STAT GU4206/GR5206 Homework 3 [100 pts]
Due 11:59pm Monday, October 28th on Canvas

Your homework should be submitted on Canvas as an R Markdown file. Please submit the knitted .pdf (or .html) file **along with the .Rmd file**. We will not (and cannot) accept any other formats. Please clearly label the questions in your responses and support your answers by textual explanations and the code you use to produce the result. Note that you cannot answer the questions by observing the data in the "Environment" section of RStudio or in Excel – you must use coded commands.

Goals: regular expressions, character functions in `R`, and web scraping.

In this assignment, we're going to scrape the 2019-2020 Brooklyn Nets Regular Season Schedule (they're a basketball team from Brooklyn that plays in the NBA). We will take the regular season schedule from `http://www.espn.com/` and reassemble the game listings in an `R` data frame for computational use.

To do this, perform the following tasks:

i. Open the link `https://www.espn.com/nba/team/schedule/_/name/bkn/season/2020/seasontype/2`. Display the source code and copy and paste this code into a text editor. Then save the file as `NetsSchedule1920` 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. **If this does not work and you are unable to open the file up, do not worry and skip to problem ii.**

ii. Use the `readLines()` command we studied in class to load the `NetsSchedule1920.html` file into a character vector in `R`. Call the vector `nets1920`. **Note:** You can also skip Part i all together and use the `readLines()` command on the url `https://www.espn.com/nba/team/schedule/_/name/bkn/season/2020/seasontype/2`, which is arguably easier.

   a. How many lines are in the `NetsSchedule1920.html` file?

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

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

Using `NetsSchedule1920.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.

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.

iv. Using the expression you wrote in (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. **Hint:** We did something like this in class.

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.

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.

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.

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.