Econ 5253 - Spring 2024 **Problem Set 8** **Due: Apr. 2 beginning of class**

This problem set will give you practice in manipulating matrices, optimizing commonly used objective functions, and in checking your answers using simulated data.

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 `/DScourseS24/ProblemSets/PS8/` and name the file `PS8_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 at least three files:

- `PS8_LastName.R` (you can also do this in Python or Julia if you prefer)
- `PS8_LastName.tex`
- `PS8_LastName.pdf`

1. Type `git pull origin master` from your OSCER `DScourseS24` 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. (`git merge upstream/master -m "commit message"`)
3. Install the package `nloptr` if you haven't already.
4. Using R, Python, or Julia, create a data set that has the following properties:
   - Set the seed of the random number generator by issuing the (R) command `set.seed(100)`[1]
   - X is a matrix of dimension $N = 100,000$ by $K = 10$ containing normally distributed random numbers, except the first column which should be a column of 1's.
   - ε (call it `eps` in your code) is a vector of length N containing random numbers distributed $N\left(0, \sigma^2\right)$ where $\sigma = 0.5$ (so $\sigma^2 = 0.25$).
   - β (call it `beta` in your code) is a vector of length 10. Let β have the following values:

$$\beta = \begin{bmatrix} 1.5 & -1 & -0.25 & 0.75 & 3.5 & -2 & 0.5 & 1 & 1.25 & 2 \end{bmatrix}' \tag{1}$$

   - Now generate Y which is a vector equal to $X\beta + \varepsilon$.

[1] Similar commands exist in Python and Julia.

5. Using the matrices you just generated, compute $\hat{\beta}_{OLS}$, which is the OLS estimate of β using the closed-form solution (i.e. compute $\hat{\beta}_{OLS} = (X'X)^{-1} X'Y$). [HINT: check here for matrix algebra operations in R] How does your estimate compare with the true value of β in (1)?

6. Compute $\hat{\beta}_{OLS}$ using gradient descent (as we went over in class). Make sure you appropriately code the gradient vector! Set the "learning rate" (step size) to equal 0.0000003.

7. Compute $\hat{\beta}_{OLS}$ using `nloptr`'s L-BFGS algorithm. Do it again using the Nelder-Mead algorithm. Do your answers differ?

8. Now compute $\hat{\beta}_{MLE}$ using `nloptr`'s L-BFGS algorithm. The code for the gradient vector of this problem is listed below:

```
gradient <- function(theta,Y,X) {
grad     <- as.vector(rep(0,length(theta)))
beta     <- theta[1:(length(theta)-1)]
sig      <- theta[length(theta)]
grad[1:(length(theta)-1)] <- -t(X)%*%(Y - X%*%beta)/(sig^2)
grad[length(theta)]       <- dim(X)[1]/sig - crossprod(Y-X%*%beta)/(sig
   ^3)
return ( grad )
}
```

9. Now compute $\hat{\beta}_{OLS}$ the easy way: using `lm()` and directly calling the matrices Y and X (no need to create a data frame). Make sure you tell `lm()` not to include the constant! This is done by typing `lm(Y ~ X -1)`

   Use `modelsummary` to export the regression output to a .tex file. In your .tex file, tell me about how similar your estimates of $\hat{\beta}$ are to the "ground truth" β that you used to create the data in (1).

10. 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 or simply drag-and-drop using VS Code. 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.

11. You should turn in the following files: .tex, .pdf, and any additional scripts (e.g. .R, .py, or .jl) 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. `~/DScourseS24/ProblemSets/PS8`).

12. 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 PS8 files appear in the appropriate place in your other local repository.