Homework 3 Solutions

i.

Open the link http://www.espn.com/nba/team/schedule/_/name/BKN/seasontype/2. Display the source code and copy and paste this code into a text editor. Then save the file as `NetsSchedule1819` using a .html extension. Once the file is saved, check that you can open the file and it displays the 2018-2019 Brooklyn Nets Regular Season Schedule.

ii.

```
setwd("/Users/linxiliu/Dropbox/Teaching/GR5206/Homework/Homework 3 Web scraping")
nets1819 <- readLines("NetsSchedule1819.html")
```

```
## Warning in readLines("NetsSchedule1819.html"): incomplete final line found
## on 'NetsSchedule1819.html'
```

The number of lines in the file corresponds to the length of the vector *nets1819*.

```
length(nets1819)
```

```
## [1] 106
```

I can find the number of characters in each line of the file by running *nchar(nets1819)* since *nchar()* vectorizes. This will return a vector of length 106 with each element telling the number of characters in the corresponding line of the file. Then we can take a sum of these values to give the total number of characters.

```
sum(nchar(nets1819))
```

```
## [1] 462597
```

Finally, I can use the *max()* command, with *nchar(nets1819)* as its input, to find the maximum number of characters in any line of the code.

```
max(nchar(nets1819))
```

```
## [1] 249820
```

iii. In the first game of the regular season, the Nets are playing the Detroit Piston in Detroit Wednesday, October 17 at 7:00PM. In the last game of the season, the Nets are playing the Miami Heat in Brooklyn on Wednesday, April 10 at 8:00PM.

iv. It's line 64.

v. I use a regular expression to search for a capital letter, followed by two lowercase letters, a comma, a space, a capital letter, two lowercase letters, a space, and then one or more digits. This regular expression is found in *date_exp*. Then I use *grep()* to search *nets1819* for lines with dates in them. These lines are stored in *game.lines*. Looking at *game.lines* I see information on the first and last games.

```
date_exp <- "[A-Z][a-z]{2},\\s[A-Z][a-z]{2}\\s[0-9]+"
game.lines <- grep(date_exp, nets1819)
game.lines[1]
```

```
## [1] 64
```

```
choosen_line=nets1819[game.lines[1]]
```

vi.

```
line82 <- strsplit(choosen_line, split="</use></svg></a></td></tr><tr")[[1]]
length(line82)
```

```
## [1] 82
```

vii

```
length(grep(date_exp, line82))
```

```
## [1] 82
```

```
regmatches(line82[1], regexpr(date_exp,line82[1]))
```

```
## [1] "Wed, Oct 17"
```

```
regmatches(line82[82], regexpr(date_exp,line82[82]))
```

```
## [1] "Wed, Apr 10"
```

viii. *gregexpr()* returns the starting locations and the lengths of each of the game dates, then we can actually extract the information using *regmatches()*. Since the output of *regmatches()* is a list, we use the *unlist()* command to turn it into a vector.

```
date.locations <- gregexpr(date_exp, line82)
date <- regmatches(line82, date.locations)
date <- unlist(date)
```

ix. Extracting the game times is similar to extracting the dates, but now my regular expression searches for one or more digits followed by a colon, 2 digits, a space, and then either AM or PM.

```
time_exp <- "[0-9]+:[0-9]{2} (PM|AM)"
time.locations <- gregexpr(time_exp, line82)
time <- regmatches(line82, time.locations)
time <- unlist(time)
```

x. In my solution, I use the fact that in each line, the string `<div class=f̈lex items-center opponent-logo>̈<span class=p̈r2>̈` appears before the home or away information. So my regular expression searches for `<div class=f̈lex items-center opponent-logo>̈<span class=p̈r2>̈` followed by ‘@’ or `<div class=f̈lex items-center opponent-logo>̈<span class=p̈r2>̈` followed by ‘vs’. As in part (v) and (vi) I use *gregexpr()* and *regmatches()* to actually extract the strings which match the regular expression. Since these strings include `«div class=f̈lex items-center opponent-logo>̈<span class=p̈r2>̈` before ‘@’ or ‘vs’, I then use the *substr()* command just the ‘@’ or the ‘vs’. Finally, I create the *home* vector from this information.

```
away_exp <- "<div class=\"flex items-center opponent-logo\"><span class=\"pr2\">@|<div class=\"flex ite
away.locations <- gregexpr(away_exp, line82)
away <- regmatches(line82, away.locations)
away <- substring(away, 64, nchar(away))
home <- rep(1, length(away))
home[away == "@"] <- 0
```

xi. In my solution, I use the fact that in each line, the string `<img`
`salt=.̤+¨`
`stitle=` appears before the opponent.

```
opponent_exp <- "<img\\salt=\".+\"\\stitle="
opponent.locations <- gregexpr(opponent_exp, line82)
opponent <- regmatches(line82, opponent.locations)
opponent <- unlist(opponent)

opponent <- substr(opponent, 11, nchar(opponent)-8)
```

xii.

```
schedule <- data.frame(date, time, opponent, home)
schedule[1:10,]
```

```
##           date    time     opponent home
## 1  Wed, Oct 17 7:00 PM      Detroit    0
## 2  Fri, Oct 19 7:30 PM     New York    1
## 3  Sat, Oct 20 7:00 PM      Indiana    0
## 4  Wed, Oct 24 7:00 PM    Cleveland    0
## 5  Fri, Oct 26 8:00 PM  New Orleans    0
## 6  Sun, Oct 28 5:00 PM Golden State    1
## 7  Mon, Oct 29 7:30 PM     New York    0
## 8  Wed, Oct 31 7:30 PM      Detroit    1
## 9   Fri, Nov 2 7:30 PM      Houston    1
## 10  Sun, Nov 4 6:00 PM Philadelphia    1
```