Hwk3_bs2996

Bangda Sun

October 3, 2016

i. Use the `readLines()` command we studied in class to load the NetsSchedule.html file into a character vector in *R*. Call the vector nets1617.

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

```
nets1617 <- readLines("C:\\Users\\Bangda\\Desktop\\GR5206\\Hwk3\\NetsSchedule.html")
# number of lines in html file
length(nets1617)
```

```
## [1] 811
```

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

```
# total number of characters
sum(nchar(nets1617))
```

```
## [1] 127835
```

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

```
# max number of characters in a single line
max(nchar(nets1617))
```

```
## [1] 7211
```

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 2016-2017 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?

Answer: First game is playing against Boston Celtics at Wednesday, Oct 26, 7:30 PM(ET); Last game is playing against Chicago Bulls at Wednesday, Apr 12, 8:00 PM(ET).

iii. Now, open NetsSchedule.html using a text editor. To do this you may need to right-click on the file and tell your computer to use a text editor to open the file. What line 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.

```
Answer: The 315th line holds the information of the first game. The 396th line holds the information of
```

Using **NetsSchedule.html** we'd like to extract the following variables: the date, the game time (ET), the opponent, and whether the game is home or away. Looking at the file in the text editor, locate each of these variables. For the next part of the homework we use regular expressions to extract this information.

iv. Write a regular expression that will capture the date of the game. Then using the `grep()` function find 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).

```
date.info <- "[A-Z]{1}[a-z]{2}[[:punct:]]\\s[A-Z]{1}[a-z]{2}\\s[0-9]{1,2}"
date.data <- grep(nets1617, pattern = date.info)
# the first game
date.data[1]
```

```
## [1] 315
```

```
date.data[length(date.data)]
```

```
## [1] 396
```

v. Using the expression you wrote in (iv) along with the functions `regexpr()` 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.

```
# find the full match expression
date <- regmatches(nets1617, regexpr(date.info, nets1617))
head(date)
```

```
## [1] "Wed, Oct 26" "Fri, Oct 28" "Sat, Oct 29" "Mon, Oct 31" "Wed, Nov 2"
## [6] "Fri, Nov 4"
```

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

```
# hours comes first and then ':' and minutes, then comes the 'PM', since all games time will be in 'PM'
time.info <- "[0-9]{1}[:][0-9]{2}\\s[P][M]"
time <- regmatches(nets1617, regexpr(time.info, nets1617))
head(time)
```

```
## [1] "7:30 PM" "7:30 PM" "8:00 PM" "7:30 PM" "7:30 PM" "7:30 PM"
```

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.

```
# first match '<li class="game-status">' and follows '@' or 'v'
home.info <- '[<][i-l]{2}\\s[a-z]{5}[=]\\"[a-z]{4}[-][a-z]{6}\\"[>][@|v]'
home <- regmatches(nets1617, regexpr(home.info, nets1617))
head(home)
```

```
## [1] "<li class=\"game-status\">@" "<li class=\"game-status\">v"
## [3] "<li class=\"game-status\">@" "<li class=\"game-status\">v"
## [5] "<li class=\"game-status\">v" "<li class=\"game-status\">v"
```

```
# substring the last character and if it is 'v' then it is true
home <- (substr(home, nchar(home[1]), nchar(home[1])) == "v")
```

```
# convert TRUE and FALSE to 1 and 0
home <- as.numeric(home); head(home)
```

```
## [1] 0 1 0 1 1 1
```

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.

```
# similariy we can find the character before the opponents' name, it is the website with opponents' nam
# 1) single word like 'Chicago', with website name 'chicago-bulls'
# 2) two words like 'Golden State', with website name 'golden-state'
# 3) Piladelphia: it's website name contains numbers '76'
opponent.info <- '/[a-z]*[-]*[a-z]+[-][6-7]*[a-z]+\"[>][A-Z]{1,2}\\s*[A-Z]*[a-z]*'
opponent <- regmatches(nets1617, regexpr(opponent.info, nets1617))
head(opponent)
```

```
## [1] "/brooklyn-nets\">Clubhouse"   "/boston-celtics\">Boston"
## [3] "/indiana-pacers\">Indiana"    "/milwaukee-bucks\">Milwaukee"
## [5] "/chicago-bulls\">Chicago"     "/detroit-pistons\">Detroit"
```

```
tail(opponent)
```

```
## [1] "/philadelphia-76ers\">Philadelphia"
## [2] "/orlando-magic\">Orlando"
## [3] "/chicago-bulls\">Chicago"
## [4] "/boston-celtics\">Boston"
## [5] "/chicago-bulls\">Chicago"
## [6] "/brooklyn-nets\">Clubhouse"
```

```
# check the number of rows and find that the first and last row is nets itself that should be removed
length(opponent)
```

```
## [1] 84
```

```
opponent <- opponent[-1]
opponent <- opponent[-length(opponent)]
# substring the name of opponents from the previous string
opponent.info2 <- '[A-Z]{1,2}[a-z]*[-]*[6-7]*[a-z]*\\s*[A-Z]*[a-z]*'
opponent <- regmatches(opponent, regexpr(opponent.info2, opponent))
head(opponent)
```

```
## [1] "Boston"    "Indiana"   "Milwaukee" "Chicago"   "Detroit"   "Charlotte"
```

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?

```
net.data <- data.frame(Date = date, Time = time, Opponent = opponent, Home = home)
net.data[1:10,]
```

```
##           Date    Time  Opponent Home
## 1  Wed, Oct 26 7:30 PM    Boston    0
## 2  Fri, Oct 28 7:30 PM   Indiana    1
## 3  Sat, Oct 29 8:00 PM Milwaukee    0
```

```
## 4  Mon, Oct 31 7:30 PM   Chicago    1
## 5   Wed, Nov 2 7:30 PM   Detroit    1
## 6   Fri, Nov 4 7:30 PM Charlotte    1
## 7   Tue, Nov 8 7:30 PM Minnesota    1
## 8   Wed, Nov 9 7:00 PM NY Knicks    0
## 9  Sat, Nov 12 9:00 PM   Phoenix    0
## 10 Mon, Nov 14 0:30 PM        LA    0
```

```
# we can see that this data match the first 10 games as seen from the web browser
```