring(home,2, 2)
head(home,5)

## [1] "1" "0" "0" "1" "0"

# test
length(home)

## [1] 82

home[1] == '1' & home[length(home)] == '1'

## [1] TRUE
```

viii. Finally we would like to find the opponent, again capture this information using a regular expression. Extract these values and save them to a vector called opponent. Again, to write your regular expression you may want to use the HTML code around the names to guide your search.

```
opponent_exp <- regexpr(">[A-Z]*[a-z]*\\s?[A-Z]?[a-z]*</a></li></ul></td>",
nets1718)
opponent     <- regmatches(nets1718,opponent_exp)
opponent     <- substr(opponent,2,nchar(opponent)-19)
head(opponent,5)

## [1] "Indiana"   "Orlando"   "Atlanta"   "Orlando"   "Cleveland"

# test
length(opponent)

## [1] 82

opponent[1] == "Indiana" & opponent[length(opponent)] == "Boston"

## [1] TRUE
```

ix. Construct a data frame of the four variables in the following order: date, time, opponent, home. Print the frame from rows 1 to 10 Does the data match the first 10 games as seen from the web browser?

```
result <- data.frame(date=date,time=time,opponent=opponent,home=home)
head(result,10)

##           date     time    opponent home
## 1  Wed, Oct 18  7:00 PM     Indiana    1
## 2  Fri, Oct 20  7:30 PM     Orlando    0
## 3  Sun, Oct 22  3:30 PM     Atlanta    0
## 4  Tue, Oct 24  7:00 PM     Orlando    1
## 5  Wed, Oct 25  7:30 PM   Cleveland    0
## 6  Fri, Oct 27  7:30 PM   NY Knicks    1
## 7  Sun, Oct 29  6:00 PM      Denver    0
## 8  Tue, Oct 31  7:30 PM     Phoenix    0
## 9   Fri, Nov 3 10:30 PM Los Angeles    1
## 10  Mon, Nov 6  9:00 PM     Phoenix    1

# the first 10 records are as same as the information on the web
```