HW_3

Ming Zhong mz2692

10/7/2018

i.

Open the linkhttp://www.espn.com/nba/team/schedule/_/name/BKN/seasontype/2. Save the page asNetsSchedule1819using a .html extension. Once the file is saved,check that you can open the file by a text editor or import it inR.

```
#readline
nets1819=readLines("/Users/zhongming/Desktop/assignment/intro to ds/NetsSchedule1819.html",warn=F)
```

ii.

Use thereadLines()command we studied in class to load theNetsSchedule1819.htmlfile into a character vector inR. Call the vectornets1819.

```
length(nets1819)#number of lines
```

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

```
sum(nchar(nets1819))#total of character
```

```
## [1] 462591
```

```
max(nchar(nets1819))#max of character
```

```
## [1] 249689
```

a.There are 106 lines.
b.The total number of characters are 462591
c.The maximum number of characters are 249689

iii.

Open the webpage. You should see a table listing all the games scheduled for the 2018-2019 NBA season. There are a total of 82 regular season games scheduled. Who andwhen are they playing first? Who and when are they playing last?
Ans:
Who andwhen are they playing first?
Detroit Wed,Oct17,7:00 PMlinebreak
Who and when are they playing last?
Miami Wed,APr 10,8:00 PMlinebreak

iv.

OpenNetsSchedule1819.htmlusing your browser and again look at its source code.What line in the file holds information about the game of the regular season (date, time,opponent)? It may be helpful to useCTRL-ForCOMMAND-Fhere and also work betweenthe file inRand in the text editor.
Ans:64 line

v.

Write a regular expression to extract the line that contains the time, location, andopponent of all games.

```
#regx of the target line
mypattern = '<tr class="(filled )*(bb--none )*Table2__tr
Table2__tr--sm Table2__even" data-idx="[0-9]{1,}".*</tr>'
#extract the line
line=regmatches(nets1819[64],gregexpr(mypattern,nets1819[64]))
```

vi.

Write a regular expression to split the whole line into 82 lines, with each line displayingthe information of one game. (You may obtain some hint from problem (vii).)

```
#pattern to split the lines
pattern='data-idx="[1-9][0-9]*"'
#extract and split the line into 82 lines
list=unlist(strsplit(as.character(line),pattern))[3:84]
```

vii.

Write a regular expression that will capture the date of the game. Then using thegrep()function find the lines in the file that correspond to the games. Make surethatgrep()finds 82 lines, and the first and last locationsgrep()finds match thefirst and last games you found in (ii).

```
#pattern of date
pattern1='[A-Z][a-z]{2}, [A-Z][a-z]{2} [0-9]+'
grep(pattern1,list)#grep total 82 lines
```

```
## integer(0)
```

viii.-xi

```
#pattern of date
pattern1='[A-Z][a-z]{2}, [A-Z][a-z]{2} [0-9]+'
#pattern of time
pattern2='[0-9]+:[0-9]{2} (PM|AM)'
#pattern of home and away
pattern3='">(@|vs)'
```

```
#pattern of Team name
pattern4='<img alt="[A-Z][A-z]+.*" title'
#pattern to extract team name
pattern5='".*"'
#store date
date=unlist(regmatches(list, gregexpr(pattern1, list)))
#store time
time=unlist(regmatches(list,gregexpr(pattern2,list)))
#store home and away
home=unlist(regmatches(list,gregexpr(pattern3,list)))
#create function to trim the home and away
awayorhome<-function(x){
  for (i in 1:length(x)){
    x[i]=substr(x[i],start=3,stop=nchar(x[i]))
  }
  return(x)
}
#home and away stored successfully
home=awayorhome(home)
#extract opponent
opponent=regmatches(list,gregexpr(pattern4,list))
#refine the extraction of opponent
opponent=unlist(regmatches(opponent,gregexpr(pattern5,opponent)))
#create function to refine the opponents
getopp<-function(x){
  for (i in 1:length(x)){
    x[i]=substr(x[i],start=2,stop=nchar(x[i])-1)
  }
  return(x)
}
#store opponent
opponent=getopp(opponent)
```

xii.

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 10games as seen from the web browser?

```
#group all data into data frame
df=as.data.frame(cbind(date,time,opponent,home))
#print the first 10 rows
head(df,10)
```

```
##   opponent home
## 1     <NA> <NA>
```

Ans: It matches perfectly.