Econ 5253 - Spring 2023 **Problem Set 5** **Due: Mar. 7 beginning of class**

This problem set will give you practice with aggregating data from the web.

As with the previous problem sets, you will submit this problem set by pushing the document to *your* (private) fork of the class repository. You will put this and all other problem sets in the path `/DScourseS23/ProblemSets/PS5/` and name the file `PS5_LastName.*`. Your OSCER home directory and GitHub repository should be perfectly in sync, such that I should be able to find these materials by looking in either place. Your directory should contain four files:

- `PS5_LastName.R` (you can also do this in Python or Julia if you prefer)
- `PS5_LastName.tex`
- `PS5_LastName.pdf`

1. Type `git pull origin master` from your OSCER `DScourseS23` folder to make sure your OSCER folder is synchronized with your GitHub repository.
2. Synchronize your fork with the class repository by doing a `git fetch upstream` and then merging the resulting branch.
3. Find a webpage that has data that interests you, or that might be useful to you, but which doesn't have an API. Collect data from that webpage by following these steps:
    - Use SelectorGadget to select the CSS tags that correspond to the data you'd like to collect. This could be from anywhere—Wikipedia, a news website, a sports or pop culture website, etc.
    - Use a package such as `rvest`, `BeautifulSoup`, or `Gumbo.jl` to parse the HTML and CSS code.
    - Manipulate the parsed code so that you end up with a tabular data set (e.g. a data frame or a CSV file).
    - In a new TeX file (`PS5_LastName.tex`), tell me about the data. What about it interests you? Will it be useful to you at some point in the future? Did you use any online tutorials to help you parse the data? If so, which ones?
4. Find a website or other data source that hosts an API. For example: twitter, Quandl, Google Finance, or other sources. There exist many user-written packages in R, Python, and Julia that have set up the leg work for interfacing with more prominent APIs.
    - Use the API to generate a table of data that is interesting/meaningful to you in some way.

- In another section of the same TeX file as referenced in the previous question, tell me about what you found in the data (if anything) that was interesting or useful to you. What packages did you use (if any)?

5. Compile your .tex file, download the PDF and .tex file, and transfer it to your cloned repository on OSCER using your SFTP client of choice (or via `scp` from your laptop terminal). You may also copy and paste your .tex file from your browser directly into your terminal via `nano` if you prefer, but you will need to use SFTP or `scp` to transer the PDF.[1]

6. You should turn in the following files: .tex, .pdf, and any additional scripts required to reproduce your work. Make sure that these files each have the correct naming convention (see top of this problem set for directions) and are located in the correct directory (i.e. `~/DScourseS23/ProblemSets/PS5`).

7. Synchronize your local git repository (in your OSCER home directory) with your GitHub fork by using the commands in Problem Set 2 (i.e. `git add`, `git commit -m "message"`, and `git push origin master`). More simply, you may also just go to your fork on GitHub and click the button that says "Fetch upstream." Then make sure to pull any changes to your local copy of the fork. Once you have done this, issue a `git pull` from the location of your other local git repository (e.g. on your personal computer). Verify that the PS5 files appear in the appropriate place in your other local repository.

[1] If you want to try out something different, you can compile your .tex file on OSCER by typing `pdflatex myfile.tex` at the command prompt of the appropriate directory. This will create the PDF directly on OSCER, removing the requirement to use SFTP or `scp` to move the file over.