Homework 4

Statistical Computing, 36-350

Due Wednesday Sept 30, 2015

Your homework must be submitted in R Markdown format. We will not (indeed, cannot) grade homeworks in other formats. Your responses must be supported by both textual explanations and the code you generate to produce your result. (Just examining your various objects in the "Environment" section of R Studio is insufficient—you must use scripted commands.)

Background: In this homework, we'll show you how you might become rich yourself by collecting the schedule for the upcoming 2015-2016 NHL season, including the links to Ticketmaster, so that you can corner the resale market and get super rich.

Just kidding! There's no way with what we've taught you so far that you'll be able to get past the anti-bot measures on Ticketmaster without defeating the CAPTCHA systems that were designed and built right here at CMU. What you can do, though, is reassemble this series of games in an R data frame for machine-readable use. A To do so, you will use regular expressions to extract the useful information from the surrounding HTML code.

1. Use the `readLines()` command to load the file at http://www.stat.cmu.edu/~ryantibs/statcomp/homework/NHL1516.html into a character vector called `nhl1516`. Remember to pass this URL directly to the `readLines()` function.

    a. How many lines does it contain?
    b. What is the total number of characters in the file?
    c. What is the maximum number of characters in a line?

2. Take a look at the webpage `NHL1516.html`. You should see the game table on the screen. There are 1230 regular-season games scheduled. Who is playing in the first game? In the final game?

3. Now, download the file `NHL1516.html` to your computer and open in a text editor. What line in the file corresponds to game 1? Which line corresponds to game 1230? How do each of these lines begin?

4. Our goal is to extract useful information about the games – the date, game time (in Eastern Time), away and home teams. As a first step, write a regular expression to capture the date. Use `grep()` to check that this has exactly 1230 matches and that the first and last locations match the first and last games (use this for question 5a).

5. We will extract some information to save, about games.

    a. Using the regular expression above, and the functions `regexpr()` and `regmatches()`, extract all the dates from the text and create a corresponding vector `date`. Save this for a further step.
    b. Now, identify the away and home teams with regular expressions for each. Extract and save these values in their own vectors. Use the HTML around these names to guide your search. You may have to add escape marks to properly recognize some characters – for example, periods must be escaped as `\\.` in your expression.
    c. Identify the game time in the code and create a regular expression for it. Note that there are two times – the local time and the Eastern time. Make sure your expression gets the Eastern time using clues from the HTML. Save it.

6. Construct a data frame consisting of these four variables. Print the frame from rows 1225 to 1230. Does the data match that in the last 6 rows of the table as seen from your web browser?

7. **Bonus question**: Create a regular expression to extract all the away teams coming to play with Pittsburgh Penguins (at home).