Econ 5253 - Spring 2026 **Problem Set 2** **Due: Feb. 10 beginning of class**

This problem set will provide an opportunity for you to practice accessing computing resources on the OSCER server here at OU. As with the previous problem set, 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 `/DScourseS26/ProblemSets/PS2/` and name the file `PS2_LastName.pdf`.

1. Make sure you are able to access OSCER.
2. Setting up SSH with GitHub on OSCER. Please follow these steps closely to allow SSH authentication on OSCER:
    1. `ssh` into OSCER from your RStudio terminal
    2. at the bash prompt, type `cd .ssh`
        - If you get an error that that directory doesn't exist, then type `mkdir .ssh`, followed by `cd .ssh`
    3. type `ssh-keygen -t rsa -b 4096 -C "your.email@address.com"` where you fill in with the email address you use for GitHub
        - when it asks you for the file in which to save the key, label it "ghDS26"
        - you may enter a passphrase to protect the ssh key but it is not required; you can simply hit enter twice
    4. it should tell you "Your identification has been saved in ghDS26" and "Your public key has been saved in ghDS26.pub" and then print out some other stuff about a fingerprint and randomart image.
    5. type `cd ~`
    6. in a web browser, visit `https://github.com/settings/keys` and click "new SSH key". Name it "OSCER"
    7. Copy the contents of your `ghDS26.pub` file and paste it into the key box. You can do this in a couple of ways:
        - type `cat ~/.ssh/ghDS26.pub` and then select all of the text with your mouse cursor. In some terminals, this also copies the text. Otherwise you can try and right-click and then copy it. Then paste it into the box on the GitHub SSH key webpage. (Note: the tilde ~ should be typed directly next to the slash with no space.)
        - email the key to yourself by typing `mail -s "public key" your.email@address.com < ~/.ssh/ghDS26.pub` where you replace `your.email@address.com` with your actual email address. Then copy/paste it into the box on the GitHub SSH key webpage

Note: Before proceeding to questions 3 or 4 (which involve git operations), complete question 9 to set up the ssh-agent in your .bash_profile file. If all of that was too much for you, you can also check out this YouTube video for a walkthrough: https://www.youtube.com/watch?v=s6KTbytdNgs

3. If you have already cloned your forked GitHub repository to your home directory on OSCER, you will need to adjust the URL for the remote by following these steps:
    1. Make sure you are in your local repository directory by typing `cd ~/DScourseS26`. (Note: the tilde ~ should be typed directly next to the slash with no space in between.)
    2. Type `git remote set-url origin git@github.com:[your-github-username]/DScourseS26.` where you replace `[your-github-username]` (including the brackets) with your actual GitHub username. Note that you can also get this url from the same place that you get the clone on your github fork. Just go to the "SSH" instead of "HTTPS" section.
    3. Do a `git pull` and then a `git push` to make sure things work well with the new SSH tokens.
4. If you have not already, clone your forked GitHub repository to your home directory on OSCER by following these steps:
    1. Make sure you are in your home directory by typing `cd ~`. (Note: the tilde ~ is a shortcut for your home directory and should be typed directly next to the slash with no space in between.)
    2. Type `git clone git@github.com:[your-github-username]/DScourseS26.git` where you replace `[your-github-username]` (including the brackets) with your actual GitHub username.
    3. Check that everything works by typing `ls` and hitting enter. You should see a directory called "DScourseS26" and if you change to that directory, you should see an identical directory and file structure to what is on your GitHub private fork of the course repo.
5. Go to `www.overleaf.com` and create another .tex document, this time naming it `PS2_LastName.tex`. In it, write down a list (in a LaTeX `itemize` environment) that outlines the main tools of a data scientist (as discussed in class). Compile the PDF.
6. Compile your .tex file, download the PDF and .tex file, and transfer it to your cloned repository on OSCER. There are many ways to do this; you may ask an AI chatbot

(which will probably give you some terminal commands you can use at the RStudio terminal or similar) or simply drag-and-drop using an SFTP client like FileZilla, WinSCP or Cyberduck. Do **not** put these files in your fork on your personal laptop; otherwise git will detect a merge conflict and that will be a painful process to resolve.

7. Make sure that your .tex and .pdf files have the correct naming convention (see top of this problem set for directions) and are located in the correct directory. If the directory does not exist, create it using the `mkdir` command.

8. Update your OSCER git repository (in your OSCER home directory) by using the following commands:

   - `cd ~`
   - `cd DScourseS26/ProblemSets/PS2`
   - `git add PS2_LastName.*`
   - `git commit -m 'Completed PS2'`
   - `git push origin master`

   Once you have done this, issue a `git pull` from the location of your other local git repository (e.g. on RStudio on your personal computer). Verify that the PS2 directory appears in the appropriate place in your other local repository.

9. Make sure that you have R version 4.0.2 (or higher) installed on OSCER. To do this, take the following steps:

   1. At the command prompt, type `nano ~/.bash_profile` and press enter.
   2. Type `module load R/4.0.2-foss-2020a` somewhere in the middle of that file.
   3. At the bottom of the file, add the following improved ssh-agent configuration (this prevents starting multiple ssh-agent processes):

```
# https://rabexc.org/posts/pitfalls-of-ssh-agents
ssh-add -l &>/dev/null
if [ "$?" == 2 ]; then                      # Check connection to ssh-agent
  test -r ~/.ssh-agent && \
    eval "$(<~/.ssh-agent)" >/dev/null      # If no connection, source saved env

  ssh-add -l &>/dev/null
  if [ "$?" == 2 ]; then                    # Check connection to ssh-agent agai
    (umask 066; ssh-agent > ~/.ssh-agent)   # No agent running; start one
```

```
    eval "$(<~/.ssh-agent)" >/dev/null      # Source new env variables
    ssh-add ~/.ssh/ghDS26                   # Add private keys
  fi
fi
```

4. Then push Control+X and then type "yes" (or "y") when it asks if you want to modify the file.

10. Now open R (on OSCER), verify that it is running the version mentioned above, and install the `xml2` package by typing `install.packages('xml2')` at the command prompt. (**NOTE: The package is "X-M-L 2," NOT "X-M 12"**) Answer 'yes' to install it in your home directory. Enter the number of any of the server mirrors that come up (I usually choose Texas). Building the package may take a couple of minutes. Once that has finished, repeat the process, but substitute `tidyverse` for `xml2`. Building `tidyverse` may take as much as 10-15 minutes. You don't need to watch it build, but be sure not to close your SSH terminal window or otherwise disconnect your session. When you are done with the build, exit R and log off of OSCER. You can always exit a program (or logout) on OSCER by typing Control+D.

11. Update your fork of the class repository on both your personal laptop as well as on OSCER, since I've been continuously updating things (and as part of PS1, everyone in the class submitted pull requests). The command to do this is `git pull upstream master`. A step-by-step help for how to do this is located here. More simply, you may also just go to your fork on GitHub and click the button that says "Sync fork" or "Fetch upstream" or similar.