STA 141C - Big Data & High Performance Statistical Computing **Spring 2023**

Homework # 2

Lecturer: Bo Y.-C. Ning *Due May 12, 2022*

Due **May 12, 2023** by 11:59pm.

This homework has two major goals: 1) compare the computational speeds for solving linear equation using QR, Cholesky, and GE/LU methods; 2) learn how to implement parallel computing in R or python.

Directions:

1. Submit your homework using the file name "**LastName_FirstName_hw2**"
2. Answer all questions with complete sentences. For proofs, please provide the intermediate steps.
3. Your code should be readable; writing a piece of code should be compared to writing a page of a book. Adopt the **one-statement-per-line** rule. Consider splitting a lengthy statement into multiple lines to improve readability. (You will lose one point for each line that does not follow the one-statement-per-line rule)
4. To help understand and maintain code, you should always add comments to explain your code. (homework with no comments will receive 0 points). For a very long comment, break it into multiple lines.
5. Submit your final work with one **.pdf** (or **.html**) file to Canvas. I encourage you to use LaTeX for writing equations and proofs. Handwriting is acceptable, you have to scan it and then combine it with the coding part into a single .pdf (or .html) file. Handwriting should be clean and readable.
6. For Jupyter Notebook users, put your answers in new cells after each exercise. You can make as many new cells as you like. Use code cells for code and Markdown cells for text.
7. This assignment will be graded for correctness.

Questions:

1. Read in the 'longley.dat' with the response (number of people employed) in the first column and six explanatory variables in the other columns (GNP implicit price deflator, Gross National Product, number of unemployed, number of people in the armed forces, "noninstitutionalized" population % 14 years of age, year). Include an intercept in you model.
2. Assuming linear model $y \sim N(X\beta, \sigma^2 I)$, compute 1) regression coefficients $\hat{\beta} = (X'X)^{-1}X'y$, 2) standard errors of $\hat{\beta}$, which is $\hat{\sigma}\sqrt{\text{diag}((X'X)^{-1})}$, and 3) variance estimate $\hat{\sigma}^2 = (y - X\hat{\beta})'(y - X\hat{\beta})/(n-p)$ using following methods: GE/LU decomposition, Cholesky decomposition, and QR decomposition, and compare the computation speed for each method. Please compute them directly using numerical linear algebra functions; you can use the "black-box" function (e.g., lm() in R or sklearn.linear_model.LinearRegression in python) **only to check your results.** (Hint: chol2inv() function in R computes the inverse of a matrix from its Cholesky factor. In python, you may try cho_solve())
3. One popular regularization method is the ridge regression, which estimates regression coefficients by minimizing a penalized least squares criterion

$$\frac{1}{2}\|y - X\beta\|_2^2 + \frac{\lambda}{2}\|\beta\|_2^2,$$

show that the ridge solution is given by

$$\hat{\beta}_\lambda = (X'X + \lambda I_p)^{-1} X'y.$$

4. Compute the ridge regression estimates $\hat{\beta}_\lambda$ at a set of different values of λ (e.g., 0, 1, 2, ..., 100) by solving it as a least squares problem. Plot the ℓ_2-norm of the ridge coefficients $\|\hat{\beta}_\lambda\|$ as a function of λ. You can use either QR or Cholesky method.

5. Implement your code using parallel computing.

6. Find out which method is the `lm()` function in R is using? And which algorithm is being used? **Or** find out which method is the linear regression function (there are multiple, but only need to choose one) in numpy/scipy is using? And which algorithm is being used?