# Homework 2

## M1399.000100, Seoul National University, Spring 2021

### Due 23:59 Sunday, 2021-04-25

# M1399.0001000 Homework 2, Spring 2021, Seoul National University

Due 23:59 2021-04-25

**No late submission is accepted.**

## Q1. LU decomposition

1. Let $A\in\mathbb{R}^{n\times n}$ be nonsingular. Show that $A$ has an LU decomposition if and only if for each $k$ with $1\le k \le n$, the upper left $k\times k$ block `A[1:k,1:k]` is nonsingular. Prove that this LU decomopsition is unique.

2. Write an R function `LUdecomp` with interface

   ```
   LUdecomp(obj, tol=1e-8)
   ```

   The decomposition **must** be done *in place*. That is, if $A=LU \in \mathbb{R}^{n\times n}$, the $U$ should overwrite the upper triangular part of the input matrix `A`, and the strictly lower triangular part of `A` should be overwritten by the same part of the $L$ matrix computed. (Where does the diagonal part of $L$ go?) Since R does not allow in-place modification of atomic objects, you are recommended to use a Reference Class (RC) object. The RC for this homework can be declared by

   ```
   setRefClass("LUclass",
       fields = list(
           mat = "matrix",
           vec = "vector"
       )
   )
   ```

   A RC object can be created, for instance, by

   ```
   A <- matrix(c(1.0, 0.667, 0.5, 0.333), nrow=2)
   b <- c(1.5, 1.0)
   obj <- new("LUclass", mat=A, vec=b)
   ```

   Matrix `A` stored in `obj` can be referenced by `obj$mat`, and vector `b` can be by `obj$vec` (field `vec` is reserved for the next question).

   You must also implement partial pivoting: function `LUdecomp` must return a `list` that consists of two elements: the first element of the list is the array of the permutation indexes of the rows, and the second element is the indicator of success: if `A` is (numerically) singular, the function must return the row index where singularity occurs (where may a singularity occur in the LU decomposition?) as the second return value; otherwise return `0`. Use `tol` to determine the singularity.

3. Using the `LUdecomp` function written above, write function `LUsolve0` that solves the linear equation \(Ax = b\) with interface

   ```
   LUsolve0(obj)
   ```

   again *in place*. That is, in addition to `A` overwritten by `LUdecomp`, vector `b` should be overwritten by the solution \(A^{-1}b\). Then, write a wrapper function

   ```
   LUsolve(A, b)
   ```

   which does **not** alter neither `A` nor `b` but solves \(Ax=b\) by calling `LUdecomp`. Compare your results with the R expression `solve(A, b)`.

Write your functions in a separate `.R` file within your branch.

# Q2. Choleksy decomposition

1. 심송용, 제 2장 연습문제 2.3

2. Complete the proof of the Cholesky decomposition by showing that
   - \(\mathbf{A}_{22}\) is positive definite, and
   - \(\mathbf{A}_{22} - \mathbf{b} \mathbf{b}^T = \mathbf{A}_{22} - a_{11}^{-1} \mathbf{a} \mathbf{a}^T\) is positive definite of size \((n-1)\times(n-1)\).

# Q3. QR decomposition

1. 심송용, 제 6장 연습문제 6.1, 6.2.

2. From the lecture note on QR decompostion, explain why classical Gram-Schmidt (`cgs()`) fails with the given matrix `A`.

3. From the same lecture note, explain why the modified Gram-Schmidt (`mgs()`) fails with the given matrix `B`. Will the classical Gram-Schmidt succeed?

4. Implement the Householder QR decomposition in R.

   - The algorithm should be **in-place**: let the \(\mathbf{R}\) matrix occupy the upper triangular part of the input \(\mathbf{X}\in\mathbf{R}^{n\times p}\). Below the diagonal place the vectors \(\mathbf{u}_k\) that define the Householder transformation matrix \(\mathbf{H}_k=\mathbf{I}-2\mathbf{u}_k\mathbf{u}_k^T/\mathbf{u}_k^T\mathbf{u}_k\). By setting the first element of \(\mathbf{u}_k\) to 1, you can fit in these vectors in \(\mathbf{X}\). The algorithm should return an additional vector storing the values of \(\mathbf{u}_k^T\mathbf{u}_k\). This is how the LAPACK routine `geqrf` is designed. Note that \(\mathbf{Q}\) can be recovered from \(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p\). The function interface should be

     ```
     householder <- function(qrobj)
     ```

taking a Reference Class (RC) object

```
setRefClass("QRclass",
    fields = list(
        mat = "matrix",
        vec = "vector"
    )
)
```

- Write a separate routine

```
recoverQ <- function(qrobj)
```

that recovers $Q$. (*Hint*. For $\mathbf{P}_i=\mathbf{I}-2\mathbf{v}_k\mathbf{v}_k^T$ with $\|\mathbf{v}_k\|=1$, $\mathbf{P}_1\mathbf{P}_2 \cdots \mathbf{P}_{n} = \mathbf{I}-\mathbf{V}\mathbf{T}\mathbf{V}^T$, where $\mathbf{V}=[\mathbf{v}_1 | \mathbf{v}_2 | \dotsb | \mathbf{v}_n]$ for some *upper triangular* matrix $\mathbf{T}$.)

- Using your function, compute the QR decomposition of the matrices `A` and `B` of the previous question. Compare the orthogonality of the computed $Q$ matrix.

# Q4. Least squares

The Longley data set of labor statistics was one of the first used to test accuracy of least squares computations. This data set is available at the National Institute of Standards and Technology (NIST) website, consists of one response variable (number of people employed) and six predictor variables (GNP implicit price deflator, Gross National Product, number of unemployed, number of people in the armed forces, 'noninstitutionalized' population $\ge$ 14 years of age, year) observed yearly from 1947 to 1962, and is also available in R package .

1. Download the data set from the NIST website, read into R, and construct a data matrix $\mathbf{X}$ for linear model $\mathbf{y}=\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$. Include an intercept in your model.
    - Using the R command `svd()`, list up the 7 singular values of $\mathbf{X}$. What is the condition number of $\mathbf{X}$?
    - Construct the Gram matrix $\mathbf{G} = \mathbf{X}^T\mathbf{X}$. List up the 7 singular values of $\mathbf{G}$. What is the condition number of $\mathbf{G}$?

2. Using the function you wrote for Q2, compute the regression coefficients $\hat{\boldsymbol{\beta}}$, their standard errors, and variance estimate $\hat{\sigma}^2$. Verify your results using the R function .

3. Using the Cholesky decomposition of $\mathbf{G}$, compute the regression coefficients $\hat{\boldsymbol{\beta}}$, their standard errors, and variance estimate $\hat{\sigma}^2$. Compare the results with the values of the above question.

# Q5. Iterative method

1. Show that the norm $\|\mathbf{x}\|_{\delta}$ in the lecture note on iterative methods is indeed a vector norm

2. 심송용, 제 3장 연습문제 3.3.

3. Consider the system of linear equations \[ \begin{split} x_1 + 4x_2 + x_3 &= 12, \\ 2x_1 + 5x_2 + 3x_3 &= 19, \\ x_1 + 2x_2 + 2x_3 &= 9. \end{split} \]

    - Determine the \(\mathbf{D}\), \(\mathbf{L}\), and \(\mathbf{U}\) matrices of the Gauss-Seidel method and determine the spectral radius of \((\mathbf{D} + \mathbf{L})^{-1}\mathbf{U}\).

    - Do two steps of the Gauss-Seidel method starting with \(\mathbf{x}^{(0)} = (1, 1, 1)\), and evaluate the Euclidean norm of the difference of two successive approximate solutions.

    - Do two steps of the Gauss-Seidel method with successive overrelaxation using \(\omega = 0.1\), starting with \(\mathbf{x}^{(0)} = (1, 1, 1)\), and evaluate the Euclidean norm of the difference of two successive approximate solutions.