5206_hw3_sw3449

Shuyuan Wang, sw3449

10/22/2019

i. save a file

The file is saved as `NetsSchedule1920.html`.

ii. Use the `readLines()` command to load the `NetsSchedule1920.html` file into a character vector in `R`. Call the vector `nets1920`.

```
nets1920 <- readLines("/Users/wangshuyuan/Desktop/GR5206-DS/hw3/NetsSchedule1920.html")
#(a) The number of lines in NetsSchedule1920.html
length(nets1920)
```

```
## [1] 71
```

```
#(b) Total number of charachters in the file
sum(nchar(nets1920))
```

```
## [1] 431488
```

```
#(c) Maximum number of characters in a single line
max(nchar(nets1920))
```

```
## [1] 276154
```

iii. 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.

```
date_exp <- '[A-Z][a-z]{2},\\s[A-Z][a-z]{2}\\s[0-9]+'
game_line <- grep(nets1920,pattern=date_exp)
game_line
```

```
## [1] 67
```

iv. Using the expression you wrotein (iii) along with the functions `gregexpr()` and `regmatches()`, extract the dates from the text file. Store this information in a vector called `date` to save to use below. Display the first six dates of the extracted dates vector.

```
date_locations <- gregexpr(nets1920[game_line],pattern=date_exp)
date <- regmatches(nets1920[game_line],date_locations)[[1]]
head(date,6)
```

```
## [1] "Wed, Oct 23" "Fri, Oct 25" "Sun, Oct 27" "Wed, Oct 30" "Fri, Nov 1"
## [6] "Sat, Nov 2"
```

v. Use the same strategy as in (iii) and (iv) to create a `time` vector that stores the time of the game. Notice that the length of this vector might be shorter because it only captures the games for the remainder of the season and the season is more than half over. Display the first six times of the extracted time vector.

```
time_exp <- '[0-9]+:[0-9]{2}\\s(AM|PM)'
time_locations <- gregexpr(nets1920[game_line],pattern=time_exp)
time <- regmatches(nets1920[game_line],time_locations)[[1]]
head(time,6)
```

```
## [1] "7:30 PM" "7:30 PM" "6:00 PM" "7:30 PM" "7:00 PM" "7:00 PM"
```

vi. 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. Display the first six values of the home vector.

Since before every '@' or 'vs' there is always an HTML code `<span class="pr2">`, we could use this to search for '@' or 'vs'. After obtaining the relevant string containing '@' or 'vs', use `substr` to obtain the keyword and create the indicator vector.

```
home_exp <- '<span class="pr2">(@|vs)'
home_locations <- gregexpr(nets1920[game_line],pattern=home_exp)
home_away <- regmatches(nets1920[game_line],home_locations)[[1]]
home_away <- substr(home_away,nchar('<span class=\"pr2\">')+1,nchar(home_away))
home <- ifelse(home_away=='vs',1,0)
head(home,6)
```

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

vii. 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.

Since before every opponent's name there is always an HTML code `<img alt="opponent_name" title=`, we could use this to search for the opponent's name. After obtaining the relevant string containing the opponent's name, use `substr` to extract the opponent individually.

```
opponent_exp <- '<img alt=\"([A-Z][A-Z]|[A-Z][a-z]+|[A-Z][a-z]+\\s[A-Z][a-z]+)\" title='
opponent_locations <- gregexpr(nets1920[game_line],pattern=opponent_exp)
opponent_str <- regmatches(nets1920[game_line],opponent_locations)[[1]]
opponent <- substr(opponent_str,11,nchar(opponent_str)-8)
head(opponent,6)
```

```
## [1] "Minnesota" "New York"  "Memphis"   "Indiana"   "Houston"   "Detroit"
```

viii. Construct a data frame of the four variables in the following order: `date`, `time`, `opponent`, `home`. Print the head and the tail of the dataframe. Does the data match the games as seen from the web browser? Note The `time` vector can have `NA`'s for the games that were already played.

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

```
##          date    time  opponent home
## 1 Wed, Oct 23 7:30 PM Minnesota    1
## 2 Fri, Oct 25 7:30 PM  New York    1
## 3 Sun, Oct 27 6:00 PM   Memphis    0
## 4 Wed, Oct 30 7:30 PM   Indiana    1
## 5  Fri, Nov 1 7:00 PM   Houston    1
## 6  Sat, Nov 2 7:00 PM   Detroit    0
```

```
tail(my_df)
```

```
##           date    time      opponent home
## 77  Sun, Apr 5 6:00 PM        Dallas    1
## 78  Tue, Apr 7 8:00 PM Oklahoma City    0
## 79  Thu, Apr 9 8:00 PM     Milwaukee    0
## 80 Sat, Apr 11 8:00 PM       Chicago    0
## 81 Mon, Apr 13 7:00 PM     Cleveland    0
## 82 Wed, Apr 15 7:30 PM     Milwaukee    1
```